Introducing Recent Electrical Engineering Developments into undErgraduate cuRriculum

Co-funded by the Erasmus+ Programme of the European Union

Reference No.: IREEDER-D2.2
Date: June. 15, 2021
Version: v2.0

DELIVERABLE 2.2

Teaching Materials - Cyber Security Lecture Notes

Written by	**Responsibility**
Felipe Gil-Castiñeira (UVIGO)	WP4 Leader
Cristina López-Bravo (UVIGO)	Member
René Lastra Cid (UVIGO)	Member
Enrique Costa-Montenegro (UVIGO)	Member
Beatriz Lorenzo Veiga	Member
Marios Raspopoulos (UCLAN)	WP2 Leader
Eliana Stavrou (UCLAN)	Member
Fabrizio Granelli (UNINT)	Member
Jonathan Rodriguez (IT)	WP5 Leader
Georgios Mantas (IT)	Member
Maria Papaioannou (IT)	Member
Claudia Barbosa (IT)	Member
Filippos Pelekoudas Oikonomou (IT)	Member
Edited by	
Felipe Gil-Castiñeira (UVIGO)	WP4 Leader
Marios Raspopoulos (UCLAN)	WP2 Leader
Approved by	
Saud Althunibat (AHU)	Project Coordinator

This publication was produced with the financial support of the European Union. Its contents are the sole responsibility of the partners of IREEDER project and do not necessarily reflect the views of the European Union

Introducing Recent Electrical Engineering Developments into undErgraduate cuRriculum

Introduction to Cybersecurity

Editor: Felipe Gil-Castiñeira

University of Vigo

May 2021

Co-funded by the Erasmus+ Programme of the European Union

This publication was produced with the financial support of the European Union. Its contents are the sole responsibility of the partners of IREEDER project and do not necessarily reflect the views of the European Union

Abstract

This document serves as a comprehensive handbook for the "**Cybersecurity Course**" prepared within the scope of the IREEDER project which is funded by the Erasmus+ programme. (Call: Capacity Building in the field of Higher Education, Project No. 609971-EPP-1-2019-JO-EPPKA1-CBHE-JP (2019-1975/001-001)).

The course aims to present the fundamental concepts in cybersecurity in order to teach the basic techniques for optimizing security on personal computers and small networks and to teach how to design and code secure applications. The specific learning outcomes are:

1. To recognize and apply the fundamental concepts related to cybersecurity and cybersecurity management (such as confidentiality, integrity and availability, vulnerability, threat, risk, security policies, guides and standards).
2. To apply security design principles to the engineering lifecycle, using the appropriate security models and architectures, tools, controls and countermeasures, based on security standards.
3. To apply secure design principles to network architecture, actively securing network components and communication channels.
4. Identify and use the principal security operations: logging and monitoring, implementing protection and mitigation measures, using recovery strategies, responding to incidents, and updating the systems.
5. To examine and apply security in the software development life cycle, enforcing software security controls, and assessing both software effectiveness and security.
6. To appraise the impact of new technologies, such as cloud computing, smart grid or BYOD (Bring Your Own Device) on cybersecurity.

Chapter 1 serves as an introductory overview of the CS field. It defines Security and Risk Management, basic concepts (confidentiality, integrity, availability and privacy), legal and

regulatory issues, documented security policy, standards, procedures, and guidelines, risk management concepts and threat modelling. **Chapter 2** serves as an introduction to security engineering and how to implement and manage an engineering lifecycle using security design principles. It defines security models and architectures, countermeasures based upon information systems security standards. **Chapters 3 and 4** are about Cryptography & Key Management & Cryptography Services. They cover Cryptographic Lifecycle, Cryptographic Types, Public Key Infrastructure (PKI) and Key Management practices, Digital Signatures, Digital Rights Management (DRM), Non-repudiation, Integrity (hashing and salting), Methods of Cryptanalytic Attacks. Chapters 5, 6 and 7 are about Communications and Network Security. **Chapter 5** is about secure design principles and cryptography used to maintain communications security. **Chapter 6** presents the securing network components, data, operation of hardware, transmission media, network access control devices and endpoint security. **Chapter 7** is about securing communication channels, remote access, data communications and virtualized networks. Chapters 8, 9 and 10 are about Security Operations. **Chapter 8** describes login, monitoring and access control. **Chapter 9** is about intrusion detection and prevention. **Chapter 10** describes recovery and incident response, and **chapter 11** is about security assessment and testing, assessment and test strategies and penetration testing. **Chapter 12** presents security in the software development life cycle and software protection mechanisms. Finally, **chapter 13** overviews the impact of new technologies on Cybersecurity, Advanced Persistent Threats (APTs), BYOD and Technology Customization

Table of Contents

1 Security and Risk Management

Author(s): Maria Papaioannou
Georgios Mantas
Claudia Barbosa
Jonathan Rodriguez

1.1 Basic concepts: confidentiality, integrity, availability, and privacy

Computer security consists of all the measures and controls to ensure confidentiality, integrity, and availability of information system assets (i.e., hardware, software, firmware) and of the processed, stored, and communicated information.

This definition introduces the three principal security objectives:

1. **Confidentiality** includes *Data Confidentiality* and *Privacy*. The first term ensures that classified information is only made disclosed to authorized individuals, while the second one that individuals have the absolute control on what personal information of them may be gathered and stored, as well as who may be authorised to have access to that information.
2. **Integrity** includes *Data integrity* and *System integrity.* The first term ensures that only authorized individuals may modify particular data and only in a specified manner, while the second ensures that a system performs its expected function in a manner free of accidental and/or deliberate unauthorized manipulation of the information system´s assets system and the system itself.
3. **Availability:** Ensures that a system works promptly and as it is intended to work, and service is always available to authorized individuals when they ask for it.

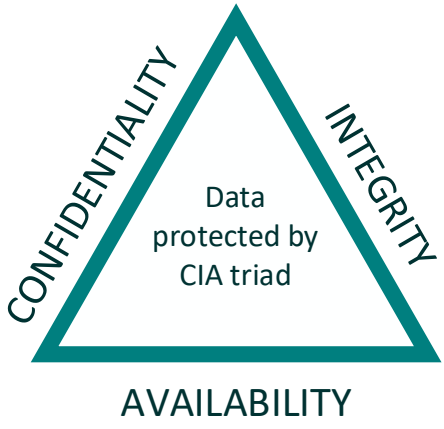


Figure 1-1: CIA triad [1]

Generally, the use of the CIA triad is a well-established approach to define security objectives for information security assets. Nevertheless, security experts feel that there is the need to include additional security concepts in order to complete the overall security picture.

Authenticity, Accountability and Reliability constitute three of the most commonly mentioned additional security objectives:

4. **Authenticity:** ensures that an entity or system is genuine and thus, it might be able to be authenticated and trusted. This term includes trust in the legitimacy of (i) a message transmission, (ii) a message itself, and/or (iii) a message´s originator. This property implies validating that individuals and systems are who they claim they are, as well as that the source of inputs arriving at the information system are trusted and reliable.
5. **Accountability:** introduces the requirement for individual actions to be traced exclusively to that particular individual. The need for accountability arises from the fact that truly secure systems are not yet an achievable goal. Therefore, systems must keep records of the occurred activities to permit later forensic analysis in order to be capable of tracing a security breach to a responsible party, or of aiding in transaction disputes.
6. **Reliability:** This term assures that an information system or component is able to function under specified conditions for a stated period of time.

Figure 1-2: Basic security concepts [1]

1.2 Legal and regulatory issues

Cybercrime or computer crime is a commonly used term to describe criminal activities in which computer or computer networks may be a target, a tool, or a place of criminal activity. The U.S. Department of Justice provided a categorization of computer related crime as follows:

- **Computers as storage devices: A** computer device can be used to further unauthorized and criminal activities by utilizing them as a passive storage medium, such as to store (i) stolen username-password lists, (ii) credit card numbers without any authorization from the legitimate entities, (iii) exclusive corporate data, or (iv) pirated commercial software, etc.
- **Computers as communications tools:** This cybercrime form includes crimes that are committed online, such as illegal trade and without any authorization of controlled substances, alcohol, prescription drugs, and guns; fraud and online scams; gambling and online betting; and child pornography.
- **Computers as targets:** This cybercrime form targets the availability, system integrity, data integrity, data confidentiality, and/or privacy of a computer system. In particular, the adversary may try to take control of the computer system without authorization (e.g., theft of service), to access and obtain data stored on that computer system, to intrude computer´s or server´s availability, or to modify stored information.

1.2.1 Law Enforcement Challenges

The deterrent effect of law enforcement on computer and network attacks is the successful criminal arrest and prosecution.

In particular, **law enforcement agencies (LEA)** consider that cybercrime poses exceptional complexities, and thus there is the need for sophisticated grasp of the technology to properly investigate cybercriminal activities. Lack of resources does not help to advance in this challenging area. More precisely, to deal with this kind of crime, investigators may need considerable computer capabilities in terms of communications and storage capacity, as well as processing power. Finally, an additional handicap is the global nature of cybercrime. For

instance, the majority of cybercrimes involve attackers who may be far away from the target system. Therefore, in order for the investigation to successfully advance, cooperation and collaboration with remote LEA is an important requirement to be considered. Initiatives like the International Convention on Cybercrime which establishes a universal terminology for cybercrimes and a framework for standardizing laws globally are very promising towards this direction.

1.2.2 Intellectual Property Infringement

Intellectual Property (IP) consists of the three main types of: (i) copyrights, (ii) trademarks, and (iii) patents, for which legal protection is available. The available legal protection is against the invasion of the rights protected by those patents, copyrights, and trademarks, in other words IP infringement. The right to seek legal protection against anyone infringing his/her property is granted to the Intellectual Property (IP) holder. Nevertheless, infringement may vary based on the type of IP.

1.2.2.1 Copyright

Legal protection for copyrights covers the protection of fixed expression of an idea. However, copyright law does not protect the idea itself. In case that (i) the work is original, and (ii) the creator has put this original work into a tangible form (e.g., a paper or hard copy, a software, a multimedia form in general), the creator can claim copyright at a national government copyright office, and start the process for the copyright.

Thereafter, the creator and copyrights holder have the exclusive rights secured against infringement of *Reproduction, Modification or derivative-works, Distribution, Public performance,* and *Public display.*

Table 1-1: Exclusive rights secured against infringement

Exclusive rights	**Description**
Reproduction right	Only the copyrights holder is authorized to reproduce and make copies of the work

Modification or derivative-works right	Concerns modifying a work to create a new or derivative work
Distribution right	Authorizes the copyright owner to publicly sell, rent, lease, or lend copies of the work
Public performance right	Authorizes the copyright owner to produce live performances of the work
Public display right	Authorizes the copyright owner to publicly display a copy of the work directly or by means of a film, slide, or television image

1.2.2.2 Patent

A patent for an invention consists the grant of a property right to the inventor. In particular, this includes "the right to prevent non-authorized individuals from making, utilizing, offering for sale, or selling" the invention in a nation or "importing" the invention into the nation. A patent may be (i) **Design patent,** (ii) **Utility patent,** and (iii) **Plant patent.**

Design patents may be granted to inventors of a novel, original, and ornamental design for an article of manufacture. **Utility patents** may be granted to inventors of any new and useful process, machine, article of manufacture, or composition of matter, as well as of any new and useful improvement thereof. Finally, **Plant patents** may be granted to anyone who discovers and asexually reproduces any distinct and new variety of plant.

1.2.2.3 Trademark

A trademark can be anything used in goods trade to indicate the traded their origin, as well as to differentiate them from the other traded goods in the market. It might be a word related to the traded goods, a symbol, a name, or a device. Equivalent to a trademark is the servicemark. However, the servicemark is mostly used to identify and differentiate the origin of a service rather than a product.

1.2.3 Intellectual Property Relevant to Network and Computer Security

Several forms of intellectual property are applicable in network and computer security. However, some of the most significant are the software, the databases, the digital content,

and the algorithms. First of all, **Software** comprises programs developed by individuals or vendors, as well as shareware, proprietary software developed by an institution for internal use. In these cases, copyright protection is available if desired authorized individuals apply for such a process. Furthermore, **Databases** include information gathered and managed in a way leading to a possible commercial value and thus, such databases may be protected and secured by copyright rights if desired. In addition, **Digital content** includes any original digital work demonstrated utilizing digital devices. Finally, **Algorithms** are software and/or source in general with a potential commercial value, like the RSA public-key cryptosystem, a patentable algorithm that was previously mentioned.

1.2.4 Privacy

Privacy often considerably overlaps with computer security. On one hand, the collection and storage in information systems of personal information by LEA, economic incentives and national security has increased dramatically lately. It is expected that aggregations of information on individuals constitute the most economically valuable electronic asset. While individuals are becoming more and more concerned about their privacy. Therefore, a wide spectrum of technical and legal approaches supporting the privacy rights of individuals has been introduced by global organizations.

1.2.5 European Union (EU) Directive on Data Protection

A great number of national governments and international organizations worldwide have introduced laws and regulations to protect privacy rights of individuals. In particular, in 1998, the European Union considered the Directive on Data Protection to: (a) guarantee that EU countries preserve basic privacy rights when handling individual data, and (b) prevent EU countries from impeding the free exchange of individual data in the EU. It is worthwhile to highlight that the Directive does not consist a law. The Directive is structured around the next principles regarding individual data usage:

- **Notice:** Organizations and corporations that handle personal information must always inform individuals what individual data they are gathering, the uses and exploitation

of that information, and what are the options that the individual may have regarding the collection, storage and processing of their personal information.

- **Consent:** Individuals must be able to decide whether and how their personal data is used by, or disclosed to, third parties. In particular, they must always have the right not to have any confidential information to be collected or used without explicit authorization.
- **Consistency:** Organizations and corporations may utilize personal data only in alignment with the conditions of the **notice** provided by the subject owning the data, as well as based on his/her choices with respect to its use.
- **Access:** Individuals must be able to access their own data at any time they desire, as well as to correct, alter, or delete any portion or the whole personal information.
- **Security:** The integrity and confidentiality of personal data must be protected and secured using efficient and effective security means – technical or other - by the responsible organizations and corporations.
- **Onward transfer:** Third parties collecting personal data must provide an equivalent level of privacy protection as the organization from whom the data are received.
- **Enforcement:** The EU Directive on Data Protection grants a private right of action to data subjects when organizations and/or governments do not follow the law and regulations regarding personal privacy rights. On top of that, each EU state member has a regulatory enforcement agency concerned with privacy rights enforcements.

More recently, the EU adopted further directives relevant to data privacy. In particular, in 2002, the Directive on Privacy and Electronic Communications introduced imposing an obligation on member states to safeguard the confidentiality of communications and related traffic data. In addition, in 2006, the Data Retention Directive introduced imposing an obligation on EU countries to ensure that communications service providers retain specified categories of communications data for a period of 6–24 months, and to make this data available to competent national authorities in accordance with national law. However, this latter directive was declared invalid by the Court of Justice of the European Union as being unjustified interference with the privacy rights enshrined in the EU Charter. This illustrates the difficult task legislators face balancing data surveillance with appropriate levels of privacy.

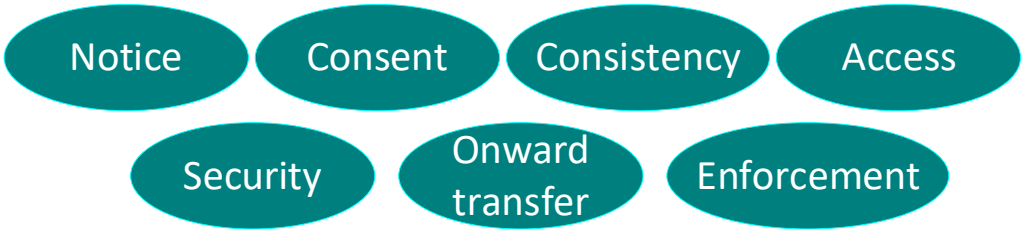


Figure 1-3: Principles of EU Directive on Data Protection [1]

On top of that, a set of operational requirements in a Privacy Class, that should be applied in every trusted system, is introduced in the Common Criteria specification. The definition of the set of functional requirements in the Privacy class aims to protect a user against the detection and the misuse of his/her identity by other individuals. This Common Criteria specification consists a valuable guide on the design and development of privacy mechanisms, including:

1. **Anonymity:** Assures that an individual may be user of a particular resource or service without disclosing his/her identity. On top of that, this means that the resource or service will not look for the real user´s identity to grant access. In addition, other subjects using the same resource or service will not be able to identify a particular user. It is worthwhile to highlight that anonymity, authorization and access control are totally different security functions, as the latter are linked to digital user IDs and not to personal data of users.
2. **Pseudonymity:** Assures that an individual may be user of a particular service or resource without revealing his/her identity, however still accountable for this kind of usage. In this case, the system shall assign an alias to the user in order to prevent other users from identifying this particular user. However, the identification of the user can be done by the system through his/her alias if needed.
3. **Unlinkability:** Assures that a user can be able to create multiple user profiles for accessing a number of different services and/or resources without allowing malicious actors to link that his/her profiles come from the same user.
4. **Unobservability:** Assures that an individual may be user of a particular service and/or resource without others observing that the particular user is using the resource or service.

Nevertheless, with the rapidly increasing interest of governments and non-government organizations to learn as much as possible personal information, technical approaches and policies should be applied to protect privacy rights of individuals. Regarding the technical approaches, the requirements to protect personal data stored on information systems can be addressed in part using the technical mechanisms developed for database security.

With regard to social media sites, technical controls include the provision of **suitable privacy settings** to manage who can view data on individuals, and notification when one individual is referenced or tagged in another's content. That is, by providing suitable access controls to this data, but on a scale far larger than that used in most IT systems. Although social media sites include some form of these controls, they are constantly changing. This causes frustration for users, who struggle to keep up to date with these mechanisms, and also indicates that the most appropriate controls have yet to be found.

Another technical approach for managing privacy concerns in big data analysis is to **anonymize the data**, removing any personally identifying information, before release to researchers or other organizations for analysis. Unfortunately, a number of recent examples have shown that such data can sometimes be re-identified, indicating that great care is needed with this approach. Done correctly, though, it does enable the benefits from big data analysis whilst avoiding issues of individual privacy concerns.

In terms of policy, guidelines are needed to manage the use and reuse of big data, ensuring suitable constraints are imposed to preserve privacy. In the context of human research, the guidelines for the use of digital data address the following fields:

(i) **Consent:** assures that participants will be explicitly informed about their participation in the particular research and thus they will be able to take informed decisions.

(ii) **Privacy and Confidentiality:** Privacy assures that that the participating individuals have the control over who can access their personal data, while Confidentiality assures that only authorized individuals should have access to personal data about the participants.

(iii) **Ownership and authorship:** Addresses who is responsible for handling and accessing the personal data about the participants, and at what point does an individual give up their right over the control on their personal data.

(iv) **Data sharing—assessing the social benefits of research:** Addresses the social benefits that result from data matching and reuse of personal data from one source and/or research project in another.

(v) **Governance and custodianship:** Address the oversight and implementation of the management, organization, access, and preservation of digital personal data.

1.2.6 Ethical Issues

Nowadays, the information systems are ubiquitous and valuable in organizations of all types, and thus privacy and security problems and ethical questions arise from the potential misuse and abuse of personal information and electronic communications. Ethics consist of the moral principles of a system that relate to the benefits and harms of certain actions, as well as to the appropriateness and wrongness of motives and ends of those actions.

1.2.6.1 Ethical Issues Related to Computers and Information Systems

Computers are used, nowadays, for storing personal data, such as healthcare records and other confidential information, and negotiable assets, such as bank records, and other financial information. Therefore, other types of databases have significant value. They can only be viewed, developed, managed, and modified by authorised means. Therefore, ethical concerns come from the different computer aspects. For instance, unauthorized use of **repositories and processors of personal information** stored in computer systems raises issues of appropriateness or fairness. In addition, there is the need for establishing new concepts of ownership as other assets have. Furthermore, **instruments of acts** should be established to define to what degree must both computer services and users of computers, data, and programs be responsible for the integrity and the appropriateness of a computer output. Finally, **symbols of intimidation and deception** should be carefully considered.

1.2.7 Codes of Conduct

Although a professional may be expected to have an internal moral compass, many areas of conduct may present moral ambiguities. Therefore, ethical codes of conduct adopted by professional societies are required in order to provide guidance to professionals and, at the same time, to articulate what both employers and customers have a right to expect.

A professional code of conduct may serve certain purposes, as following:

1. Instill confidence to the users of the product and/or service.
2. Be educational both for professionals to commit to take over a particular quality level their work, and managers to support their ethical responsibilities.
3. Provide a measure of support, in cases when the professional´s ethical actions create conflict with the costumer or consumer.
4. Support deterrence and discipline for professionals and employees.
5. Enhance the profession´s public image.

1.3 Security policies and standards

1.3.1 Security Policies

A corporate security policy defines the organizational objectives and strategies, as well as what process is leveraged in order to accomplish them. A security policy of an organization can be a large document or, in most cases, several documents related to each other. In particular, a security policy defines whether a system or a set of systems are “secure”. It is worthwhile to mention that a secure system under one policy may not be secure under a different policy. Security policies can be informal, formal, abstract, or highly mathematical in nature. A security policy considers all relevant aspects of ***confidentiality, integrity, and availability*** security objectives as discussed in previous sections.

Definition: A security mechanism is an entity or procedure that imposes, typically, some part of the security policy.

1.3.2 Security Standards

Security standards have been established to cover: (i) management practices, and (ii) the overall architecture of both security mechanisms and services.

Various international organizations and institutions have participated in the development and promotion efforts of security standards. Nevertheless, the most important organizations are the following:

- **Internet Society (ISOC)** is a professional membership society and includes the Internet Engineering Task Force (IETF) and the Internet Architecture Board (IAB). There is no doubt that ISOC provides leadership in addressing and handling issues that confront the future of the Internet.
- **The International Telecommunication Union (ITU)** is a United Nations agency in which governments and the private sector coordinate global telecom networks and services.
- **National Institute of Standards and Technology (NIST)** is a physical sciences laboratory and a non-regulatory U.S. federal agency that aims to promote private sector innovation and U.S. industrial competitiveness. It is used by organizations all around the globe.

The International Organization for Standardization (ISO) is a global nongovernmental federation of national standards bodies promoting the development of standardization and related activities with the aim to facilitate the international exchange of goods and services, and to develop cooperation in the fields of economic, intellectual, technological, and scientific activity.

1.4 Security risk management concepts

IT security management is the formally recognized process for securing critical assets cost-effectively. The main steps of the IT security management are:

1. Firstly, to determine a clear view of the IT security objectives and the overall risk profile of the given organization.
2. Secondly, to perform **risk assessment** for each critical asset in the organization that should be protected. It must provide the information necessary to decide what management, operational, and technical controls are required to: (i) reduce the risks identified to an acceptable level, or (ii) accept the resultant risk.

Figure 1-4: IT Security Management Overview [1]

According to the conceptual framework ISO13335 managing security, **IT security management** is defined as *a process used to achieve and maintain appropriate levels of confidentiality, integrity, availability, accountability, authenticity, and reliability.*

This process, from ISO27005 and ISO13335, is illustrated in Figure 1-5 particularly focuses on the internal details relating to the risk assessment process, since IT security risk assessment process is a key part of an organization's overall management plan and thus it should be incorporated into the broader risk assessment of all the organization's assets and business processes. On top of that, IT management is a cyclic process and must be undertaken continuously to maintain control over the volatility of the IT technology and the risk posture of organizations and institutions.

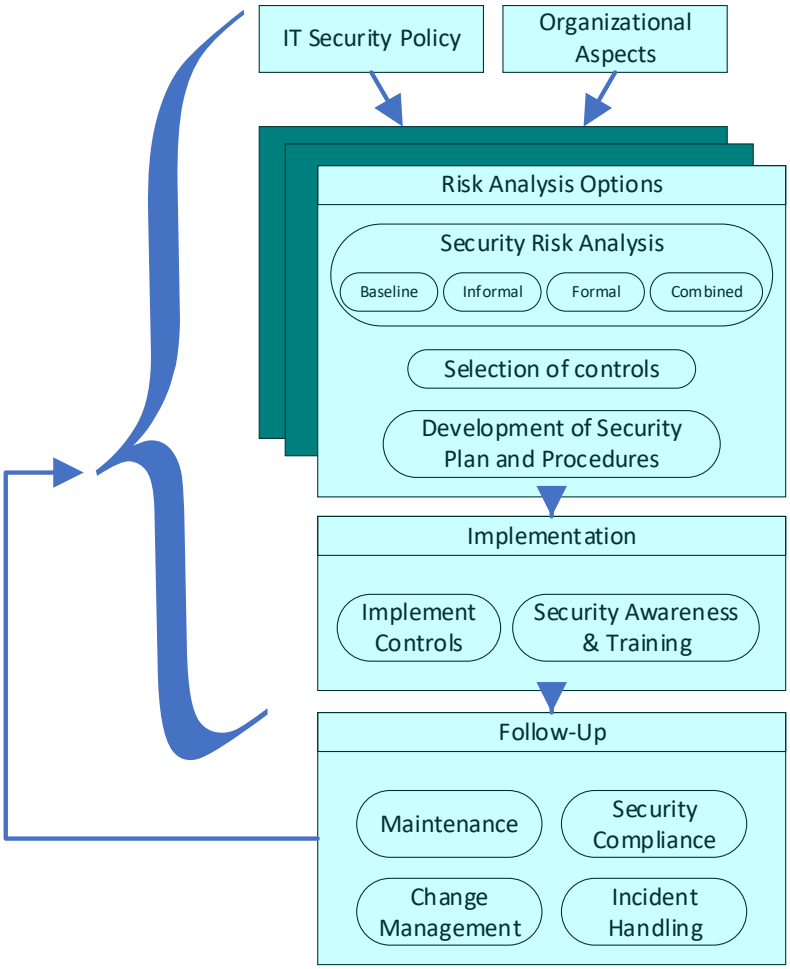


Figure 1-5: IT Security Management Process Overview [1]

1.4.1 The Plan-Do-Check-Act Process Model

This process model composes a security standard, based on the security risk management process in ISO 27005, for managing information security. This process should be constantly repeated and result in the proper management of the security needs of the interested parties. It consists of the following steps:

- **Plan:** Determine security policies, goals, processes, and procedures; perform risk assessment; develop risk treatment plan with suitable and efficient selection of controls or, otherwise, acceptance of risk.
- **Do:** Implement the risk treatment plan.
- **Check:** Monitor and maintain the risk treatment plan.

- **Act:** Maintain and improve the information security risk management process in response to identified changes, incidents, or received review.

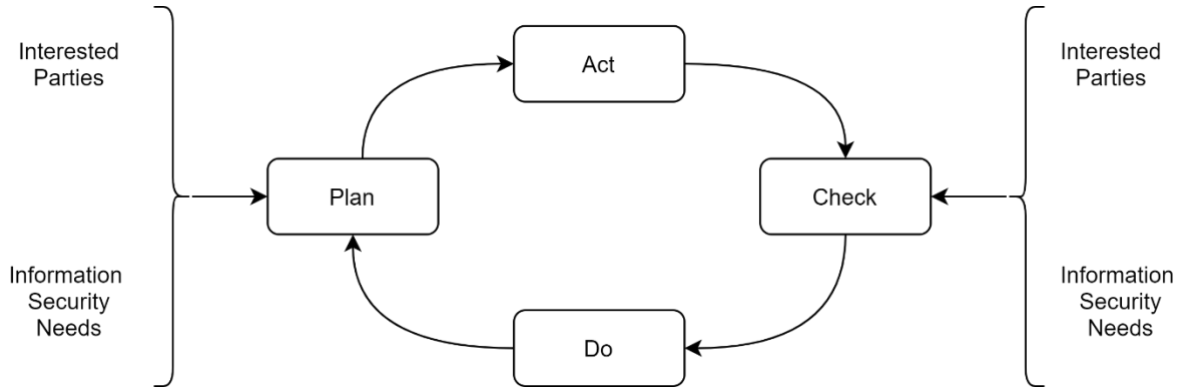


Figure 1-6: The Plan-Do-Check-Act Process Model [1]

1.4.2 Security Risk Assessment

The **security risk assessment** is the key risk management component of the IT security process. This process is critical in order to assure that available resources will be developed effectively to efficiently secure IT system without wasting organization´s time and money. The main idea of the security risk assessment is that firstly, each asset is assessed, and afterwards, each potential risk to this particular asset is evaluated. The result will be that the greater the assigned risk is, the most effort should be put. However, this practise is clearly unfeasible in practice for the following reasons: (i) firstly, the time and effort needed to perform the risk assessment is neither achievable nor cost effective, even for big and well-resourced companies, and secondly, the quick growth of changes in ICT technology and the threat environment implies that such assessments will be outdated when they are finalized.

In addition, it is crucial the decision whether the level of risk is appropriate to accept or not. Ideally, the target would be to completely eliminate all potential risks, but this is clearly impossible. Therefore, in practice, the goal is that safeguards (i.e., organizational resources) may be deployed to reduce risks proportional to the potential costs to the organization if certain risk occurs. For that, the likelihood of the risk's occurrence should also be considered. Thus, setting the acceptable risk level is basically prudent management taking into account a reasonable resources cost based on existing personnel resources, money, and time of the

organization. The risk assessment process´s goal is to support management with necessary data towards reasonable decisions on where should deploy available resources (i.e., money, time and human resources).

The range of size of organizations significantly varies from small businesses to big corporations, and therefore there is also a wide spectrum of formal standards for performing efficient IT security risk assessment based on the organization´s needs. ISO 13335 reports four basic approaches for the identification and mitigation of potential risks to an organization's IT infrastructure, as following: (a) Baseline approach, (b) Informal approach, (c) Detailed risk analysis, and (d) Combined approach.

The decision of the suitable approach for a particular organization will be determined by the available money, time, and personnel resources. On top of that, the selection should be made based on a primary risk analysis on high-level within the organization which will determine the value of the IT systems and their criticality for the organization's objectives. In addition, legal constraints may also enforce particular approaches for the IT security risk assessment. This information should be established in the phase of the organization´s IT security objectives, strategies, and policies development.

It is worthwhile to highlight that the combined approach is considered the most cost effective by the ISO13335 considers in most circumstances, for most organizations, and thus its use for performing risk assessment is highly recommended. This approach provides, as quickly as possible, reasonable levels of protection, and then examines and adjusts the protection controls developed on key systems over time, combining elements of the other three approaches, namely baseline, informal, and detailed risk analysis approaches.

A set of international and national standards broadly agree on the typical risk assessment process, as illustrated in the following Figure 1-7:

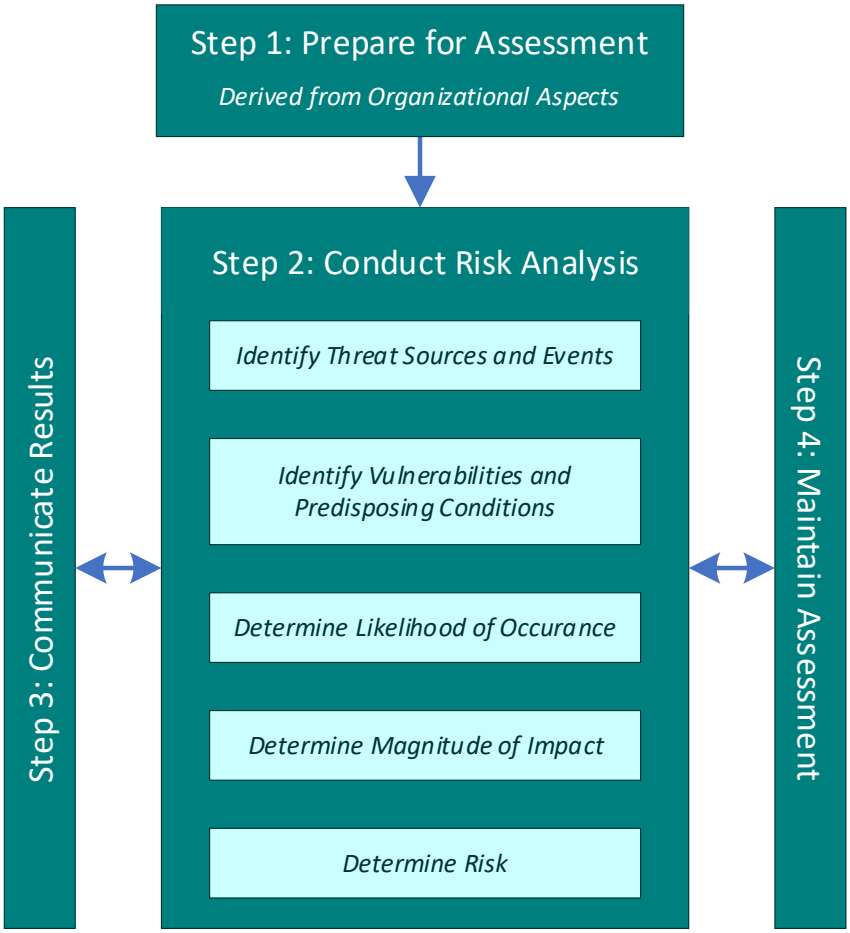


Figure 1-7: Process of Risk Assessment [1]

1.4.2.1 Establishing the Context

The initial phase of the risk assessment is to establish the context or characterize the system to define the fundamental parameters for performing the risk assessment.

First of all, this process examines the security objectives, as well as evaluates the organization´s wide risk exposure. The process investigates the connection between a particular institution and the broader sociopolitical environment in which it operates. As a result, it is recognized that organizations are targeted by attackers based on their general operational functions. For instance, government or banking industries are anticipated to be at higher risk compared to industries such as farming. Actually, a number of classifications has been established by international organizations indicating which industries are more vulnerable, and thus they require a prudent management plan to mitigate possible identified risks.

At the point of determining the broad risk exposure of the organization, legal and regulatory issues related to the organization and its mission must also be identified and reported. All identified elements contribute to: (a) the provision of a baseline for the risk exposure of the organization, as well as (b) an early idea of the resources that are required in order the organization to successfully manage these risks and conduct appropriate business.

Next step is the determination of the risk appetite of the particular organization and the risk level which is acceptable by organization´s senior management. These decisions are not just IT specific, on the contrary, they strongly depend on the type of organization, reflecting the broader management attitude of the organization in terms of how it conducts business.

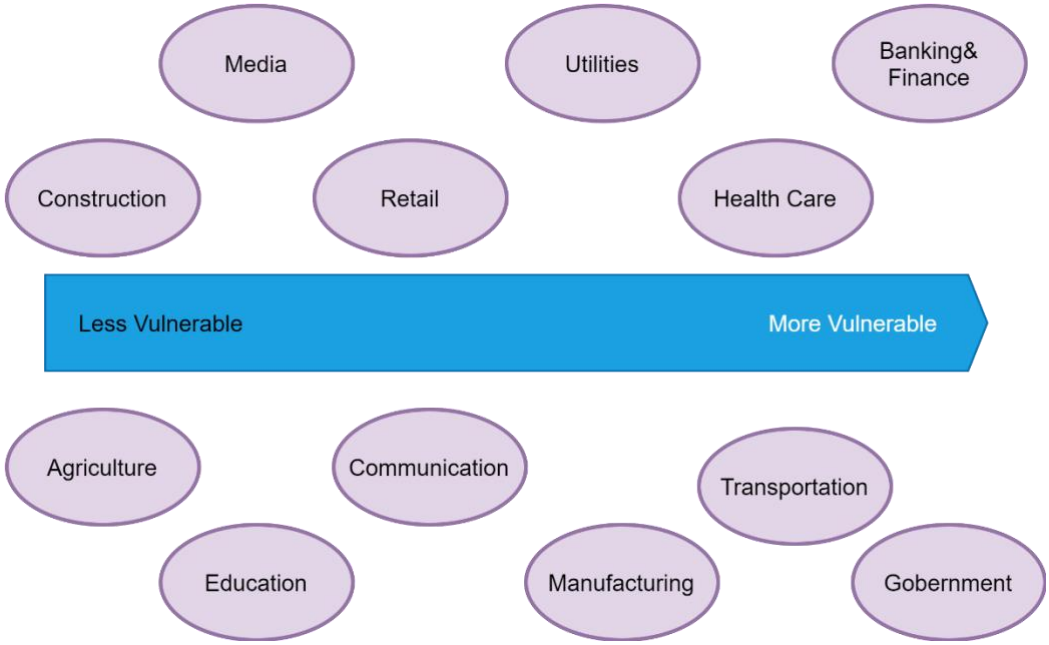


Figure 1-8: Generic Classification on Organizational Risk Context [1]

1.4.2.2 Assets Identification

Asset identification and evaluation constitute the final step of this first process in the risk assessment. An organizational **asset** is considered everything with a significant value to the organization and, in particular, to the fruitful achievement of its business objectives, and thus needs to be secured. Although it is desired aim to identify every organizational asset, this is not feasible in practice and thus it is more realistic to consider the ones that contribute the most to achieving the organization's main goals and functions, and whose compromise or loss

would cause a serious impact on the profitable operation of the organization. There are tangible or intangible assets, and it may include individuals that operate and maintain IT systems, documentation and training tutorials on this kind of systems, computer and communications hardware infrastructure, and software.

In most cases, the risk assessment is managed and performed by security experts, that they will not be probably familiar with the operation and structures of the particular organization. Thus, there is the need for the list of identified assets to be accompanied with descriptions of their use by, and value to, the specific institution. This can be provided by employees who work in the relevant areas within the organization.

1.4.2.3 Threat Identification

The following process in the risk assessment is to detect the potential risks and/or threats that the identified organizational assets are exposed to. It is worth to highlight that the terms threat and risk, although they have distinct meanings, in the current context, they are usually used interchangeably. In the following, some useful definitions are presented:

- **Asset:** A valuable system resource or capability that requires protection.
- **Threat:** A potential exploitation of a system vulnerability in certain asset, which may compromise its security and cause, in general, harm to the whole system or owner of this asset.
- **Vulnerability:** A weak point in the design, implementation, or operation and management of an organizational asset that a threat could exploit to compromise its security.
- **Risk:** The possibility for compromise or loss computed as the likelihood that an asset´s vulnerability is exploited by some threat, multiplied by the magnitude of harmful impact that this occurrence result to the asset's owner.

To identify potential significant risks to the identified assets, two fundamental questions should be addressed: 1) Who or what could cause harm to the particular asset? and 2) How could this happen?

Addressing the first of these fundamental questions requires the identification of the potential risks and/or threats to the identified assets. In general, anything that might prevent an asset from delivering proper levels of the key security objectives (i.e., **confidentiality, integrity, availability, accountability, authenticity,** and **reliability)**, can be considered a threat or potential risk. It is worth to mention that a single threat may target multiple assets, as well as one asset may be exposed to multiple threats.

1.4.2.3.1 Threat Modelling

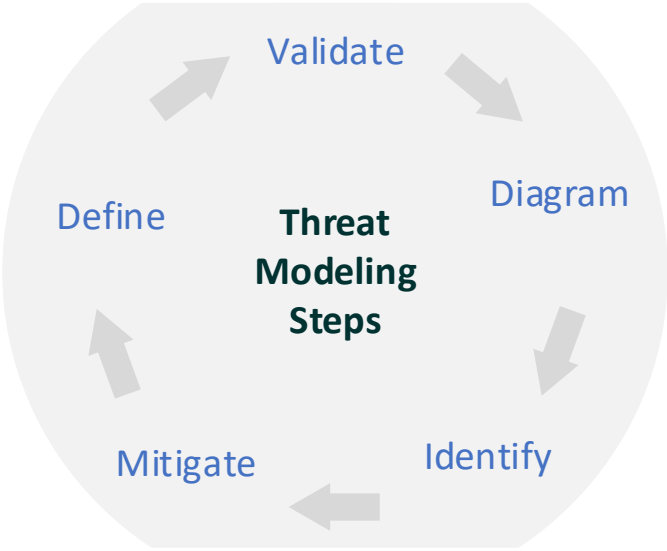


Figure 1-9: The 5 major threat modelling steps

The Threat Modelling is a cyclic process and should happen constantly, enabling to progressively refine the considered threat model and to further reduce potential risks. On top of that, it should be part of the routine development lifecycle.

As depicted in Figure 1-9, the five major threat modelling steps are to: (1) define system´s security requirements, (2) create an application diagram based on the identified requirements, (3) identify system´s threats, (4) mitigate the identified threats, and (5) validate that the identified threats have been successfully mitigated.

1.4.2.4 Vulnerability Identification

This process includes the identification of weaknesses and/or flaws in the systems or the processes of the organization that a potential risk or threat can utilize. Vulnerability identification contributes towards the determination of how applicable the threat can be to the organization and its importance. The vulnerability itself does not mean, certainly, harm to an organization´s asset. There must also be some threat that can exploit certain vulnerability.

Again, the result of this vulnerability identification step should be a list of vulnerabilities and threats, along with descriptions of how and why they might take place in order the security experts to determine their relevance to the particular organization and perform a successful risk assessment.

1.4.2.5 Analyze Risks

After identifying the major organizational assets and the potential vulnerabilities and threats to these assets, the determination and categorization of the risk level that each of these pose to the regular operations of the organization is the next step of the risk assessment. On top of that, risk analysis contributes to management to evaluate these risks and determine how best the managers could treat them. The risk analysis process involves:

1. To specify the occurrence likelihood of each identified threat to an asset.
2. To determine and evaluate the impact to the organization in case of this threat occurrence.
3. To assign an overall risk for each threat combining the information from steps 1 and 2.

The risk is represented as following:

Risk = (Probability value that threat happens) x (Monetary cost to organization)

where the likelihood of occurrence is specified as a probability value, while the impact to the organization is represented as a monetary cost.

In practice, the risk represents the value that a threatened asset has for the organization, and thus managers are able to specify a reasonable budget to spend in order to diminish the likelihood of its occurrence to a level acceptable by the particular organization´s management. However, in most cases, it is particularly challenging to define accurate probability values, and realistic monetary costs, and thus most risk analysis approaches make use of qualitative ratings, rather than quantitative. Therefore, the aim is then to classify the risks values so as to identify the threats that the most urgently need to be managed and mitigated, rather than to compute absolute and accurate risk values.

1.4.2.6 Analyse Existing Controls

All the management, operational, and technical procedures contributing to reduce the organization´s exposure to risks constitute the security **controls**. In particular, their aim is to prevent a threat source to exploit organization´s vulnerabilities. Existing security control should be identified by key organizational staff using checklists before the determination of the likelihood of a threat to occur.

After the identification of the existing security controls, the **likelihood** of occurrence for every identified threat should be specified. Typically, the likelihood is described qualitatively, see Table 1-2. The selection of the best descriptions and tables is, typically, identified at the beginning of the risk assessment process (i.e., context establishment). There is no doubt that exact and appropriate rating will be very uncertain and debatable, in most cases. This clearly reflects the ambiguity in the precise meaning and the uncertainty over the likelihood of a threat occurrence of the qualitative risk score ratings. Security experts and risk analysts consider the asset and vulnerability/threat identification descriptive details from the preceding steps in risk assessment and decide the most suitable risk score rating taking into consideration the overall risk environment and existing controls of the organization. Although it is fundamental that they keep notes of any uncertainty and debate arises during the selection of ratings process, eventually organizational management will take the business decision required towards the response to this rating. After all, the main objective of risk assessment is to support and guide organization´s management regarding the existence of certain risks, as well as towards their decision on how to respond most appropriately these

risks providing them with suitable and realistic information. Table 1-2 summarizes a typical categorization of risk likelihoods with brief descriptions.

Table 1-2: Risk Likelihood [1]

Rating	Likelihood	Description
1	Rare	Occurrence in very exceptional circumstances, also referred to as ``unlucky´´ or very unlikely.
2	Unlikely	Possible but not expected occurrence given organization´s security controls, recent incidents, and circumstances.
3	Possible	Might or might not occur at some time in the future. If it occurs, it might be challenging to control it because of external factors.
4	Likely	Expected occurrence in some circumstances.
5	Almost certain	Expected occurrence in most circumstances.

Having specified the likelihood of occurrence of a threat, the risk analysts should determine the threat consequence that suggests the impact on the organization of the certain threat in question eventuating. This is clearly distinct from the threat likelihood identification. For instance, although the likelihood of occurrence of a threat may be rare or unlikely, if the threat consequences on the are severe, then it undoubtedly creates a great risk to the organization and thus proper actions must be planned immediately. Again, similar to the likelihood ratings, the consequence is described in qualitative values, as shown in Table 1-3, and there is, typically, expected to be ambiguity when selecting the optimal rating to set.

On the contrary to the likelihood determination, the consequence determination should be made based on the organization's management judgment that know the organization's current practices and arrangements, rather than the risk analyst´s opinion. This is because it must be realistic and in relation to the overall impact on the organization should this particular threat occur, and not only on the affected system. Let´s consider that a fire occurred destroying completely an organization´s database. However, the impact of this occurrence on the organization might be a minor inconvenience (if all data stored in the database were duplicated to a different place), or a major disaster (if the database had the sole copy of a costumer´s confidential records).

Table 1-3: Risk Consequences [1]

Rating	Consequence	Description
1	Insignificant	Consequence of a minor security breach in a particular field within the organization. Impact may last few days, while the cost to address it is minor.
2	Minor	Consequence of a minor security breach in one or two fields within the organization. Impact may last less than a week, while the cost to address it can be generally covered by team or project resources.
3	Moderate	Limited systematic and ongoing security breach with an impact of up to 2 weeks. Management intervention will generally be needed, though may still be able to be addressed by team or project resources.
4	Major	Ongoing systematic security breach with an impact of 4-8 weeks. Senior management intervention will be needed and resources to be addressed. Expected substantial costs. Possible but not expected loss of business or organizational outcomes.
5	Catastrophic	Major systematic security breach with an impact of 3 or more months. Senior management intervention for the duration of the incident will be needed. Expected substantial costs. Expected significant harms to the organization like the loss of a business customer or the loss of the public or political confidence. Possible criminal or disciplinary action against organization´s personnel.
6	Doomsday	Multiple instances of major systematic security breaches with undetermined impact duration. Senior management will be needed to place the organization under voluntary administration or other major restructuring form. Expected loss of business and unavoidable failure to meet organizational objectives and functions. Costs are expected to result in annual losses for a couple of the following years.

Having identified the likelihood of occurrence of a threat and consequence of each specific threat, the next step is to assess a **risk level** for this threat. This can be done using a table similar to Table 1-4 that maps the suggested values to a particular risk level. This kind of table is a qualitative equivalent of the ideal quantitative risk calculation indicating the interpretation of the assessed levels.

Table 1-4: Risk Level Determination and Meaning [1]

	Likelihood				
Consequences	**Rare**	**Unlikely**	**Possible**	**Likely**	**Almost Certain**
Insignificant	L	L	L	M	H
Minor	L	L	M	H	H
Moderate	M	M	H	H	E
Major	H	H	E	E	E
Catastrophic	H	E	E	E	E
Doomsday	E	E	E	E	E

Risk Level		Description
Low (L)		Is manageable by routine procedures.
Medium (M)		Is manageable by existing particular monitoring and response procedures.
High (H)		Requires senior management and team leader attention for appropriate management and planning. Adjustment of controls within the existing resources may be required.
Extreme (E)		Requires executive management attention for detailed research and management planning. Substantial adjustment of controls and significant costs (possibly exceeding initial forecasts) are expected to manage the risk.

The outcomes of the performed risk analysis are recorded in a **risk register**, providing senior management and team leaders with efficient information towards proper decisions as how to best possibly cope with the risks identified. The risks should be sorted in decreasing order of level including details of how such values were determined, such as proper justification, and supporting evidence used.

Once the details of risk register have been documented, management and team leaders should whether any actions in response should be taken, taking into consideration the organization´s risk profile and its preparedness to accept a certain level of risk, as defined in the initial phase of the risk analysis, namely establishing the context. Typically, organization´s management would accept without performing any further actions all threats with assigned risk levels below the acceptable level, while for threats with risk higher than this, further treatment actions will be required.

1.5 Documented security procedures and guidelines

1.5.1 Security Control

A security control, also known as security safeguard, or countermeasure, contributes towards the respond and elimination of an organization's IT systems identified threats. As defined in [1]:

"An action, device, **procedure**, or other measure that reduces risk by eliminating or preventing a security violation, by minimizing the harm it can cause, or by discovering and reporting it to enable corrective action."

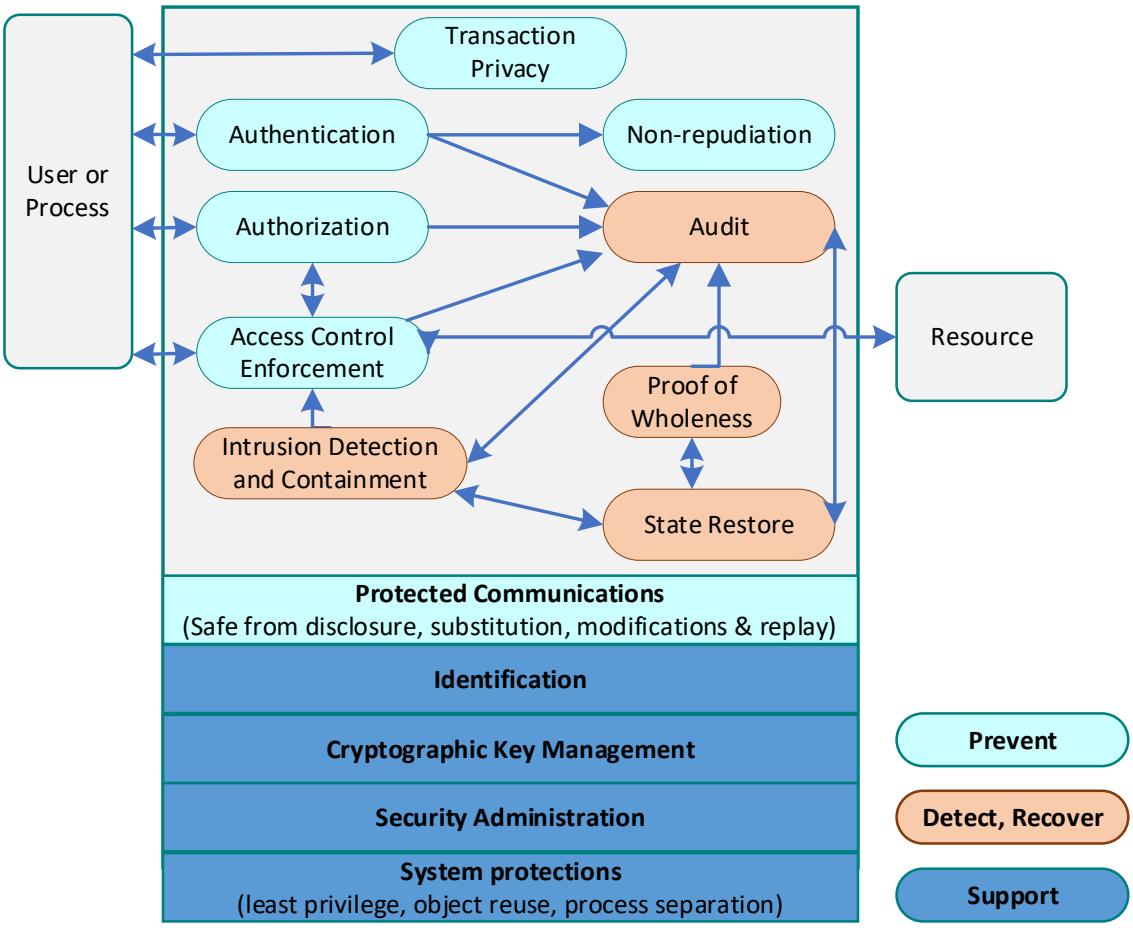


Figure 1-10: Typical technical security controls [1]

A number of national and international standards provide lists of security controls, including NIST SP 800-53, ISO 27002, ISO 13335, and FIPS 200. Although, there is generally a broad agreement among all standards about the detailed lists of the typical security control types that should be used within the organizations, ISO 27002 is considered as the master list of

security controls. Table 1-5 provides a common list of security controls families within the management, operational, and technical classes, based on the Table 1 in NIST SP 800-53).

Table 1-5: NIST SP800-53 Security Controls

Family of Security Control	**Class**
Planning	Management
Risk Assessment	
Security Assessment	
Security Authorization	
Program Management	
Services and System Acquisition	
Identification & Authentication	Technical
Access Control & Authorization	
Audit & Accountability	
System & Communications Protection	
Maintenance	Operational
Configuration Management	
Awareness & Training	
Incident Response	
Contingency Planning	
Personnel Security	
Media Protection	
System & Information Integrity	
Physical & Environmental Protection	

It is worthwhile to highlight that every control class contains a long list of specific controls that can be selected based on the nature and resources of the given organization.

Typically, a combination of these security controls is combined to achieve an acceptable security level in the particular industry or government. The overall risk profile, resources, and capabilities of the organization will lead to an appropriate and prudent selection of security

controls. Afterwards, the selected ones should be implemented across all the organization's IT systems, with suitable adjustments for addressing broad requirements of specific systems.

NIST SP 800-18 standard suggests that in certain cases, appropriate adjustments may be required to consider when selecting appropriate security controls. For example, security controls for wireless networks are not applicable for the use of cryptography. In addition, If the whole organization is managed centrally and not by the managers or team leaders of specific systems, then control changes would require to be agreed to and managed centrally. On top of that, scalability issues should also be taken into account as security controls may vary in size and complexity in relation to the organization employing them. Finally, security controls must always be adjusted and aligned to the outcomes of the risk assessment of organization´s systems.

1.5.2 Security Compliance

The checking of **security compliance** consists an audit process which aims to verify compliance with the security plan of the organization reviewing its security functions and processes. In general, the audit may be conducted by internal or external personnel verifying the suitability of the created policies and plans, the suitability of the chosen controls, and the correct maintenance of used controls based on specific checklists.

In any case, this audit process should be conducted on: (i) once they are implemented, new organizational IT systems and services; and (ii) periodically, on existing IT systems and services, frequently as part of a broader audit of the organization, or at any time that changes to the security policy of the organization are happening.

1.6 Referencing

[1] "Computer Security: Principles and Practice", 4/e, by William Stallings and Lawrie Brown, 2018, Pearson

[2] Computer Security Art and Science", 2nd Edition, by Matt Bishop, November 2018, Addison-Wesley Professional, ISBN: 9780134097145

[3]Computer Security Fundamentals, by Chuck Easttom, 2019, Pearson Education, Inc.

2 Security Engineering: Introduction

Author(s): Felipe Gil Castiñeira
Cristina López Bravo
René Lastra Cid

2.1 Implement and manage an engineering life cycle using security design principles

Systems engineering is an interdisciplinary technique that uses an iterative method to transform consumer expectations into the concept of a system, its architecture, and design, resulting in an optimized operating system. It is applicable in the life cycle [4].

Multiple, interconnected security approaches covering the personnel, technologies, and organizational facets of information systems are needed to secure information and systems against the broad spectrum of threats. This is because of the increasingly collaborative nature of different systems and networks. No system can be adequately protected without simultaneously securing all interconnected systems.

Security architects must realize that unless they are given the rare ability to plan, create, and incorporate an infrastructure from the ground up, “ripping and replacing” what is already in place in lieu of what they are making, the technologies they are dealing with and incorporating into the architecture would have its own issues, questions, and challenges.

The following are the features of security architecture. The security architecture has its own security approach which makes up its own views and perspectives. Non-normative flows across networks and between programs are handled by security architecture. The security architecture produces its own regulatory flow array and incorporates special components in the design. Security architecture integrates one-of-a-kind, single-purpose components into the design, necessitating a distinct collection of skills and competencies among business and IT architects.

The device protected by multiple overlapping security mechanisms does not leave the system vulnerable if one of them is malfunctioned or circumvented. The protection of information security can be successfully protected by user training and learning and through well develop policies and procedures.

The Common Criteria include a systematic methodology to capture security requirements, to document and validate security capacity and to promote international cooperation in IT security. The use of Standard Standards "safety profiles" and "security goals" helps in the development of IT security-related products and systems. The Common Criteria

methodology's rigor and repeatability allow for a detailed specification of user protection requirements. Safety goals provide system integrators with essential details for product acquisition and stable IT system deployment.

Systems engineering models and processes are usually organized across the notion of a life cycle. ISO/IEC 15288:2008 covers applications and lifecycle stages of international devices engineering. It describes a number of four processes: technology, project, contract and business.

ISO/IEC 21827:2008 [6], specifies the essential characteristics of a company's safety engineering process. It captures common industry practices. The model provides a standard metric for safety engineering practices and includes:

- Activities such as development, operation, maintenance, and decommissioning (entire lifecycle);
- Management, organizational, and engineering activities across the entire organization;
- Connections with other fields: device, hardware, applications, measurement engineering at the same time and human elements;
- Management, operation, and maintenance of the system;
- Examples of interactions with other organizations: system management, acquisition, evaluation and accreditation.

Five stages of life cycle preparation commonly used in the NIST SP 800-27 Rev A standard [7].

- **Initiation**. The system requirement is defined and the purpose of the system is recorded. One of the tasks is to conduct an impact assessment under FIPS-199.6.
- **Development/Acquisition**. Determination of safety criteria, inclusion of safety requirements in the specifications, and system acquisition are the tasks to be performed.
- **Implementation**. Installing/activating controls, security testing, certification, and accreditation are all examples of implementation activities.
- **Operation/Maintenance**. Maintenance functions include operations and safety management, performance assurance, monitoring and inspection.

- **Disposal.** This process includes the transfer, archiving, discarding or elimination of information, as well as the cleaning of the media

SP 800-27 principles:

Security Foundation

- Establish a sound security policy;
- Safety is part of the overall system design;
- Determine physical and logical security boundaries;
- Developers must be trained in secure software development.

Risk Based

- Minimize risk as much as possible;
- External networks are vulnerable;
- Identification of trade-offs among risk reduction and cost increases or operating productivity reduction;
- Apply system security measures to meet the organization's security objectives.
- Safeguard encoding, transmission and storage information;
- Customized products
- Protect from threats of all sorts.

Ease of Use

- Establish open standard encryption;
- When developing security requirements, use common language;
- Create a stable and rational infrastructure update mechanism to allow for routine implementation of new technology;
- Aim for operational simplicity. Boost Your Resilience;
- Use layered encryption;
- Create and run an IT structure that minimizes harm and is robust in the face of adversity;
- Assure that the infrastructure is stable in the face of anticipated challenges, and that it will continue to be so;
- Vulnerabilities should be limited or contained.
- Isolate mission-critical tools (e.g., records, processes, etc.) from public access networks.

- Independent operating systems and network infrastructures using boundary mechanisms.
- Create and enforce auditing systems to identify improper usage and assist with incident investigations.
- Ensure adequate readiness by developing and testing preparation or emergency response protocols. Vulnerabilities should be minimized.
- Make an effort to keep it simple.
- Reduce the number of device components that must be trusted;
- Use the least amount of privilege possible;
- Don't use authentication measures that aren't absolutely required.
- Ensure that a system's shutdown or disposal is done safely;
- Recognize and avoid typical vulnerabilities and errors.

Design with Network in Mind

- Implement protection by using a mixture of physical and logical steps;
- Develop protection mechanisms to resolve various overlapping data domains;
- Authenticate users and systems;
- Maintain accountability by using distinct identities.

2.1.1 Examples of safe development life cycle frameworks

- The **Cisco Secure Development Lifecycle**
- Microsoft's reliable computer security development life cycle
- The Maturity Construction Safety Model V is a standard set by the Centers for Medicare and Medicaid Services (CMS) (BSIMMV).

2.1.1.1 The Cisco Secure Development Lifecycle

The Cisco Secure Development Lifecycle (SDL) [8] is a framework for improving product resiliency and reliability.

Components of Cisco SDL are: product, security requirements, security architecture, secure encryption, secure inspection and vulnerability testing.

Product Security Requirements

There are two categories of product protection standards that should be addressed by products, internal requirements and market-based requirements.

The PSB shall also cover credentials and key controls, encryption requirements, the anti-spoofing feature, the integrity and tamper protection and the management.

MARKET-BASED REQUIREMENTS

Common criteria Certification, Cryptographic validation for devices with encryption capability, IPv6 certification, Department of Defense (DoD), Unified Capabilities Approved Products List, and North American Electric Reliability Corporation, Critical Infrastructure Protection are examples of product certifications that could be requested (NERC-CIP)

Third-Party Security

The incorporation of third-party applications, both proprietary and open source, in products is a common business activity. When third-party bugs are found, products and consumers can be affected. Cisco uses a centrally managed intellectual property registry to monitor devices that use third-party applications internally.

A set of tools are used:

- Cisco's security teams are immediately notified about suspected third-party program risks and vulnerabilities from a constantly maintained archive, allowing for fast investigation and mitigation.
- Cisco uses software to review source code and images to increase the quality and integrity of third-party repositories.

Secure Design

Based on established risk, consumer needs, and industry best practices, Product Security Requirements were compiled from internal and external sources. There are two categories of product protection standards that should be addressed by products:

- Considering security while designing
- Threat Modelling to verify the security of the design

SECURITY IN MIND DURING DESIGN

Internal security training plans enable production and test teams to deepen their security knowledge.

THREAT MODELING T O VALIDATE THE DESIGN'S SECURITY

Threat modelling is a method for determining and prioritizing security threats to a system. The engineers track the movement of data across a framework to define confidence boundaries and inflection points where data could be manipulated while modelling risks. Mitigation techniques should be implemented after possible vulnerabilities and risks have been established.

Secure Coding

SECURE CODING STANDARDS

Coding principles require programmers to adhere to a series of rules and instructions that are defined by the project's and organization's needs. Potential security flaws may arise as a result of coding and design errors. Developers will benefit from security training by learning safe coding standards and best practices.

COMMON SECURITY MODULES

Cisco-managed libraries are designed to reduce security vulnerabilities and enhance the capabilities of engineers.

Static Analysis

Cisco SDL defines the main security testers that identify source code errors in C and Java. A collection of checkers has been identified by internal review, field trials, and restricted business unit implementations to maximize identification of security concerns. Static analyses are performed by Cisco developing groups with enabled security checks and warnings and high-priority problems are solved

Vulnerability Testing

Vulnerability testing ensures that devices are thoroughly examined for security flaws.

The research is personalized to each product by determining:

- All protocols introduced in the product;
- By default available ports and services;
- Protocols, ports, and utilities that are likely to be used in a standard consumer setup;

Products are tested to assess their ability to withstand probes and assaults.

- Protocol robustness testing;
- Typical attacks and scans using open source and commercial hacker software;
- Web server scanning;

Developing a proper security testing strategy requires the use of a number of security tools from different sources that Cisco developers use to search for security flaws. Vulnerability testing is used to guarantee that the products are checked for security flaws.

2.2 Security models and architecture

2.2.1 Common System Components

Hardware, firmware, and software layers combine to provide computing services in modern computer systems. Any of the more common system modules, as well as their functions, should be understood by the security architect. The key elements would be discussed: processors, storage, peripherals, and the operating system.

Processors are the brains of a computing machine, running calculations as well as solving problems and carrying out device functions. Knowledge is stored on recording systems for both long and brief periods of time. Peripherals (such as routers, printers, and modems) are machines that either input or collect data from processors. The operating system (OS) acts as the glue that keeps all of these parts together, as well as the framework for programs, utilities, and the end-user.

Security tasks are spread through these modules when needed to ensure that the device can successfully protect information assets.

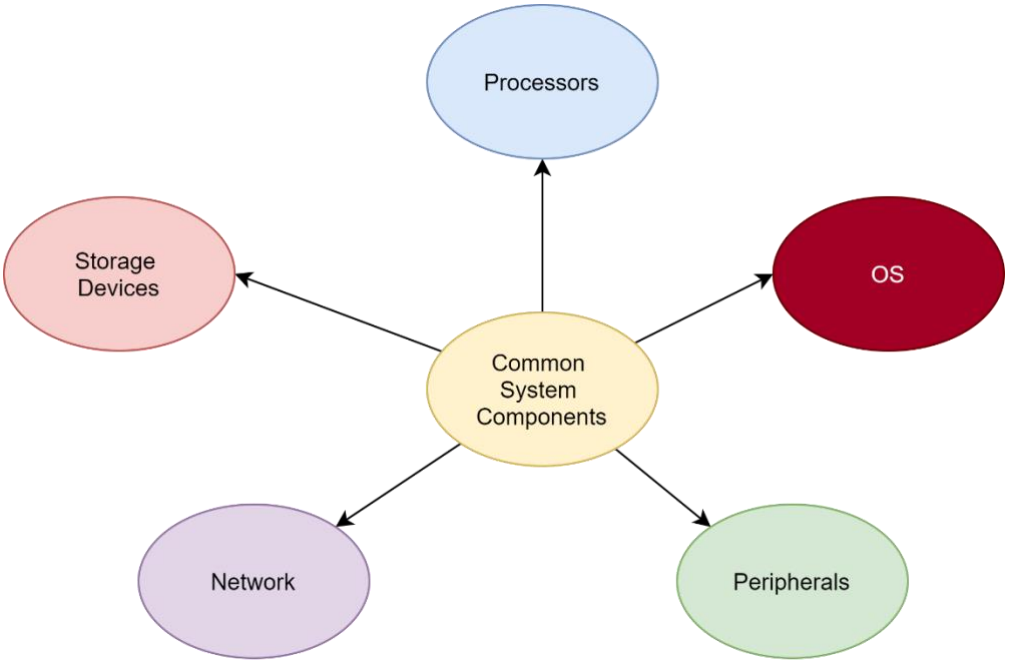


Figure 2-1: Common System Components [4]

2.2.1.1 Processors

Processors must have the following main features in order to resolve security issues on various levels:

- Tamper detection sensors
- Cryptocurrency acceleration
- A physical mesh of battery-backed logic
- The ability to configure a system with safe boot capabilities
- On-the-fly encrypt and decrypt capability in a secure memory access controller
- Countermeasures for static and differential power measurement (SPA/DPA)
- Smart card UART controllers

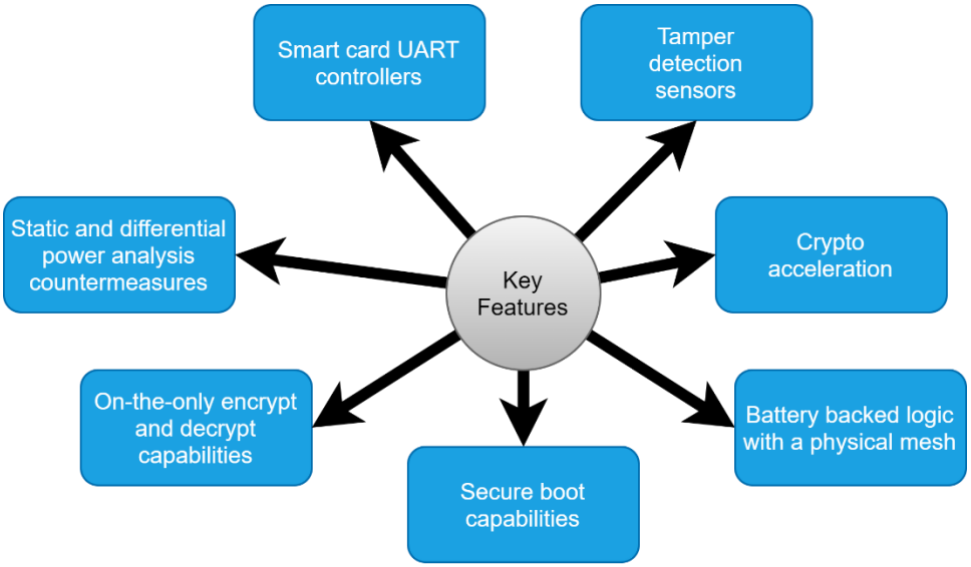


Figure 2-2: Processor Key Features [4]

2.2.1.2 Memory and Storage

The memory management is at the heart of the machine architecture. Memory is used to preserve information properties. Memory is used to run programs. Any system's security infrastructure relies heavily on memory. Each of the major types of memory necessitates a unique approach to security.

❖ Memory Protection

The following are the three most important memory protections:

- **Segmentation**: The division of a computer's memory into segments is known as segmentation.
- **Paging**: Paging partitions the memory address space into pages of equal size. A page table is a table that converts virtual memory to physical memory. Since each new file can be allocated from anywhere in physical memory, page tables make it easy to assign extra memory. It's impractical for a program to navigate a website that hasn't been assigned to it directly. Each memory address leads to a page reserved for that program or causes a page fault interrupt. From the program's point of view, unassigned pages and pages assigned to any other application do not have any addresses.

- **Protection keying**: Protection keying is a process that separates physical memory into blocks of a given size. A security key value is often assigned to each operation. On accessing memory, the hardware verifies whether the security key of the actual process matches the value associated with the given memory block; otherwise, an exception is thrown.

Additional memory management strategies for computer security include: executable space protection and address space layout randomization (ASLR).

ASLR entails rearranging the locations of a program's main data areas at random. The low probability of an intruder guessing the coordinates of randomly placed places is the basis for address space layout randomization. The search room is expanded, which improves security.

2.2.2 Enterprise Security Architecture

The Enterprise Security Architecture (ESA) focuses on a conceptual framework for a range of security utilities and applies the core elements of the information security system in the organisation. ESA's primary goal is to create a long-term vision for security services in the organization, and its primary goal is to establish goals for security service creation by providing insight into the information security program's preparation.

Figure 2-3: ESA Architecture [4]

2.2.2.1 Objectives

Internal skill sets, permits, and arrangements may be leveraged to reduce the need for preparation and personnel augmentation. It allows for a scalable response to current and potential challenges, as well as core role requirements. The architecture's implementation should be adaptable in order to provide protections and countermeasures against existing and emerging threats. It must also be adaptable in order for the organization's key applications to perform and fit properly.

As a consequence, an infrastructure that embraces and interacts with an appropriate security policy, reliable security, and high-quality security program should be the end product.

2.2.2.2 Benefits

ESAs aim to: Provide IT architects and senior management with guidance; Establish a future-state infrastructure framework for the defense world that is based on a small number of security services; Security protocols and guidelines should be supported, enabled, and expanded; Describe defense strategies in general;

Make use of industry guidelines and templates to ensure that current security practices are followed; Establish the security framework for technology; Provide an understanding of the effect of progress and deployment in other domains on security posture (better, worst, no change); Manage IT solution risk continuously across the project; Introduce common and reusable security services to save money and increase flexibility; Create a stable system for end-of-life solutions and dismantling as necessary.

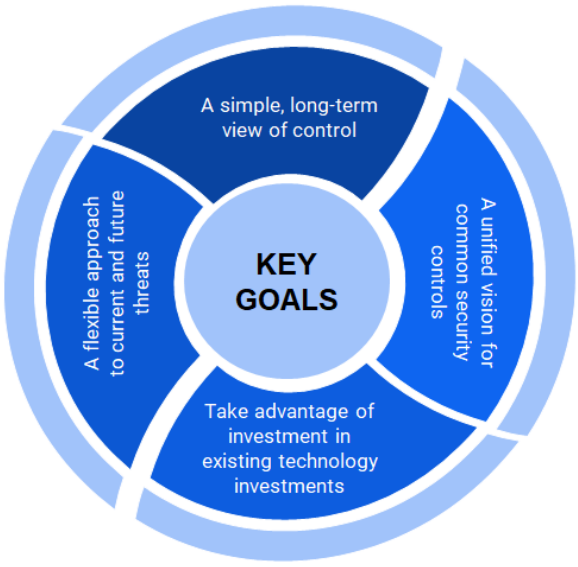


Figure 2-4: ESA Key Goals [4]

2.2.3 Common Security Services

A variety of security roles can be used as the basis for enterprise-wide security services. The bulk of ESAs make a distinction between various types of facilities. Below is a taxonomy of utilities that can be used as building blocks in ESAs.

- **Boundary Control Services**. Using choke points, border control systems reinforce security control zones by isolating entry points from one zone to another by providing a set of common points for accessing or transmitting information.
- **Access Control Services**. These terms apply to how and when data will flow from one collection of systems to another. Boundary control mechanisms improve security control areas by isolating entry points, also known as choke points, from one zone to another. They offer a collection of common points for obtaining access to or exchanging data between security zones. Some of the mechanism used to secure higher trust sensitive objects from the lower trust sensitive assets are stable networking appliances, firewalls, boundary-routers, proxies and other boundary networks.
- **Integrity Services** include: content filtering, antivirus, whitelisting, file integrity services and intrusion protection programs.

- **Cryptographic Services** can be adopted and reused by a wide range of systems that create a public key infrastructure (PKI). Popular encryption and hashing services, methods and technologies may also be included.
- **Audit and Monitoring Services** center on secure processing, review of audited events through centralized logging and management. It also focuses on events through intrusion detection systems (IDS) and other services.

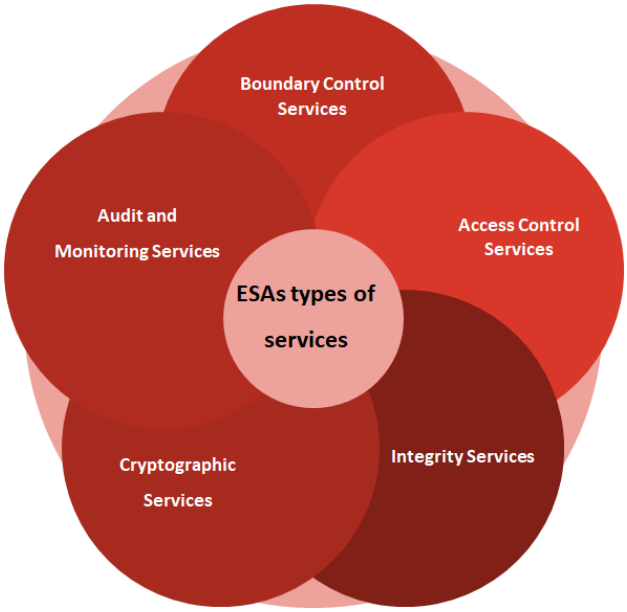


Figure 2-5: Common Security Services [4]

The following questions will assist you in comprehending the complexity:

- In contrast to the world in which they operate, how are intelligence properties safeguarded?
- To access the information asset, what degree of authentication is needed?
- Is there a gap between accessing the asset over a secure network and accessing it over an untrusted network? Are there any conditions that are identical within the internal networks?
- What protections will be in order to guarantee confidentiality? Is there a difference in the appropriate degree of secrecy based on when or how the asset is accessed?
- What safeguards must be in place to ensure the availability of secure resources? Will access restrictions have a favourable or unfavourable effect on them?

- Are there any properties that need a high level of integrity? How will asset confidentiality be protected if many organizations have access to them?
- When the information properties can have such disparate characteristics, how will the architect know how to implement these trade-offs?

2.2.3.1 Common Architecture Frameworks

This is a framework that you can use to create a wide variety of architectures. It defines a method to simplify the creation of the architecture. Often includes a list of criteria and recommended operating procedures. They may provide details on the manufacturer's products, modules or parts that meet the standards. The most commonly used design frameworks are listed below:

2.2.3.1.1 Zachman Framework

Both organizations involved in the creation of the architecture will communicate and collaborate using Zachman Framework. It establishes a conceptual framework for combining diverse viewpoints, such as plan, concept, and construction. The Framework is a conceptual mechanism for defining and organizing the descriptive representations that are essential in business management.

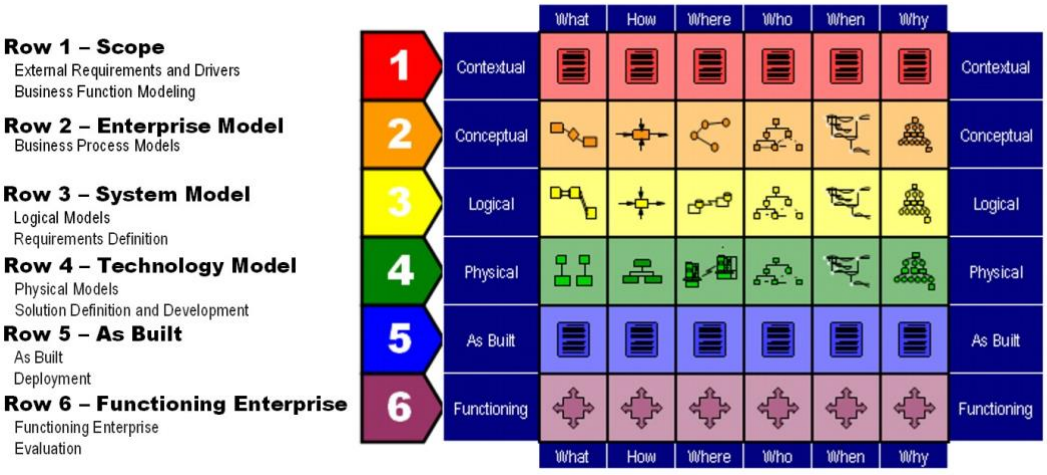


Figure 2-6: Common Security Services. https://commons.wikimedia.org/wiki/File:Simplification_Zachman_Enterprise_Framework.jpg. Al Zuech, Director, Enterprise Architecture Service at the US Department of Veterans Affairs., Public domain, via Wikimedia Commons

2.2.3.1.2 (SABSA) Framework

Sherwood Applied Business Security Architecture consists of a complete lifecycle for security architecture design, evaluating market requirements to create a chain of traceability. It provides a six-layered architecture, representing a perspective on the design, development and use of the target system.

2.2.3.1.3 The Open Group Architecture Framework (TOGAF)

TOGAF defines an architecture content framework (ACF) that defines standard building blocks and components, and a series of reference models.

An Architectural Development Method (ADM) focused on four architectural realms that defines the step-by-step mechanism used by TOGAF architects:

- Business architecture which specifies the organization's organizational strategy, policy, structure, and core business processes.
- Data architecture is a concept that refers to the arrangement of an organization's conceptual and physical data properties, as well as the data storage tools that go with them.
- Applications architecture is a model for the different applications that will be implemented, their interactions, and their connections to the organization's key business processes, as well as mechanisms for resources to be exposed as enterprise functions for integration.
- Technical architecture, also known as information architecture, refers to the hardware, software, and network infrastructure used to deliver key, mission-critical applications.

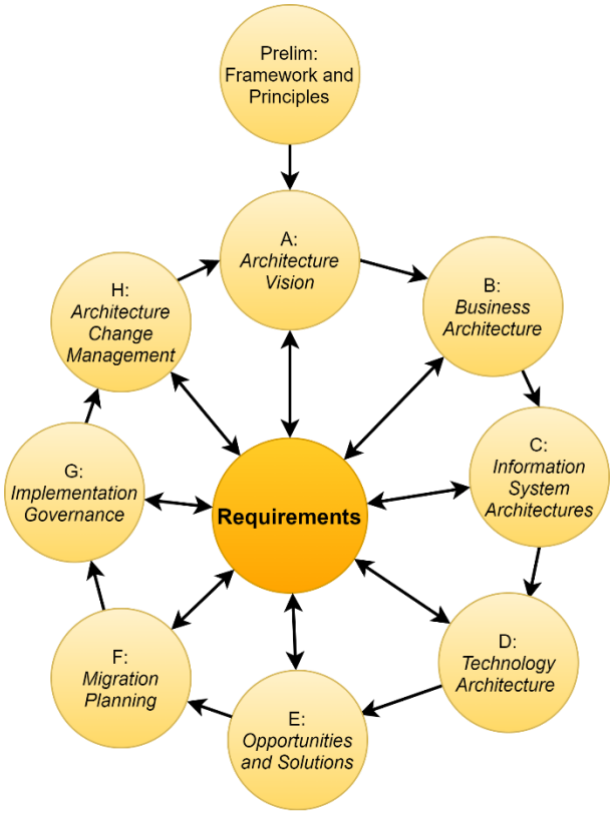


Figure 2-7: TOGAF architecture [4]

2.2.3.1.4 IT Infrastructure Library (ITIL)

ITIL is a set of operating practices that guide IT infrastructure and information management operations. Describes the organizational framework and capability needs of an IT organization. It also describes the set of operating procedures and practices that drive IT operations and infrastructure. Continuous service development; service operations, service planning, service architecture, and service transition, and are the five main practices or tasks of ITIL v4.

- Service Strategy The strategy describes the selection of programs being introduced or going to be implemented. In general, success or loss is assessed by customer service ability. For any other ITIL operations, the operation strategy introduces the specifications.
- Service Design. ITIL focuses on structures and architectures for management that direct or restrict design.

- The primary goal of Service Transition is to translate designs into operating services using a common project management framework. It's also in charge of monitoring improvements to current programs.

This ITIL component gives input on all other facets of service management, based on the need for improvement.

2.2.4 Types of Security Models

- **State Machine Model.** It explains how a system behaves when it transitions from one state to another. The goal is to specify what acts are tolerated at any given time in order to maintain a safe state.
- **Multilevel Lattice Models.** A non-interference model's aim is to ensure that high-level behaviour (inputs) do not affect what lower-level users see (outputs).
- **Non-interference Models.** Concentrate about how information is shared between individual items and how it is or is not permissible. It's used to see if data were correctly safeguarded at all levels of the process. Covert networks are addressed.
- **Matrix-Based Models.** Focus about how communication is exchanged between individual items and how it is permitted or not allowed. Used to assess whether data is safe at all levels of the process.
- **Information Flow Models.** Describes rigid layers of subjects and objects, as well as simple guidelines that encourage or disallow connections between them depending on which layer they're in. Automates "least privilege" judgments based on a user's privilege period and the transaction's requirement within that span.

2.2.5 Examples of Security Models

2.2.5.1 Bell-LaPadula Confidentiality Model

The Bell–LaPadula model is one of the first methods used in the development of modern stable computer systems. Avoiding disclosure when the model system transitions from one state to another is its main objective.

It explains the four fundamental elements that go into identifying the major actors and how they vary from one another. The active parties are subjects, while the passive parties are objects. Subjects are given clearances to specify what they are able to do (read, write, or read/write) to better decide what they are allowed to do.

The Bell-LaPadula model studies the guidelines to be followed if a subject receives a specific degree of authorization and mode of access. Based on whether the subject can read, write, or read/write objects, describe different characteristics. The subject will be able to read material from objects classified at a comparable level of confidentiality or lower levels, but not from objects classified at a higher level of confidentiality. Bell-LaPadula deals with confidentiality.

2.2.5.2 Biba Integrity Model

Biba is a multi-level grid-based model. It seeks to ensure that the quality of information is protected by corruption prevention. The model is designed to avoid the modification of objects by unauthorized subjects. Based on how confident they are, Biba applies competence standards to subjects.

2.2.5.3 Clark-Wilson Integrity Model

The Clark–Wilson model focuses on integrity at the transaction level. It is essential that the transactions of approved parties should be reviewed by another party until committed to the model system, to deter an authorized party from making unwanted adjustments.

Clark and Wilson suggested that the collection of measures within each transaction be specifically planned and applied in order to tackle internal coherence.

Clark-Wilson creates a subject-program-object linkage scheme to control all relationships between subjects and objects such that the subject does not have direct access to the object. The software allows authentication and identification of subjects and restricts all access to objects under its jurisdiction.

2.2.5.4 Brewer-Nash (The Chinese Wall) Model

The aim of this model is to avoid conflicts of interest where an object with confidential information linked to two opposing parties has access to it. The rule is that customers can not

access both a customer company and one or more competitors' private information. Compared to many others, it's an interesting model because access control laws adjust depending on the behaviour of the subject.

2.2.5.5 Graham-Denning Model

The Graham-Denning access control components are: a collection of objects, a set of subjects and a set of privileges. The Graham-Denning access control components are: a collection of objects, a set of subjects and a set of privileges. Subjects have two parts: a method and a domain. This model outlines eight basic privacy privileges, called commands, that subjects can use to affect other subjects or things:

- Build Object. This feature allows you to make a new object.
- New Subject Development. The opportunity to render a new subject.
- Delete Object. The right to remove a previously created object.
- Delete Subject. This functionality helps you to eliminate a current subject.
- Read Access Right. The opportunity to see what access rights are currently in effect.
- The freedom to assign access rights is known as the grant access right.
- Access Right to Delete. The power to revoke access rights.
- Pass Control Right. The capability of transferring access rights from one subject or entity to another.

2.2.5.6 Harrison-Ruzzo-Ullman Model

A number of uniform rights and a finite set of commands make up this model. It also recognizes cases in which a subject's right to such rights should be limited. Subjects are prohibited from accessing programs that are able to execute a specific command (e.g., give read access) in order to do so.

2.3 Controls and countermeasures based upon information systems security standards

Any security architecture should be closely scrutinized to ensure that it satisfies the documented specifications. Formal protection templates and verification procedures can be

issued to prove that the specification is accurate. Vendor products can be assessed using worldwide, uniform product assessment standards, or they can be reviewed and approved in their planned implementation environment before going into production.

Common Formal Security Models consist of a framework that establishes the rules that must be applied to support and execute a security strategy.

Evaluation Criteria: The goal of system assurance should be to ensure that a system adheres to a set of safety objectives.

Certification and Accreditation: Certification and Accreditation: the product or device will be checked during the certification process to determine if the documented criteria are fulfilled. The assessment criterion must be selected at the start of the procedure. The certification process tests the hardware, software and configuration of the device in a manufacturing setting, with the parameters known. The findings of the assessment become a basis for comparing the relevant safety conditions. If the certification is positive, the assessment is carried out in the next process. Management assesses the capability of a system in the accreditation process to fulfill organisation requirements.

Product Evaluation Models: Trustworthy evaluation criteria for computer systems is developed for classified systems and more general and global in character, such as Standard Criteria.

Trusted Computer System Evaluation Criteria (TCSEC): In order to assess, identify and pick the computer systems to be used for the collection, storing and recovery of confidential or secret information in military and government systems, Trusted Computer System Evaluation Criteria (TCSEC) are used. The priority was on the implementation of secrecy, with little emphasis on other security issues, such as integrity and availability. It affected other product appraisal requirements and continued to use some of its basic methodology and terminology. TCSEC is very descriptive and very prescriptive, and varies most from other assessment standards. TCSEC identified very basic categories of security controls that could be enforced on a stable and defined level of secure networks depending on the ability to enforce them, rather than providing a flexible collection of security standards.

Table2-1: Summary of Orange Book Evaluation Criteria Divisions [4]

Evaluation Division	Evaluation Class	Degree of Trust
A-Verified Protection	A1- Verified Design	Highest
B-Mandatory Protection	B3-Security Domains B2-Structured Protection B1-Labeled Security Protection	
C-Discretionary Protection	C2-Controlled Access Protection C1-Discretionary Security Protection	
D-Minimal Protection	D1-Minimal Protection	Lowest

Information Technology Security Evaluation Criteria (ITSEC): The customer or the manufacturer has the capacity to decide and have the goods (Target of Assessment or ToE) tested against this aim from a menu of potential specifications to a safety target. Two levels were provided: level of operation and level of assurance. It covered a broader variety of security needs, including standards for integrity and availability. As regards guarantee standards, ITSEC varied considerably from TCSEC (or E levels). Assurance is characterized as a degree of trust that an evaluator is not only satisfying the functional specifications of the product, but will also continue to comply with them. ITSEC specified six different insurance tiers, each of which was harder to achieve than the last.

Table 2-2: ITSEC Requirements [4]

E1	• A security target and informal architectural design must be produced • User/Admin documentation gives guidance on Target of Evaluation (TOE) security • Security enforcing functions are tested by evaluator or developer • TOE to be uniquely identified and to have Delivery, Configuration, Start-up and Operational documentation • Secure Distribution methods to be used
E2	• An informal detailed design and test documentation must be produced • Architecture shows the separation of the TOE into security enforcing and other components

	• Penetration testing searches for errors • Configuration control and developer security is assessed • Audit trail output is required during start up and operation
E3	• Source code or hardware drawings to be produced • Correspondence must be shown between source code and detailed design • Acceptance procedures must be used • Implementation languages should be to recognized standards • Retesting must occur after the correction of errors
E4	• Formal model of security and semi-formal specification of security enforcing functions • Architecture and detailed design to be produced • Testing must be shown to be sufficient • TOE and tools are under configuration control with changes audited, compiler options documented • TOE to retain security on restart after failure
E5	• Architectural design explains the inter-relationship between security enforcing components • Information on integration process and run time libraries must be produced • Configuration control independent of developer • Identification of configured items as security enforcement or security relevant, with support for variable relationships between them
E6	• Formal description of architecture and security enforcing functions to be produced • Correspondence shown from formal specifications of security enforcing through to source code and tests • Different TOE configurations defined in terms of the formal architectural design • All tools subject to configuration control

Trusted Common Criteria: The first truly international criteria of quality assessment is published under the Common Criteria as an ISO/IEC 15408 standard. While goods in general use are accredited under CSEC, ITSEC, and other standards, it has increasingly replaced all other criteria. It offers a versatile range of functionality and guarantees. It emphasizes on standardizing the general approach to product assessment and mutually accepts those tests worldwide [9].

Table 2-3: Standard EAL Packages [4]

Name		Level of Confidence
EAL1	Functionally tested.	Lowest
EAL2	Structurally tested	
EAL3	Methodically tested and checked	
EAL4	Methodically designed, tested, and reviewed	Medium
EAL5	Semi-formally design and tested	
EAL6	Semi-formally verified design and tested	
EAL7	Formally verified design and tested	High

ISO/IEC 27001:2013

The International Organization for Standardization (ISO) is the leading producer and publisher of international standards in the world.

The 27000 set of standards deals with information technology policies. ISO/IEC 27001:2013 focuses on the standardization and certification of an organization's information security management system.[10].

ISO/IEC 27001:2013 includes guidelines for developing, operating, maintaining and advancing information security management. The main areas are: general ISMS specifications, management responsibility, internal ISMS audits, ISMS management review and ISMS progress.

ISO/IEC 27002:2013 covers the following 14 priority areas:

1. Information Security Policies: consist of providing management guidelines and assistance for information security.

2. Information security organization: this is a structured and established security structure covering information transmission facilities and the properties of information obtained or retained by third parties.
3. Human resources security: consists of providing security aspects for staff.
4. Asset Management: protects the properties of the organisation by ensuring that important data assets are detected and provided with adequate safeguards.
5. Physical and environmental security: consists of preventing access, destruction and unauthorized interaction with facilities and records.
6. Access Control: limits access to data, electronic phones, telephone and network facilities and identifies illegal activities.
7. Cryptography: provides for the right to preserve the secrecy, honesty and authenticity of documents.
8. Operational security: this consists of ensuring the operation of data processing equipment.
9. Security of communications: to ensure the correct exchange of data between organizations.
10. Supplier relationships: implement compliance protocols to secure company knowledge and properties that are open to suppliers.
11. Information security incident management: implement protocols for detecting and responding to information security events.
12. Aspects of information security in business continuity management: it consists of mitigating the effect of an event on the sensitive business.
13. Acquisitions, development and maintenance of information systems: Consists of implementing security controls in the operation and development systems.
14. Compliance: this is to ensure compliance with criminal and civil legislation and legislative, administrative or contractual responsibilities. In addition to complying with internal security procedures and practices allowing for a comprehensive audit mechanism.

Control Objectives for Information and Related Technologies (COBIT) has five principles and seven enablers.

- Principles:

  - Meet the needs of stakeholders
  - End to end coverage of the company
  - Implementation of a single integrated system
  - To make a holistic solution possible
  - Separating governance from management
- Enables:
  - Principles, policies and frameworks
  - Processes
  - Organizational structures
  - Culture, ethics and behaviour
  - Information
  - Services, infrastructure and applications
  - People, skills and competencies

Payment Card Industry Data Security Standard (PCI-DSS): The PCI Security Standards Council has developed PCI-DSS to improve data security for payment cards. The PCI-DSS offers the architect a requirements system to ensure that cardholder data is processed, stored and transmitted safely. PCI-DSS focuses on compliance with a protocol for safety accidents avoidance, diagnosis and response. Objectives are set:

- Build and manage a secure framework and network:
  - Firewall setup install and maintain
  - Do not use device passwords and other security parameters with vendor-supplied defaults
- Protect Cardholder Data:
  - Protect the cardholder information kept
  - Encrypt cardholder data transfer via free public networks
- Maintain a Vulnerability Management Program:
  - Use anti-virus tools or applications and refresh them periodically
  - Developing and maintaining stable software and frameworks
- Implement Strong Access Control Measures:
  - Limiting access by company to cardholder information
  - Assign each person with computer access a unique ID

    - Restrict cardholder physical entry
- Regular Monitor and Test Networks:
    - Track and control all connections to cardholder and network resources
    - Testing applications and procedures on a regular basis
- Information Security Policy:
    - Maintain a policy addressing the security of information for all staff

Countermeasure Selection

Counter-measurement considerations or safeguards include:

- Accountability
- Auditability
- Trusted source
- Independence
- Consistently applied
- Cost-effective
- Reliable
- Independence from other countermeasures (no overlap)
- Ease of use
- Automation
- Sustainable
- Secure
- Protects confidentiality, integrity, and availability of assets
- Can be “backed out” in event of issue
- Creates no new operational problems
- Does not leave any residual functional data

It is obvious from this list that **countermeasures** must be reproachful if used to secure the interests of an organisation. If the risk assessment has finished and there is a list of remediation measures to be carried out, it is necessary to remember that an organisation must ensure that staff with enough capacity are capable of implementing and maintaining remediation activities. This could enable the company to provide employees involved in

developing, deploying, sustaining and supporting protection measures in the community enhanced training opportunities.

Moreover, it is essential for the development, implementation, maintenance, monitoring and environmental compliance of appropriate policies with detailed processes and standards which correspond to each policy object. The organisation should allocate personnel that can be accountable for each mission, monitor activities over time, report results to senior management and give time to receive required approvals.

Security architects design considerations:

- Security architects design considerations:
- How can I use this system as a benchmark?
- Which market problems do I have to consider?
- Who are my stakeholders?
- How can I use this device configuration in the architecture as a whole?
- How are the SPOFs in this architecture going to be?

Security experts must remember:

- What are the metrics for maintaining these systems?
- To make the device work effectively, whom do I have to collaborate with?
- Why don't we tackle this or that?
- How can I convey to each of my consumer audiences the required device knowledge level?

When a security professional begins to deploy the enterprise security architecture, he or she must know:

- The tool(s) he or she is going to use.
- The end users of the system
- The time to do it
- Integrate the design of this system into my existing network.
- Where I am going to manage the system from

The practitioner's challenge is and try to organize and canalize all these thinking streams in a way that will allow him or her to deploy a cohesive and solid corporate security infrastructure.

The task for the practitioners is to try and coordinate and channel all these thought streams through a mechanism that will enable them to manage a consistent and robust corporate security infrastructure

These three safety players are important, and they lead, in their own way, to the success or loss of the company security infrastructure. All three share a whole host of items, though. They all need to work in order for the others to do their job. Both of them must ensure that dialogue about their solution part is bidirectional, straightforward and concise on problems and architectural considerations. Most notably, they must be mindful of the common sense and evaluate any acts that are done to deal with the whole architecture, rather than only the areas of the architecture for which they are responsible.

Security specialists must become specialists in understanding and managing risk, each in his or her own field, but at the same time with a shared objective in mind. This objective is to control risk so that it does not have an adverse effect on the business. This objective is maintained with anyone who, for any reason, communicates at all levels with the architecture.

End-users must use systems such that their conduct does not put them in danger and vulnerability. Device operators must ensure that the networks are up-to-date with security patching to ensure that any identified vulnerability in the system is mitigated. High management must have enough resources to make sure the facilities are managed when necessary to ensure that all customers enjoy secure working conditions.

2.4 Vulnerabilities of security architectures, designs, and solution elements

Insecure networks can be vulnerable and endangered. Common vulnerabilities include inadequate maintenance of the memory of main system components such as hardware resources and central OS functions, lack of redundancy of system systems and poor security. Hardware malfunction, server rights violation, buffer overflows and other memory attacks, Denial of Service, Reverse Technology and system hacking common challenges to system availability, integrity, and confidentiality.

Security accidents and data breaches are correlated with web applications. Many attacks have been aimed at common blogging sites or Joomla's content management systems, WordPress and Drupal.

Threat accidents are also related to human error. Mismanagement, publication errors and deletion errors are the top three errors. The easiest way to proactively fix staff mistakes would be a correctly applied and optimized data loss management technology.

Cyber-spy operation is related to reported cases of security and privacy abuses. Network segmentation, diligent log-management and authentication in two-factor systems can help to protect the side movements if an attacker already joins and tries to reach sensitive infrastructure.

Point of sale (POS) systems also suffer from data breaches. Weak, default passwords for point of sale systems are constantly being attacked by attackers.

One of the main problems of IT security practitioners were lost or robbed devices. Security architects must enforce protection of devices, enable consumers to maintain devices at all times, and implement an appropriate safeguard policy to address these risks.

One of the main problems of IT security practitioners were lost or robbed devices. Security architects must enforce protection of devices, enable consumers to maintain devices at all times, and implement an appropriate safeguard policy to address these risks.

Abuse by an employee, contractor or associate of rights may constitute an incidence of protection and infringements of records. Since the bulk of the insider assaults have taken place in the workplace network, security architects should create enhanced protections around confidential data structures. The user account activities should be examined and ex-employee user accounts easily deactivated during an overall life cycle of access management.

Crimeware is a common issue that nearly every company is dealing with, mostly financially driven attacks using automatic attack kits. The Zeus and SpyEye Trojans spread quickly across orchestrated cybercrime networks that make people download harmful malware with spam messages and Malicious links.

The malware is intended to rob bank credentials and empty bank accounts. Many infections occur precisely when a website is visited or when a malicious file is downloaded. Security

architects need security fixes deployed and browser plug-ins need to be updated. If not required, Java should be deactivated or uninstalled. Moreover, security of at least two factor will foil most threats using robbed credentials.

An attacker that compromises servers in public and private data centers in the cloud to maximize the bandwidth of the network attacks performs a distributed denial-of-service attack intended to paralyze the network or to pull down pages and web applications. Security architects should provide a strategy and plan to use an Internet Service Provider mitigating service for DDoS, and should also consider the isolation of non-active IP addresses.

Three-quarters of all cases include infected web servers that are usually massively attacked, hosting malware for drives or phishing pages by hundreds of servers.

Security architects must use other tools and data points to see if their networks face risks. Resources such as the report of Verizon Data Breach are useful because they have results across a broad range of sectors and architectures. Here are some other tools that the architect for protection may find useful in this respect.

- The Secunia Vulnerability Review 2014
  http://secunia.com/?action=fetch&filename=secunia_vulnerability_review_2014.pdf
- The Symantec Internet Security Threat Report 2014
  http://www.symantec.com/content/en/us/enterprise/other_resources/bistr_main_report_v19_21291018.en-us.pdf
- The Sophos Security Threat Report 2014
  http://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-security-threat-report-2014.pdf
- The Cisco 2014 Annual Security Report
  https://www.cisco.com/web/offer/gist_ty2_asset/Cisco_2014_ASR.pdf
- Price Waterhouse Coopers the Global State of Information Security Survey 2014
  http://www.pwc.com/gx/en/consulting-services/information-security-survey/index.jhtml
- Trustwave's 2014 Security Pressures Report
  http://www2.trustwave.com/rs/trustwave/images/2014%20Trustwave%20Security%20Pressures%

- Websense 2014 Threat Report
  http://www.websense.com/assets/reports/report-2014-threat-report-en.pdf

2.4.1 Vulnerabilities in Web-based systems

Vulnerabilities are experienced in software and processes. Remedial steps must be taken on web-based applications, customers, servers, and technology supporting them. Sound design can help reduce and remove bugs prior to developing through a stable architecture. Both risks and security measures are imposed on web applications. Because of their accessibility, online apps are insecure.

The top 10 vulnerabilities according to OWASP are [11]

1) Injection. Untrusted data is sent to an interpreter as part of a command or query when injecting flaws such as SQL, NoSQL, OS, and LDAP injection are reported. The hostile data of the attacker will cause the interpreter to execute unintended commands without proper permission.
2) Broken Authentication. Authentication and session management program features are frequently poorly designed to enable attackers to compromise credentials, keys or tokens, or leverage additional design vulnerabilities to momentarily or permanently assume the identity of other users.
3) Sensitive Data Exposure. Many web applications and APIs fail to appropriately safeguard sensitive data like banking, healthcare and PII. Attackers may rob or alter the weakly protected data for the purpose of credit card fraud, identity robbery or other crimes. Without additional security, sensitive data may be breached such as rest encryption or transit data, and when shared with the browser it needs particular care.
4) XML External Entities (XXE). For analysing external object references, often older or poorly developed XML processors use XML documents. The URI handler, the internal exchange of file, internal port scanning, remote code execution, and denial of service attack can be used for the exposing of internal files by external organizations.
5) Broken Access Control. Restrictions on what authenticated users are also not fully applied. These bugs may be used by attackers to access unauthorized functions and/or

data, such as accessing accounts of other users, viewing confidential files, changing data from other users, changing access privileges, etc.

6) Security Misconfiguration. The most prominent problem is security misconfiguration. This is typically a product of precarious configurations, unfinished configurations or ad-hoc configurations, free cloud storage and HTTP misconfiguring headers. It is not only necessary to safely configure all operating systems, framework, libraries and frameworks but also to patch/upgrade them on a timely basis.
7) Cross-Site Scripting XSS. XSS faults occur where an application has untrusted data on a new web page without correct authentication or escape or when it uses a browser-approved API and can generate HTML or JavaScript to update an existing webpage with the data supplied. XSS permits attackers to run scripts on their browsers which are captured by user sessions and default web pages.
8) Insecure Deserialization. Incorrect deserialization leads to execution of remote code. While deserialization vulnerabilities do not result in remote code execution, attacks such as replays, insertion assaults and escalation privilege may be used.
9) Using Components with Known Vulnerabilities. Includes the same privileges as the program components such as library, frameworks and other modules. A compromised aspect may be used to make a major data failure or server seizure easier. Applications and APIs using known vulnerability components will damage application defenses and allow for many attacks and impacts.
10) Insufficient Logging & Monitoring. Lack of logging and tracking, along with lack of or inadequate integration in response to incidents, enables an attacker to target additional systems, to keep persistence, to pivot more systems and to manipulate, remove or delete data. Most violation studies find that the time needed to identify a violation is over 200 days, mostly identified by outside parties instead of internal or surveillance systems.

Specific protections can help to have a specific assurance process for web server approval. They include hardening the operating system used by such servers (removing default accounts and settings, correctly setting permissions and privileges, and keeping vendor patches up to date).

Ensure exclusion or securing of administrative interfaces. Allow access only from approved hosts or networks, and use strong user authentication. Do not use certificates, or equivalent high-trust authenticators to encrypt authentication credentials in the program itself, and ensure the security of credit documents. Use blocking, and enhanced account logging and auditing, and ensure all authentication traffic is encrypted. The interface should be as secure as the rest of the program.

Input validation is important due to the simplicity of web services and software. Application proxy firewalls are sufficient, but ensure the proxies are in a position to fix buffer overload problems, authentication issues, scripting, submission of database engine-related commands (e.g., SQL commands) issues, encoding issues (such as Unicode), and URL encoding and transition. The Proxy Firewall will solve problems related to in-house and custom app data submission to ensure that the feedback for such systems is validated. (The submission shall be subject to personalized scheduling of this standard of protection.)

2.4.2 Vulnerabilities in mobile systems

2.4.2.1 Vulnerabilities and countermeasures in Mobile Applications

1. Binary Protection: Insufficient Jailbreak / Root Detection. A computer roots or jails bypass data security and system encryption schemes. Any type of malicious code, which dramatically alters the behaviour of the program logic, may be executed on the computer when a device has been compromised. Forensic applications for recovery and data are commonly often run on rooted computers.

   Solution: It is easier not to run an application on the root, jailbroken or at least any root/jailbreak detection in terms of security. The detection of the compromise of a system provides an extra layer of policy compliance and risk avoidance in order to protect data from exposure within the program.
2. Insufficient Transport Layer Protection: For all authenticated links, particularly web-accessible sites, encryption should be utilized. The back-end links should either be encrypted or the vulnerability disclosure to bad actors of authentication or session token. When confidential data, such as credits, health records or other information, is

exchanged, encryption should be used. Any program that is restored to plaintext or coerced out of the encryption mode can be violated by an assailant.

Solution: Ensure that the application has a security boundary, which establishes a secure transport guarantee for confidentiality and integrity to ensure that all information that is sent is not detected.

3. Information Leakage – Server Version: Service Version Information Leakage: Server information is present in the response. Information Leakage is an implementation flaw in which confidential data, such as web application technical details, setting, or user-specific data, is revealed.

   An attacker may use sensitive data to hack the target application, its hosting network, or its users; sensitive data leakage should be restricted or avoided wherever possible. In its most basic form, information leakage is caused by one or more of the following factors: Failure to clean out HTML/Script comments containing confidential data, incorrect application or server settings, or discrepancies in page responses for true and invalid data.

   Solution: Remove redundant data from server responses that could provide an attacker with additional information about your network.

4. Information Leakage – Sensitive Data: This is similar to the Server version in 3, but it focuses on more leakage inside the app, app-to-app communication, and so on.

   Solution: Information leakage can be classified into two types: global and resource-specific. Global information leakage vulnerabilities are often connected to long winded error messages or server / application interface version disclosures. A configuration setting will also resolve these leaks. The disclosure of developer comments, files, or confidential personal details are examples of resource-specific information leakage issues. When resource-specific leaks occur, they often necessitate immediate mitigation.

5. Insufficient Authorization/Authentication: When an application fails to perform adequate authorization checks to ensure that the user is performing a function or accessing data in accordance with the security policy, this is known as insufficient authorization. What a person, facility, or program is allowed to do should be enforced by authorization procedures. When a user logs in to a website, this does not always imply that the user has complete access to all information and features.

Solution: Implement a tried-and-true authorization system that favors policy-based configuration files over hard-coded authentication/authorization checks wherever possible.

6. Cryptography – Improper Certificate Validation: This program either does not validate SSL/TLS certificates or uses an SSL/TLS certificate validation method that does not correctly verify that the certificate was issued by a trusted provider. If the certificate cannot be validated or is not issued, the client should be configured to drop the connection. Any data sent over a link with a certificate that hasn't been properly checked is vulnerable to unauthorized access or alteration.

   Solution: Make sure your application's certificate validation is set up to correctly verify that a certificate is being issued by a legitimate Certificate Authority. Alternatively, enforce the most current IETF or CA/B Forum certificate accountability requirements.

7. Brute Force – User Enumeration: User Enumeration: A brute force attack is a technique for determining an unknown value by attempting a large number of potential values using an automated procedure. The attack takes advantage of the fact that the values' entropy is lower than it seems.

   Solution: The user enumeration vulnerability is most commonly found in the following features: Login, log, or reset your password. It is not appropriate for the application to reveal whether or not a username is legitimate. In both fields, the response to valid and invalid input should be similar.

8. Insufficient Session Expiration: When a user exits an application, the identifiers that were used during the session should be invalidated. It is possible for other users to impersonate the user and perform actions on his behalf if the server fails to invalidate the session identifiers.

   Solution: First, make sure the application has a logout button; second, make sure that when the user clicks this button, their session is properly invalidated.

9. Information Leakage – Application Cache: Confidential details may be leaked from application caches through the main application code or third-party frameworks. When it comes to safe data storage, mobile devices present a particular challenge. The devices can easily be misplaced or stolen. Many people don't lock their phones. An intruder conducting data forensics on the physical device may access the cached data.

Solution: Ensure that confidential data does not leak into the cache by mistake. It can be avoided by having a threat model for the OS, system, and platform that checks and verifies how data is treated during URL caching, keyboard press caching, logging, copy or paste caching, app backgrounding, browser cookies artifacts, HTML5 data storage, and analytic data sent to the server or another app.

10. Binary Protection – Insufficient Code Obfuscation: It's Android/Java specific, the most popular OS phone. Several tools have been developed to scramble or obfuscate the code to protect Java applications against reverse engineering. As part of the Android SDK, Google included one of these most popular tools, ProGuard. The tool ProGuard narrows, optimizes, and hides your code by removing unused code and renaming classes, fields, and procedures with semantically unclear names. The result is a smaller .apk file which is harder to reverse.

    Solution: ProGuard is built into the Android system to prevent manual invoking. ProGuard only works when you create your application in the release mode, so when you build a Debug application, you don't have to deal with the obscured code. It is completely optional but highly recommended to have ProGuard run and can help your safety on these systems.

2.4.2.2 Risk from Remote Computing

Remote users are considered protected by companies accessing their corporate network through a virtual private network (VPN). However, although VPNs provide a tunnel connection that can access the corporate intranet by only authenticated users, they are not a complete and end-in-end solution. VPNs do not guarantee that remote devices are free of software and configuration vulnerabilities. These malware examples are easily introduced and remote machines can expose these vulnerabilities to critical network assets. VPNs don't guarantee the remote and mobile devices have bugs that can be used to spread viruses or worms, without the use of software and configuration. The risks associated with the End point device may include:

- Trusted Clients.
- Network Architectures. What is the position of the infrastructure?

- Policy Implementation. Incorrect, unsatisfactory or weakly controlled hackers and malware can easily be by-passed.
- Devices stolen or lost

2.4.2.3 Risk from Mobile Workers

IT and security services face a range of devices and operating systems with the challenge of implementing and managing mobile security. One of the most common data leakage risks is often found in desktop and multi-device synchronizations.

2.4.2.4 Potential attack vectors for mobile devices:

Potential mobile device attack vectors: SMS, Wi-fi, Bluetooth, Infra-red, USB, Web browser, Email client, Third-party applications, “Jail-broken” phones, Operating system vulnerabilities, Physical access.

Potential examples of potential attacker priorities: SMS, Email, Phone, Video/Photo, Social Networking, Location Information, Voice Recording, Documents, Credentials.

Guidance should be given by the following standards

- NIST SP 800-121 Revision 1
- NIST SP 800-124 Revision 1
- NIST SP 800-40 Revision 3.

2.4.3 Vulnerabilities in embedded devices and cyber-physical systems

An **embedded system** is a device which can do computation on the real-time data. It comprises of hardware, software and mechanical parts such as sensors and actuators. The embedded system is used to perform a specific function under certain constraints [9].
The integration of applications such as web server, Secure Socket Shel (SSH) and telnet into the embedded system for user convenience has increased the attack surface for the hackers. Therefore, the need for strong security in the embedded system is very vital. Security in embedded systems is crucial as it is integrated into the real-time safety-critical applications.

Most functionalities of the system are enhanced by giving networking capabilities to them, but it increases the threats for the attackers.

The embedded systems are constrained by power, memory, and processor. The **security requirements** for the embedded system is almost the same as the requirement of the information system. The main aim of the attack is to disrupt the integrity and confidentiality of the system. Hence, the primary objective of an embedded system is to guarantee integrity, confidentiality, and availability.

Availability: It guarantees that the system is available when need by the user. Generally, it is considered as the time in which the system is able to accomplish the given task to it.

Integrity: Integrity implies that the system should not allow any modification in the data from an external factor. Integrity involves the guarantee and reliability of the data in and out of the embedded system.

Confidentiality: This requirement implies that the access to any data in the system should be properly authorized and there should not be any data available without access control.

2.4.3.1 Main threats and countermeasures

Table 2-4. Attacks and its countermeasures [12]

Category	Attack	Countermeasures
Software- based attack	Malware	▪ Anti-malware application ▪ Machine-learning-based application
	Brute force	▪ Limiting the number of tries
	Buffer overflow	▪ Hardware/Software Defender Technique
	Web-based vulnerability	▪ Sandboxing

Network-based attack	MITM	▪ Ipsec
	DNS posing	▪ DNSSEC
	Session hijacking	▪ Encryption, disposable credits
	Signal jamming	▪ Anti-jamming mechanism
Physical and side channel based attacks	Power Analysis	▪ Data Masking technique
	Timing Attacks	▪ Random clock technique
	Electromagnetic Analysis Attack	▪ Shielding techniques ▪ Asynchronism

2.4.3.1.1 Software-based attacks

Most of the vulnerabilities and threats arise in software applications. The vulnerabilities and threats of the web server are indirectly inherited to the embedded systems.

- **Malware** is a piece of code which is intended to harm the embedded system by taking control of it. Most of the malware gets into the system through improper download and execution of the executables and firmware update. The user must be conscious of the source from which they download the firmware updates and patches.
- **Brute force:** Brute force is the methodology by which the attacker tries all the password from a known dictionary file or word list. If the user sets the default password or weak password, it becomes easier for the attacker to gain control over the system.
- **Buffer overflow:** A buffer overflow occurs when the memory pointer is not restricted inside the scope. When the attacker exploits the pointer and interprets the out of bound data as code, he would be successful in taking control of the system.
- **Web-based vulnerability:** Most of the embedded systems have a web server build into it to serve the user with GUI controls, therefore, it inherits all the vulnerabilities from the web server.

2.4.3.1.2 Network-based attacks

The network capabilities help the user to remotely manage the system remotely and control it but they also inherit the networking infrastructure vulnerabilities such as Man In The Middle (MITM), packet injection and replay attacks.

- **Man In The Middle (MITM)** is a network-based attack in which attack redirects the traffic through his computer. Lack of encryption will make it easy for the attacker to interpret the data. All the data from the system will pass through the attacker system creating an opportunity for the attacker to modify the data and compromise the integrity of the system.
- **DNS poisoning:** The attacker may poison the local Domain Name System (DNS) server to modify the records to his/her needs. When the system tries to send data to a legitimate website, the data will be directed to the attacker's website because of the poisoning of the DNS. An attacker can host a fake website to gather user credentials and data. Using protocols like DNSSEC will help in mitigating the risk.
- **Session hijacking:** Authentication methodologies are integrated with sessions to control the access and bring state during communication. The attacker will use attacks such as MITM before going for session hijacking. When the data pass through the attacker network, software tools can be used to capture the session and reuse it for authentication.
- **Signal jamming:** occurs when the attacker jams the signal by causing interference and distortion of the actual signal. It happens mostly in the wireless mode of communication. This attack compromises the availability of the embedded system because the system will be incapable of receiving and sending the data.

2.4.3.1.3 Physical and side-channel based attacks

These attacks can be launched with the help of probing instruments and eavesdropping on the data flowing between the interconnections. Most of the embedded systems use complicated techniques to safeguard from physical attacks such as obfuscation. Side-channel based attacks are launched using the information revealed by the systems such as timing, power consumption, and electromagnetic leakage. It gives information on the internal

operation of the system which can aid the attacker to solve his riddle such as finding the cryptographic keys.

- **Power Analysis Attack:** The switching activities inside the hardware circuit represent the power consumption of that device. This attack needs physical access to the system and probing of the connections. These attacks are termed power analysis attacks. They can be employed in compromising the embedded system such as a smart card system.
- **Timing Attacks** exploit the timing of execution of the cryptographic algorithm to find the keys and algorithms used underneath. This is because of the reason that the computational timing is directly proportional to the cryptographic key.
- **Electromagnetic Analysis Attack:** The sensitive data from the system is identified by measuring the electromagnetic radiation from the chip. Knowledge of the system layout is required to exploit the system using this kind of attack.

Countermeasures must make sure that it guarantees confidentiality, integrity, and availability. The obstacle to the implementation of the countermeasure is due to the constraints tied with the embedded systems. Those constraints are batter, processing power and memory. Misconfiguration of a device may compromise the integrity system. The firmware of the embedded system must be updated and patched properly. Protocols must implement safety on its own rather than requiring to be forced.

2.4.3.2 Cyber Physical Systems (CPS)

Intelligent networked systems with integrated sensors, processors and actuators, the Cyber Physical Systems (CPS) detect and communicate with the environment and provide guaranteed, in safety critical applications, real-time performance. The joint behavior of the system's "cyber" and "physical" components is central in CPS systems. Each component requires the integration of computation, control, detection and networking, and component and device operation needs to be coordinated.

Partial list of areas where CPS services can be found is: transport, manufacture, health services, energy, agriculture, security, construction monitoring, emergency response systems.

PSC-based architectures face a number of challenges: cybersecurity and interoperability. Many attacks on conventional infrastructure, including websites, databases and networks, can easily be modified to target CPS solutions like smart grids and transport networks. The following are some of the linked primary technologies for incorporating and handling CPS offers:

- Abstractions, modularity and composability: to allow the combination and re-use of CPS system elements with the retention of security, safety and reliability.
- System engineering architectures and standards: for the efficient design, implementation or integration of reliable systems.
- Adaptive and predictive hybrid hierarchical control: achieving closely organized and synchronized behaviours and interactions in synchronous and distributional systems.
- Multi-physics models and software models are integrated: enabling the co-design and predictive system conduct of physically designed and compute components.
- Distributed sensing, communications, and perception: to make CPS distributed networks scalable, stable and high performance, to provide an accurate and reliable model worldwide, allowing time-conscious and time-critical functionality.
- Diagnostics and prognostics: to detect, anticipate and avoid faults in complex systems or recover from them.
- Cybersecurity: safeguarding security from malicious CPS device attacks.
- Validation, validation and certification: to accelerate the design cycle in order to market technologies while maintaining high reliability in device safety and functionality.
- Autonomy and human interaction: build autonomous CPS and human models to promote models-based design of human-used reactive systems

Besides traditional IT security mechanisms for preventing intrusion detection systems, such as authentication, encryption, firewall detection and forensics, new CPS security mechanisms need to be developed and incorporated into a company. Many areas have to be considered: risk evaluation, mechanisms of poor data recognition, architecture of system resilience and survival for attacks.

2.4.3.3 Industrial Control Systems (ICS)

In order to handle industrial processes, such as manufactures, product handling, production and distribution, Industrial Control Systems (ICS) have been used. SCADA systems are the main subgroup of ICS and process systems in large scale. They use the commercial off-the-self software on the ICS basis of traditional embedded systems platforms. Well-known ICS categories include supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and programmable logic controllers (PLC).

SCADA systems constitute the main subsection of ICS and process systems with wide distances and multifaceted locations. A Supervisory Controlling and Data Acquisition System (SCADA) is an array of interconnected devices used in industrial environments for the control and monitoring of physical equipment. It is used to automate electricity generation, transmission and distribution processes. It is also used for pipeline control, water treatment and distribution, and chemical production and processing, as well as in railway systems and other transportation services.

Main threats to process control and automation systems and the security of the industrial control system:

- Unauthorized use of remote maintenance access points: maintenance access points are intentionally created and often insufficiently protected external entrances to the ICS network.
- Online attacks via office or enterprise networks: office IT is normally connected in various ways to the network. Network links from offices to the ICS network are also available, in most situations, for attackers to obtain access to this path.
- Attacks on standard components used in the ICS network: standard IT components like software systems, application servers, or databases frequently include faults and bugs that attackers may use. When used in the ICS network too, these standard components raise the chance of a successful assault on the ICS network.
- (D) DoS Attacks: denial-of-service attacks can conflict with network links and key resources and may result in device failure to interrupt an ICS, for example.
- Human error and sabotage: Mistreatment and human error are also a major threat in regards to secrecy and availability, particularly in terms of security goals.

- Introducing malware via removable media and external hardware: the usage of removable media and mobile IT components by external personnel often involves high likelihood of malware infections.
- Reading and writing news in the ICS network: Plain text protocols are currently used in most control components, while messages are not covered. This makes reading and introducing control commands reasonably simple.
- Unauthorized access to resources: Internal attackers and subsequent attacks after an initial external intrusion are particularly simple when services and network elements do not use the methods of authentication, authorisation or vulnerability.
- Attacks on network components: Network components can be manipulated by attackers to perform man-in-the-middle attacks or make sniffing ease.
- Technical malfunctions or force majeure: Severe weather-based outages or technical malfunctions can occur at any time.

All applicable guidelines and instructions on device design must be considered by the security architect. The security practitioner must know the best practices for the management, repair, service and safeguarding of these systems. In non-nuclear critical energy infrastructure, the following principles are especially applicable to the cyber safety of ICT systems.

- The ISO 27000 series. It defines organizational and technological data protection management specifications. This basis is defined in more depth in the ISO Standard 27001 for information security management.
- ISO 27032:2012. The issue of the dynamic interaction of Internet security, network security and application security is unique to ISO 27032:2012. It also addresses controls for all stakeholders in cyberspace (consumer and provider organizations). It is singular since it specifically tackles concerns such as social engineering attacks controls, cybersecurity readiness and understanding. It requires, most significantly, a knowledge exchange and collaboration system.
- IEC 62351. For power system control activities IEC 62351 explicitly targets security details. The Security Standards of the IEC TC 57 working group, IEC 60870-5, IEC 60870-6 series, IEC 61850 series, IEC 61970 series and IEC 61968 series, are mainly applied in respect of communication protocol. These requirements apply to producers in

particular. The SGIS110 Group M/490 is expanding these principles to cover new technological aspects in the area of cyber security with intelligent grids.

- The IEC 62443 series. The IEC series 62443 (based on the ISA-99) covers industrial control systems safety protection (IACS). The emphasis is on best practices in operations. The standard targets owners of assets, system integrators and suppliers of components with different sub-standards. IEC 62443 attempts, particularly with NISTIR 7628 and ISO 27001/2, to include and comply with established standards.
- NIST Special Publication 800-39. Special Publication NIST 800-39, Risk, Organization, Mission & Information System Information Management References ISO 27000 and ISO 31000 and ISO 27005 for ISMS Module References (risk management). It calls for a unified approach to risk management.
- Cybersecurity targets for electricity networks NISTIR 7628 (Guidelines for the cyber security of smart grid). The report deals with safety standards. It identifies seven areas in the intelligent grid and describes categories of logical Interface (Operations, distribution, transmission, etc.). These configuration categories can then be extended with the security specifications (e.g. integrity, authentication, bandwidth, real-time requirements). ISO 27001, 27002 and IEC 62351 cover much of the security specifications of NISTIR 7628. The following standards provide additional links to the security measures in the Catalogue of Control System Security Recommendations: FIPS 140-2, NERC CIP and IEEE 1402 Appendix A: (Guide for Electric Power Substation Physical and Electronic Security).
- Crisis Infrastructure Protection Standards (CIP) have been established by the North America Electric Reliability Corporation (NERC). Separate criteria for the implementation of a robust cybersecurity system from CIP-002 through CIP-009 exist. After the Energy Policy Act of 2005, CIP enforcement is compulsory for electricity suppliers. The Category of Cyber Critical Assets in the Multi Electric System uses a risk-based approach. The present implementing standards are described below:
    - CIP-002-3 Critical Cyber Asset Identification
    - CIP-003-3 Security Management Controls
    - CIP-004-3a Personnel and Training
    - CIP-005-3a Electronic Security Perimeter

- CIP-006-3c Physical Security of BES Cyber Systems
- CIP-007-3a Systems Security Management
- CIP-008-3 Incident Reporting and Response Planning
- CIP-009-3 Recovery Plans for BES Cyber Systems

2.4.3.4 Identify attack taxonomy classification criteria

Based on existing attack taxonomies and attacks, we can define the following dimensions in relation with the CVE (Common Vulnerabilities and Exposures).

The following forms of specifications apply to the attacker for the preconditions:

- Internet facing device: A remote attacker can exploit several vulnerabilities in CVE records when the device is connected to the internet. The attacker does not need access privileges; only that the assailant can discover and send messages through the network to the computer.
- Local or remote access to the device: A remote attacker can exploit several vulnerabilities in CVE records when the device is connected to the internet. The attacker does not need access privileges; only that the assailant can discover and send messages through the network to the computer.
- Direct physical access to the device: The attacker requires physical access to the computer directly. There may be no need for the attacker for access to the device's services.
- Physical proximity of the attacker: Physical proximity of the attacker: In many situations the attacker needs no physical access, the attacker can be near the computer physically. For example, wireless attacks that only have to be within the target device's radio range.
- Miscellaneous: An example for a miscellaneous precondition is when the target device has to run some software or has to be configured in a certain way for the vulnerability to be exploitable.
- Unknown: In certain situations, the preconditions for the potential attack are not adequately identified and, in these cases, the preconditions are classified as unknown.

V**ulnerabilities**:

- Programming errors: Many of the vulnerabilities come from programming errors, which can lead to control flow attacks (for example, parsing of input vulnerabilities that lead to buffer overflow problems, and memory management problems, such as the use of pointers that refers to the memory locations that have been freed).
- Web-based vulnerability: A web-based management system is available for many embedded devices. Web server software on those computers is however not regularly updated. These devices are exposed to web-based attacks which take advantage of unpatched web-based vulnerabilities on the device.
- Weak access control or authentication: Many devices use default or weak passwords, and some devices have hard-coded passwords that allow backdated access for anyone with a hard-coded password.
- Misuse of cryptography: Some devices use authentication cryptographic mechanisms to protect the confidentiality of certain sensitive data. Cryptography mechanisms are often not used correctly and contribute to safety deficiencies.
- Unknown: Some CVE documents do not provide details on the vulnerability itself, similar to preconditions, when they identify the target and impact of possible attacks using unspecified vulnerability.

Attacks:

- Control hijacking attacks: These attacks are responsible for distorting the regular control flow of programs operating on an embedded computer, and usually result in the execution of the attacker's code.
- Reverse engineering: An attacker can also get confidential information through an embedded system analysis of software. This is known as reverse engineering. The attacker will identify faults in the code that can be exploited with other attack methods using reverse engineering techniques.
- Malware: An attacker can use malware to attack an embedded device. Various types of malware are available. One of the common features is that they all give the infected device unwanted, potentially dangerous features. A malware which infects an

embedded device may change the device's actions, which could result beyond the cyber domain.

- Injecting crafted packets or input: It is an attack method against protocols used by embedded devices. A similar type of attack is the manipulation of the input to a program running on an embedded device. Both packet and input crafting attacks exploit parsing vulnerabilities in protocol implementations or other programs. Replaying previously observed packets or packet fragments can be considered as a special form of packet crafting, which can be an effective method to cause protocol failures.
- Eavesdropping: Eavesdropping is a method to target embedded protocols. Both packets and input handlers exploit encoding vulnerability in protocol implementations and other applications. Replaying previously observed packets or packet fragments can be viewed as an effective protocol failure mechanism, a specific form of packet-making.
- Brute-force search attacks: Brute force search attacks can be broken by weak encryption and weak authentication methods. These include extensive search attacks of cryptographic algorithms including ciphers and MAC functions and dictionaries on authentication systems using passwords. The attacks of Bruteforce can only be performed if the search area is small enough.
- Normal use: This refers to the attack which operates via normal use an unprotected device or protocol. For instance, if the system does not have an access control mechanism in place, the attacker could access files on an embedded device just like any other user.

Effect of attacks:

- Denial-of-Service: Several CVE records describe possible attacks that lead to denial of services such as failure or system completeness.
- Code execution: A significant number of the CVE records examined recognize execution of the attacker code given on the embedded instrument as a possible attack result. This does not only include system code but also Web scripts and SQL injections.

- Integrity violation: An integrity violation of certain data or code on the device is a commonly observed effect of potential attacks. This includes modifying files and settings and illegitimate firmware updates and some device applications.
- Information leakage: In certain situations, the result of the attack is the leakage of information.
- Illegitimate access: Several attacks result in the attacker gaining illegitimate access to the device.
- Financial loss: Most attacks will generally result in financial losses.
- Degraded protection level: often the potential attack results in a lower protection level than expected. One example of this will be the use of weaker algorithms and security policies by a system than those supported.
- Miscellaneous: Some attacks lead to the redirection of users to bad websites or to the redirection of the traffic.

2.5 Referencing

[4] CISSP, Official (ISC)² guide to the CISSP CBK, Fourth Edition.

[5] ISO/IEC 15288:2008, «Online Browsing Platform (OBP),» [En línea]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:21827:ed-2:v1:en. [Último acceso: 24 November 2020].

[6] ISO/IEC 21827:2008, «Online Browsing Platform (OBP),» [En línea]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:21827:ed-2:v1:en. [Último acceso: November 2020].

[7] R. Ross, M. McEvilley y J. Oren, «Engineering Principles for Information Technology Security (A Baseline for Achieving Security), Revision A,» *NIST Special Publication (SP) 800-27 Revision A,* November 15, 2017.

[8] Cisco Secure Development Lifecycle,» [En línea]. Available: https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/cisco-secure-development-lifecycle.pdf. [Último acceso: 23 November 2020].

[9] International Organization for Standararization, «ISO/IEC 154408,» [En línea]. Available: https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html. [Last Access: November 2020].

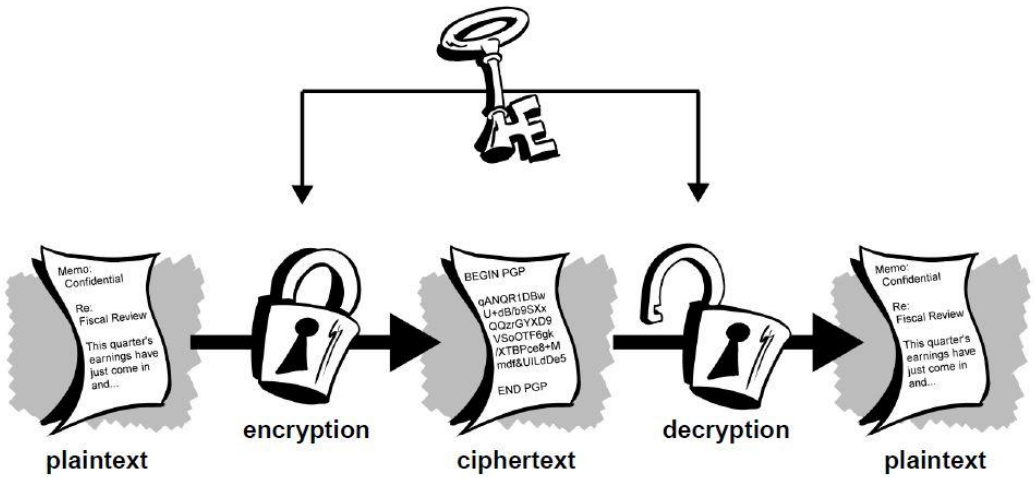
[10] The ISO 27000 Directory,» [En línea]. Available: http://www.27000.org/. [Last Access: 23 November 2020].

[11] Open Web Application Security Project, «WEB SECURITY TESTING GUIDE 4.1,» OWASP

[12] G. Rajendran, «Security in the embedded system: Attacks and Countermeasures,» *International Conference on Recent Trends in Computing, Communication and Networking Technologies,* 2019.

3 Security Engineering: Cryptography & Key Management

Author(s): Maria Papaioannou
Claudia Barbosa
Jonathan Rodriguez



3.1 Symmetric and Asymmetric Encryption

3.1.1 Cryptography

``*Cryptography provides techniques for keeping information secret, for determining that information has not been tampered with, and for determining who authored pieces of information´´* [13].

Handbook of applied cryptography. Menezes AJ, Katz J, Van Oorschot PC, Vanstone SA. CRC press; 1996 Oct 16.

ITU-T Recommendation X.800 defines the general security architecture OSI, as a useful to managers way of managing and providing security tasks. The OSI security architecture focuses on:

(i) **security attacks:** actions compromising the security of organization´s data and information,

(ii) **security mechanisms:** processes (or devices incorporating such processes) for detecting, preventing, or recovering from security attacks to organization´s information, and

(iii) **security services:** processing or communication services to counter security attacks using one or more security mechanisms, as well as to enhance the security of the organization´s information transfers and the data processing systems.

3.1.2 Cryptographic algorithms

Following the most important security mechanisms, which include:

- **Cryptographic algorithms:** This lecture covers this topic.
- **Digital signature:** Cryptographic transformation of, or data affixed to, data units that enables the data unit recipient to verify the source and the received data unit integrity, as well as protect against forgery.
- **Authentication exchange:** A mechanism that enables identity verification through information exchange.

- **Data integrity:** This category includes mechanisms for ensuring data integrity (i.e., single data units or stream of data units).
- **Routing control:** A control that enables selection of logically or physically reliable routes for transiting certain data or data streams allowing routing changes, especially in the case of suspected breach of security.
- **Notarization:** The use of a trusted third party to ensure particular properties of a data exchange.
- **Traffic padding:** The insertion of bits into gaps in data or data stream to hinder attempts of traffic analysis by malicious actors.
- **Access control:** A number of mechanisms that enforce access rights to resources.

Cryptography is an essential component in the secure storage and transmission of data and in the secure interaction between parties. The rest of the lecture provides brief technical introductions to important aspects of the use of cryptography and cryptographic algorithms.

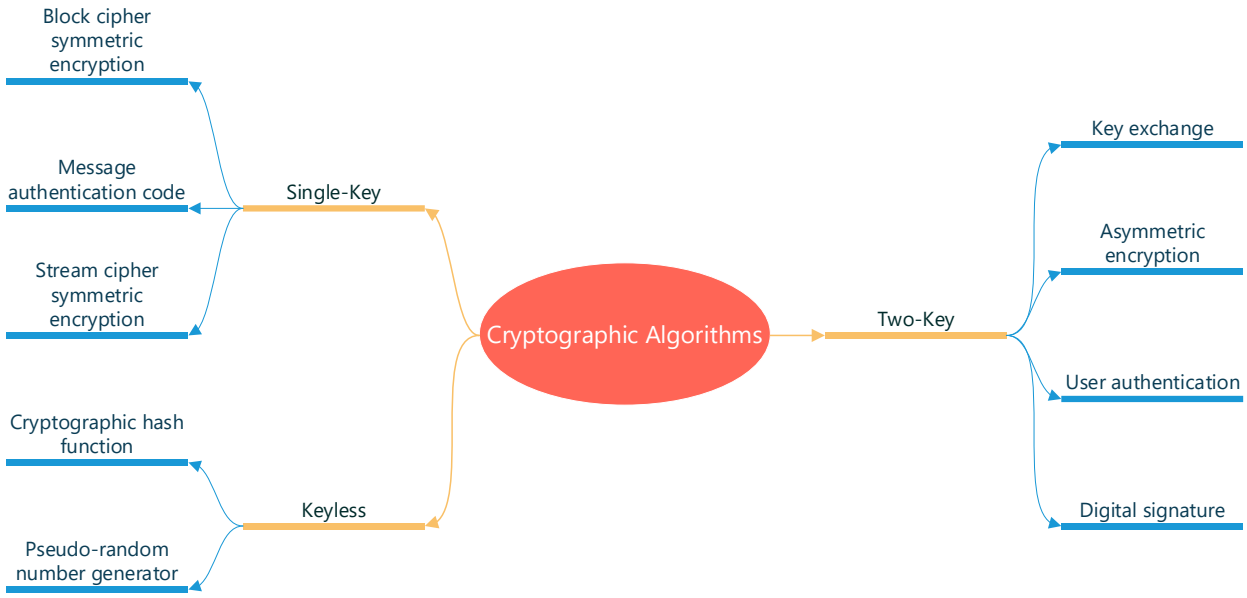


Figure 3-1: Cryptographic algorithms [1]

Cryptographic algorithms can be divided into three categories, as illustrated in Figure 3-1 and described in the list that follows:

- **Keyless:** An algorithm that does not use any keys during cryptographic transformations.

- **Single-key:** An algorithm in which the result of a transformation is a function of the input data and a single key, known as a secret key.
- **Two-key:** An algorithm in which, at various stages of the calculation, two different but related keys are used, referred to as the private key and the public key.

3.1.2.1 Keyless algorithms

Keyless algorithms are deterministic functions that have certain properties that are useful for cryptography.

One important type of keyless algorithm is the cryptographic hash function, which turns a variable amount of given input text into hash value, hash code, or digest that is a fixed-length and small value. A **cryptographic hash function** has additional properties that make it useful as part of another cryptographic algorithm, such as a digital signature.

A **pseudorandom number generator** generates a deterministic sequence of numbers or bits that has the appearance of being a truly random sequence. Although the sequence appears to lack any definite pattern, it will repeat after a certain sequence length. Nevertheless, for some cryptographic purposes, this apparently random sequence is sufficient.

3.1.2.2 Single-key algorithms

Single-key cryptographic algorithms depend on the usage of a secret key, which may be known to a single user; for example, this is the case when protecting stored data that is only going to be access by the data creator. Commonly, two parties share the secret key so that communication between the two parties is protected. For certain applications, more than two users might share the same secret key. In such case, the cryptographic algorithm protects data from those outside the group who share the key.

Encryption algorithms that use a single key are known as **symmetric encryption algorithms**. In particular, an encryption algorithm takes as input some data to be protected and a secret key and produces an unintelligible transformation on that data. A corresponding decryption algorithm uses the transformed data and the same secret key to recover the original data.

Another form of single-key cryptographic algorithm is the message authentication code (MAC). A MAC is a data element associated with a data block or message. The MAC is generated by a cryptographic transformation involving a secret key and, typically, a message cryptographic hash function. The MAC is designed so that everyone possessed the secret key can verify the message´s integrity. Thus, the MAC algorithm generates the MAC taking as inputs a secret key and a message. The recipient of the message plus the MAC can perform the same calculation on the message; if the calculated MAC matches the MAC accompanying the message, this provides assurance that the message has not been modified.

3.1.2.3 Two-key algorithms

Two-key algorithms involve two associated keys. A single user or entity is only in possession of the private key, while the corresponding public key is made available to a number of users. Encryption algorithms that use two keys are referred to as ***asymmetric encryption algorithms***. Asymmetric encryption can work in two ways:

- An encryption algorithm takes as input some data to be protected and the private key and produces an unintelligible transformation on that data. A corresponding decryption algorithm uses the transformed data and the corresponding public key to recover the original data. In this case, only the private key possessor can have performed the encryption, and any possessor of the public key can perform the decryption.
- An encryption algorithm takes as input some data to be protected and a public key and produces an unintelligible transformation on that data. A corresponding decryption algorithm uses the transformed data and the corresponding private key to recover the original data. In this case, any possessor of the public key can have performed the encryption, and only the holder of the private key can perform the decryption.

Asymmetric encryption has a variety of applications. One of the most important is the ***digital signature algorithm***. A digital signature is, typically, a numerical value computed by a cryptographic algorithm and associated with a particular data object in such a way that any recipient of the data signed with this specific digital signature can use it to validate the origin

and integrity of the data received. Usually, the signer of a data object uses the signer's private key to generate the signature, and anyone in possession of the corresponding public key can verify the validity of that signature.

Asymmetric algorithms can also be used in two other important applications. ***Key exchange*** is the process of securely distributing a symmetric key to two or more parties. ***User authentication*** is the process of authenticating that a user attempting to access an application, or a service is genuine and, similarly, that the application or service is genuine. These concepts are explained in detail in subsequent chapters.

3.1.3 Symmetric encryption overview

Symmetric encryption, also referred to as secret-key encryption, is a cryptographic scheme in which both encryption and decryption are performed using the same cryptographic key. A symmetric encryption scheme consists of five ingredients:

1. **Plaintext:** The original data block or message that is fed into the algorithm as input.
2. **Encryption algorithm:** The algorithm that performs various substitutions and transformations on the plaintext.
3. **Secret key:** An input to the encryption algorithm, in which all the exact substitutions and transformations depend on.
4. **Ciphertext:** Refers to the scrambled message output of the algorithm. It depends on both the plaintext and the secret key. So, for a certain data block, two different secret keys will output two distinct ciphertexts.
5. **Decryption algorithm:** The inverse of the encryption algorithm, using the ciphertext and the secret key to generate the original plaintext.

The secure use of symmetric encryption requires the following:

1. Firstly, a strong encryption algorithm is essential. At a minimum, the encryption algorithm should be such that an adversary who has both knowledge of the encryption algorithm, as well as one or more ciphertexts would still not be able to figure out the secret key or decipher the ciphertext.

2. Secondly, the message transmitter and receiver must have obtained, priorly, copies of the secret key in a secure way and must keep the key protected, as if an adversary discovers it and knows the algorithm, he will be able to read all communication using this key.

The generation and distribution of secret keys are essential elements of a symmetric cryptography scheme. Typically, a key generation algorithm generates a random number and derives a secret key from that number. For two parties to communicate, there are a number of possibilities for key distribution, including:

One party generates the key and transfers it to the other party in a secure fashion. The two parties engage in a secure key exchange protocol that enables them to jointly generate a key known only to the two parties. The key is generated by a third party and then is transmitted to the two communicating parties in a secure fashion.

Figure 3-3 illustrates the first alternative. One way to establish a secure channel of communication is if the two parties already share an older secret key, and the party that generates the key can encrypt the new key with the older key. Another alternative is the use of public-key cryptography to encrypt the key. Public-key cryptography is discussed subsequently.

Figure 3-3 also indicates the existence of a potential adversary that seeks to obtain the plaintext. It is assumed that the adversary can eavesdrop on the encrypted data and also knows the encryption and decryption algorithms that were used.

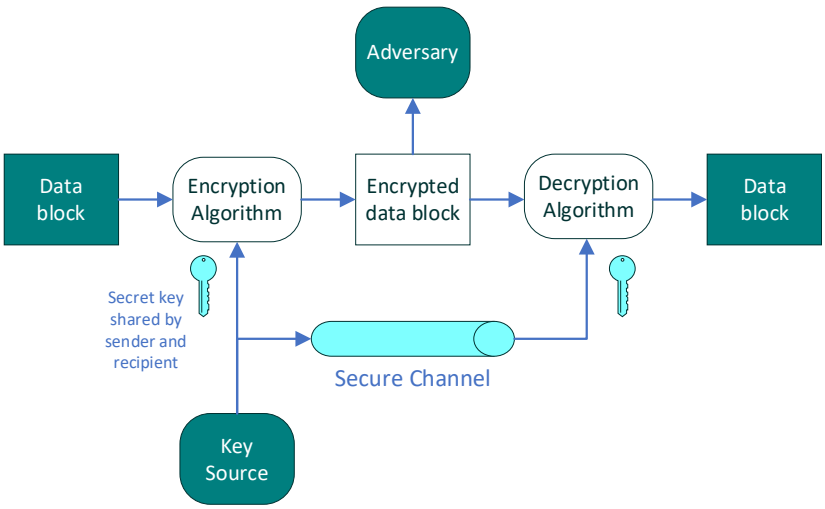


Figure 3-2: Model of a Symmetric Cryptosystem [1]

There are two general approaches an adversary can use to attack a symmetric encryption scheme, as shown in Figure 3-4. One approach is ***cryptanalysis***, in which the malicious actors rely on the algorithm´s nature, as well as on certain knowledge of the general characteristics of the plaintext or samples of the plaintext/ciphertext pairs to attack the symmetric encryption scheme. Cryptanalytic attacks exploit the algorithm´s characteristics in order to deduce a particular plaintext or to figure out the secret key being used. If a cryptanalytic attack succeeds in deducing the secret key, the consequences are catastrophic, as the adversary is able to compromise all future and past communication encrypted with that key. The second approach, referred to as the ***brute-force attack***, includes attempting every possible key on a piece of ciphertext until the adversary obtains an intelligible translation into plaintext. On average, it is essential that half of all possible keys must achieve success. Thus, a secure symmetric encryption scheme requires an algorithm that is secure against cryptanalysis and a key of sufficient length to defeat a brute-force attack.

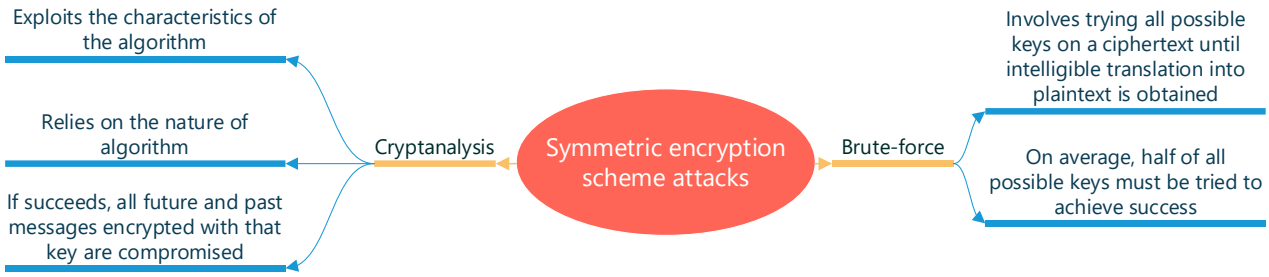


Figure 3-3: Attacks on a symmetric encryption scheme [1]

3.1.3.1 Advanced Encryption Standard (AES)

In 2001, the National Institute of Standards and Technology (NIST) published the **Advanced Encryption Standard (AES)**. AES is a symmetric block cipher that is meant to replace DES as the approved standard for a broad range of applications. Because of the popularity of AES, a significant number of efforts have been made to improve performance in terms of both software and hardware optimization. Most notably, the Advanced Encryption Standard New Instructions (AES-NI) as a hardware extension to the x86 instruction set to improve the speed of encryption and decryption was introduced by Intel in 2008. The AES-NI instruction enables x86 processors to accomplish a performance of 0.64 cycles/byte for an authenticated

encryption mode known as AES-GCM. In 2018, Intel added vectorized instructions, referred to as VAES*, to the existing AES-NI for its high-end processors attempting to push the performance of AES software further down, to a novel theoretical throughput of 0.16 cycles/byte.

In the last years, AES has become the most commonly used symmetric cipher. In general, the symmetric ciphers structure, including the AES, is quite complex and thus, cannot be easily explained as many other cryptographic algorithms, especially as the public-key ciphers such as RSA.

3.1.4 Asymmetric encryption overview

Public-key cryptography, also referred to as ***asymmetric cryptography***, requires the use of two distinct keys, rather than only one key as symmetric encryption uses. The use of two keys introduces profound consequences in confidentiality, key distribution, and authentication security areas. A public-key encryption scheme consists of six ingredients:

Plaintext

- readable data block or message that is the algorithm´s input

Encryption algorithm

- numerous transformations on the plaintext

Public key and private keys

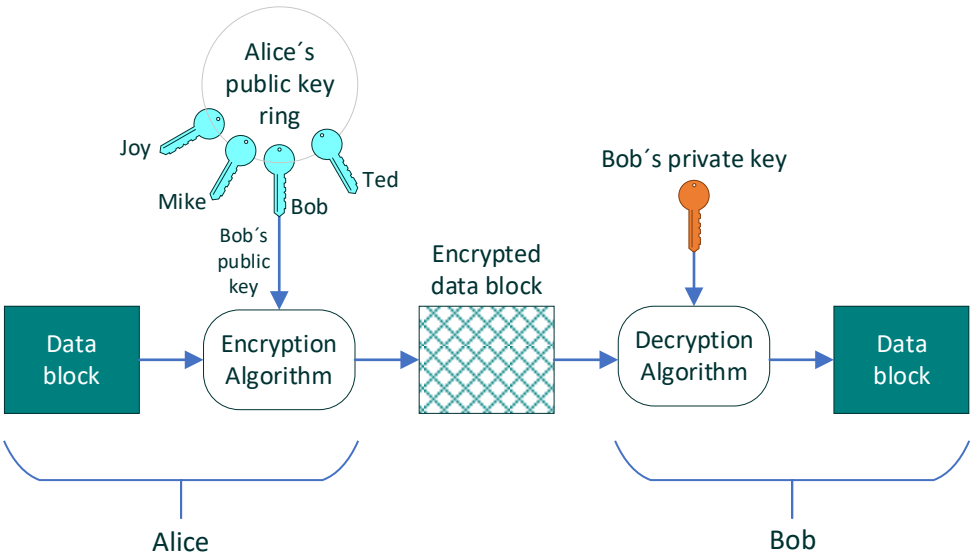
- pair of keys selected so that if one is used for encryption, the other is used for decryption

Ciphertext

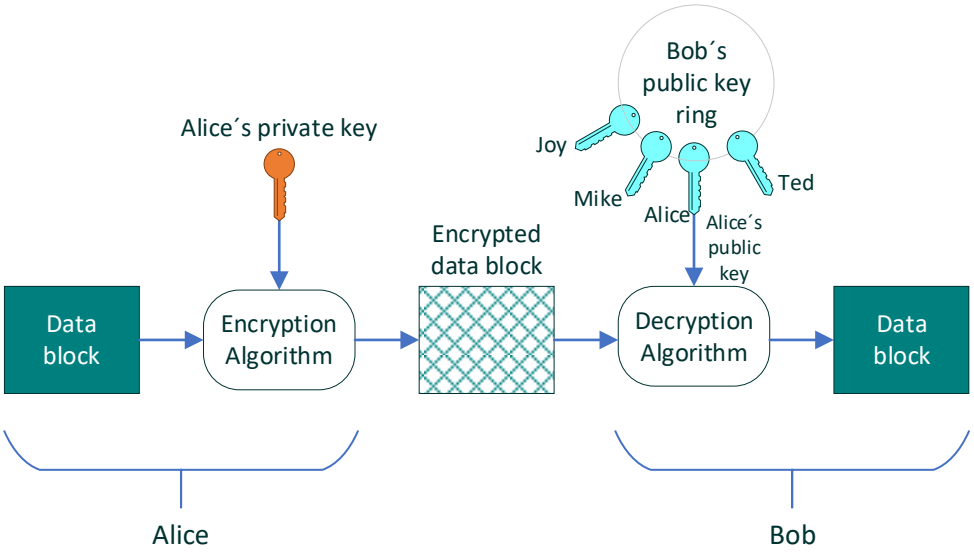
- scrambled block produced as the algorithm´s output. It depends on the plaintext and the key, for a given message, two different keys will produce two different ciphertexts

Decryption algorithm

- the ciphertext and the matching key and generates the original plaintext



(a) Public-key encryption/decryption (Alice encrypts block for Bob only)



(b) Public-key encryption/decryption (Alice authenticates block for any recipient)

Figure 3-4: An Asymmetric Cryptosystem Model [1]

The essential steps shown in part a of Figure 3-4 are as follows:

1. Each communication party generates a pair of keys (i.e., public key and private key) for the encryption and decryption of transmitted messages.
2. Each communication party places the public key in a public register or another accessible file, while his/her companion key is kept private. As part a of Figure 3-4 suggests, each user maintains a *public key ring* where he/she collects public keys from other users.

3. When Alice desires to send a classified message to Bob, she has to encrypt the message using Bob's public key.
4. When Bob receives Alice´s message, he is able to decrypt it, using his private key. We can see that no other recipient would be able to decrypt Alice´s message as only Bob is in possession of his private key.

For any order in which the key pairs are utilized, this process will output the right plaintext. On top of that, all communication parties involved have access to public keys, each participant produces locally and secretly his/her private key without distributing it to others. In this way, incoming communication remains secure as long as users private keys are kept secret and protected.

As with symmetric encryption, asymmetric key generation includes the use of a random number. In this case, the key generation algorithm computes a private key from a random number and then computes a public key as a function of the private key. Without knowledge of the private key, it is infeasible to calculate the public key. On the other hand, knowledge of the public key does not enable calculation of the private key.

Public-key encryption can be also used for the verification of a message´s source, as shown in Figure 3-4b. For instance, if Alice wants to send a message to Bob. The message is not confidential and thus it is not necessary to be kept secret, however Alice wants to ensure Bob that the particular message is indeed from her. Therefore, Alice encrypts the message using her private key. When Bob receives the ciphertext, he realizes that he can only decrypt it with public key of Alice verifying that it must have been Alice who encrypted the message: No one else has Alice's private key, and therefore no one else could have generated a ciphertext that could be decrypted with Alice's public key.

3.1.4.1 Rivest-Shamir-Adleman (RSA) Algorithm

In the **RSA** cipher, the plaintext and ciphertext are integers between *0* and *n* - 1 for some given *n,* typically of size 1024 bits, or 309 decimal digits, that is, *n* is less than 2^{1024}, making use of an expression with exponentials. In particular, plaintext is encrypted in blocks, with each block having a binary value less than some number *n*. In other words, the block size must be less

than or equal to $\log_2(n) + 1$; in practice, the block size is *i* bits, where $2^i < n \leq 2^{i+1}$.Therefore, encryption and decryption expressions, for certain plaintext block *M* and ciphertext block *C*, will be of the following form:

$$C = M^e \bmod n$$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

In such case, the message sender and receiver must know the *n* value. The sender also should know the *e* value, while only the receiver should know the *d* value. Consequently, this constitutes a public-key encryption algorithm with a pair of public and private keys as follows: $PU = \{e, n\}$ and $PR = \{d, n\}$, accordingly.

It is worthwhile to highlight that a satisfactory implementation of the public-key encryption algorithm requires the following:

1. It is possible to get the e, d, and n values such that $M^{ed} \bmod n = M$, for $M < n$.
2. It is reasonably easy to calculate $M^e \bmod n$ and $C^d \bmod n$, for $M < n$.
3. It is infeasible to determine the value *d* given the values *e* and *n*.

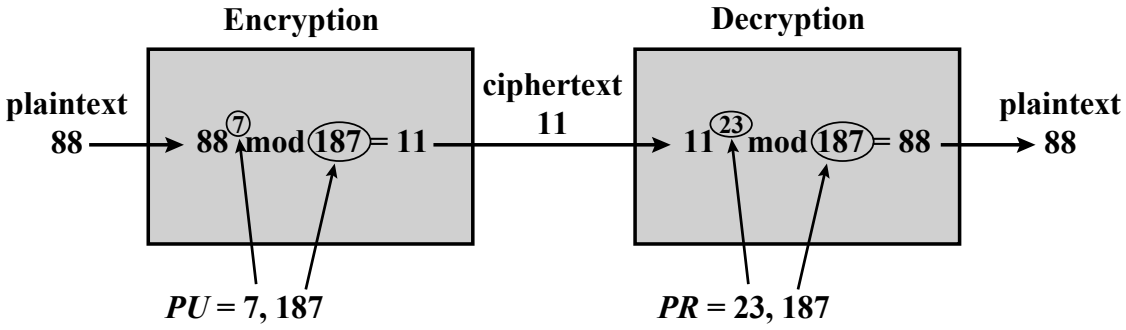


Figure 3-5: Example of RSA Algorithm [1]

A summary of the essential aspects of symmetric and asymmetric encryption are presented in Table 3-1:

Table 3-1: Symmetric and Asymmetric Encryption [1]

Symmetric Encryption	Asymmetric Encryption
Needed to Work: 1. The same algorithm with the same secret key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the secret key. Needed for Security: 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if the key is kept secret. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key	Needed to Work: 1. One algorithm is used for encryption and a related algorithm for decryption with a pair of keys, known as the public key and the private key. The two keys can be used in either order, one for encryption and one for decryption. 2. The sender and receiver must each have a unique public/private key pair. Needed for Security: 1. The private key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if the private key is kept secret. 3. Knowledge of the algorithm plus the public key plus samples of ciphertext must be insufficient to determine the private key.

3.1.5 Cryptographic Hash Functions

A cryptographic hash function gets an input of arbitrary length and maps it to a fixed-length data block that is typically shorter than the input data block. This is therefore a many-to-one function; that is, multiple input blocks produce the same output. The output is also referred to as the hash value or hash digest. A cryptographic/secure hash function is a hash function with specific properties that are useful for various cryptographic algorithms, as explained subsequently. Secure hash functions are an essential component of many security protocols and applications.

A hash function called H must present the following properties: (a) Variable input size, (b) Fixed output size, (c) Efficiency in terms of relatively easy computations for practical implementations, (d) Preimage resistant, or one-way property, such as for any given hash value h, to be computationally infeasible to compute H(y)=y, (e) Second preimage resistant, or weak collision resistant, (f) Collision resistant, or strong collision resistant, and (g) Pseudorandomness, in order to be considered useful for security applications:

The two crucial facets of message authentication are to verify that: (i) the message´s contents have not been modified, and (ii) the message´s source is authentic and trustable. The hash function can also verify a message's timeliness ensuring that the message has been artificially

delayed and replayed, and sequence relative to other messages flowing between two parties by including timestamps and sequence numbers in the message.

For message authentication, firstly, a hash value is generated for the message´s source. Then, this hash value is encrypted with the use of a secret key shared by a cooperating communication party. Then, the encrypted hash value and the message are transmitted to the destination. The incoming encrypted hash value can be decrypted by the recipient. Then, the recipient creates a new hash value from the incoming message and compares the two hash values. If only the message´s receiver and sender are in possession of the secret key´s identity, and if the hash value received matches the generated one, then:

- The recipient verifies that the message´s content has not been modified. If a malicious actor modifies only the message but not the hash value, then the recipient's calculation of the hash value will not match the received one. For a secure hash function, it is infeasible for an attacker to alter the message in such a way that the hash value is not altered.
- The recipient verifies that the message is from the claimed sender - no one else could send the message with a correct hash value since no one else has knowledge of the secret key.
- In case that a sequence number is included in the message includes (as in TCP), then the recipient of the message can verify that this is the proper sequence.

3.1.6 Implementation considerations

SP 800-12 lists the following as important management considerations for implementing cryptography within an organization:

1. **Selecting design and implementation standards:** It is almost always advisable not to rely on a proprietary cryptographic algorithm, especially if the algorithm itself is secret. Standardized algorithms, such as AES, SHA, and DSS, have been subject to intense scrutiny by the professional community, and managers can be assured that the algorithms themselves, used with the recommended lengths, are secure. Based on cost-effectiveness analysis, trends in the standard's acceptance, and

interoperability requirements, managers, team leaders and users of information systems should select the suitable cryptographic standard for them.

2. **Deciding between hardware, software, and firmware implementations:** Managers and users should study the trade-offs among security, cost, simplicity, efficiency, and ease of implementation in order to acquire various security products meeting the standards.
3. **Managing keys:** Key management is the process of administering or managing cryptographic keys for a cryptographic system or application including key servers, user procedures, and protocols.
4. **Security of cryptographic modules:** A cryptographic module contains certain control parameters, the cryptographic algorithm(s), as well as temporary storage facilities for the key(s) being utilized by the cryptographic algorithm(s). A useful tool is the NIST Cryptographic Module Validation Program (CMVP), which validates vendor offerings using independent accredited laboratories. The validation is against the security requirements in FIPS 140-2 (*Security Requirements for Cryptographic Modules*). FIPS 104-2 provides a detailed set of requirements at four security levels, against which vendor hardware, firmware, and software offerings can be evaluated.

3.2 Public key infrastructure (PKI) & Public-Key Certificates

A public-key infrastructure (PKI) enables users and digital devices to exchange data over networks securely making use of public encryption keys. A public-key certificate constitutes actually a set of data that uniquely identifies an entity. In particular, the certificate contains the entity's public key and other data and is signed in a digital fashion by a trusted entity, also refer to as ***certification authority***, in this manner binding the public key to the certain entity.

Public-key certificates are designed to provide a solution to the problem of public-key distribution. Typically, in a public-key scheme, multiple users need to have access to the public key of a given entity A, whether to encrypt data to send to A or to verify a digital signature signed by A. Each holder of a public/private key pair could simply broadcast its public key for anyone to read. The problem with this approach is that it would be easy for some attacker X to impersonate A and to broadcast X's public key improperly labeled as A's public key. To

counter this, it would be possible to set up some trusted central authority that would interact with each user to authenticate and then maintain a copy of A's public key. Any other user could then consult the trusted central authority over a secure, authenticated communication channel to obtain a copy of the key. It should be clear that this solution would not scale efficiently.

An alternative approach is to rely on public-key certificates that participants can use to exchange keys directly, in such way that is as trustworthy as if the keys were acquired directly from a public-key authority. In principle, a certificate comprises of a public key plus a key owner identifier, and the whole data block is signed by a trusted third party. The third party is, typically, a ***certification authority*** (CA) that is trusted by the user community. For instance, a government agency or a financial institution can be considered as CAs. A user may present his/her public key to the trusted authority in a secure fashion and obtain a certificate, that then can publish. So, any entity needing user's public key can acquire this certificate and verify its validity by the attached trusted signature.

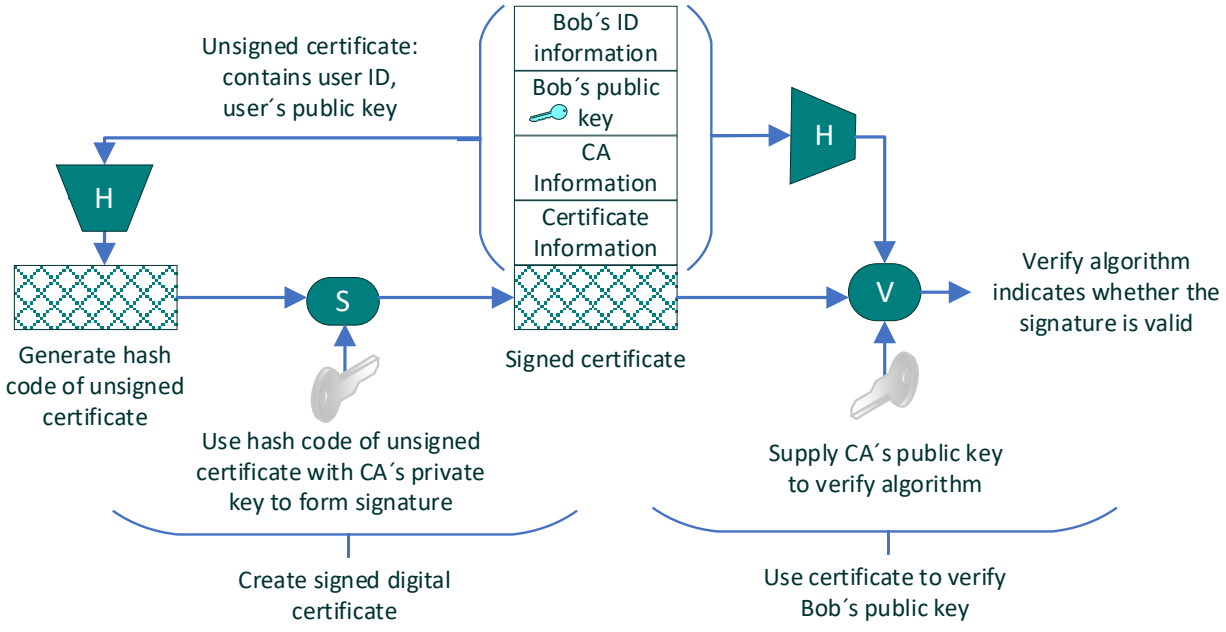


Figure 3-6: Public-Key Certificate Use [1]

Figure 3-6 demonstrates the overall scheme for generation of a public-key certificate. The certificate for Bob's public key comprises of unique identifying information for Bob, Bob's public key, identifying information about the CA, and certificate information, as well as

expiration date. This information is then signed by computing a hash value of the information and generating a digital signature using the hash value and the CA's private key. Bob can then either broadcast this certificate to other users or attach the certificate to any document or data block he signs. Anyone who needs to use Bob's public key can be assured that the public key contained in Bob's certificate is valid because the certificate is signed by the trusted CA.

A PKI architecture defines the organization and interrelationships among CAs and PKI users. PKI architectures satisfy the following requirements:

1. Any communication party is able to read a PK certificate to define the certificate's owner name and public key.
2. Any communication party is able to assure that the PK certificate originated from a legitimate certification authority and is not counterfeit.
3. Only the certification authority has the ability to generate and update certificates.
4. Any communication party can be assured that the certificate is currently valid.

Essential components of PKI Architecture:

End entity

- A process, an end user, a device, or any item that can be identified in the subject name of a public-key certificate, as well as consumers of PKI-related services and providers of PKI-related services.

Certification authority (CA)

- An authority trusted by one or more communication parties to generate and assign public-key certificates, and optionally the keys of the subjects. CAs digitally sign public-key certificates, which effectively binds the subject names to the public keys. CAs are also responsible for issuing certificate revocation lists (CRLs). A CRL identifies certificates previously issued by the CA that are revoked before their expiration date. A certificate could be revoked if: (i) the user is no longer certified by this CA, (ii) the private key of the user is assumed to be compromised, or (iii) the certificate is assumed to be compromised.

Registration authority (RA)

- An optional component associated with the end entity registration process that can be used to offload many of the administrative functions that a CA ordinarily assumes. This includes the verification of the identity of the end entity attempting to register with the PKI and obtain a certificate for its public key.

Repository

- Any method used for storing and retrieving PKI-related information, such as public-key certificates and CRLs.

Relying party

- Any communication party, user or agent that relies on the information in a certificate in making decision processes.

Figure 3-7 provides a typical architecture for a PKI.

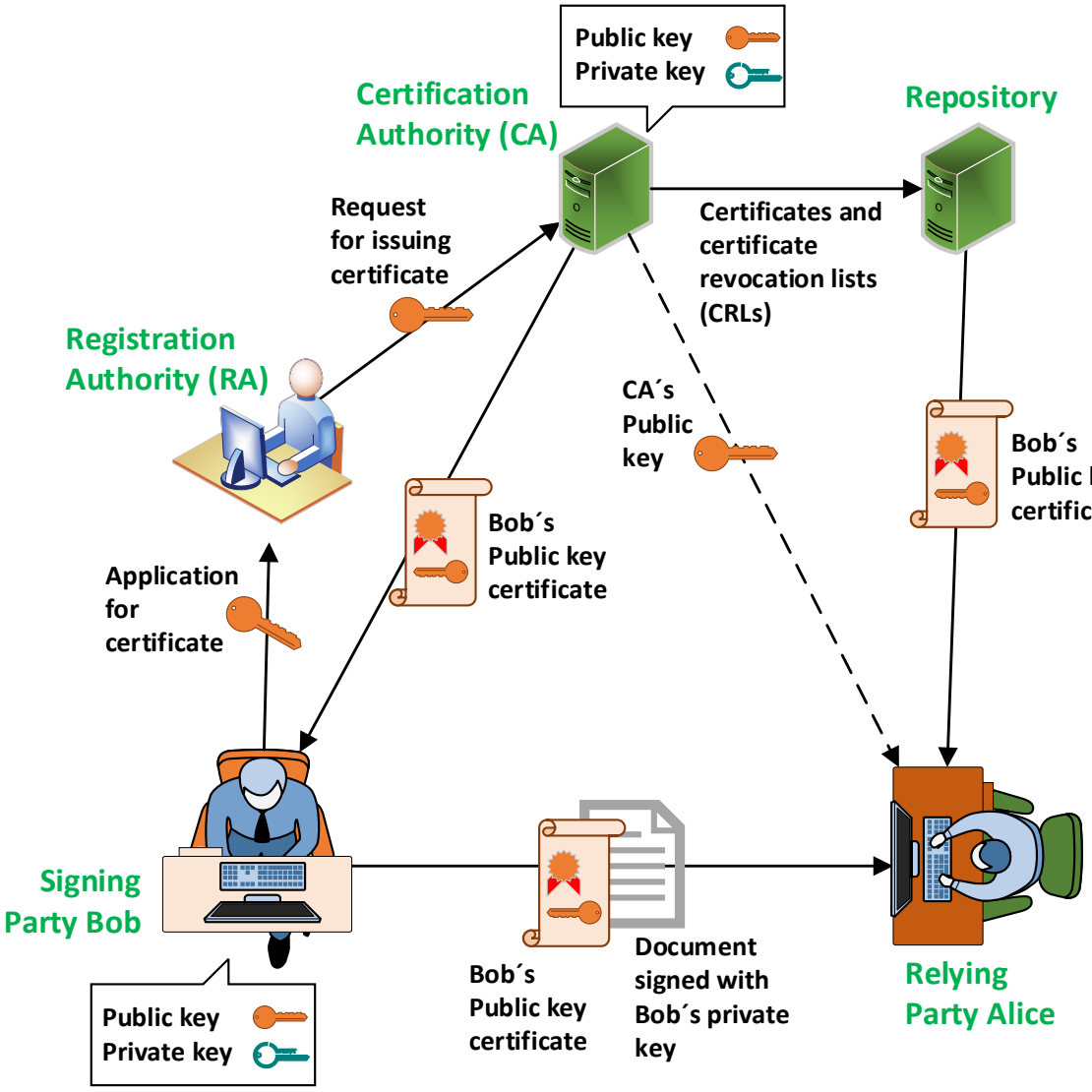


Figure 3-7: PKI Scenario [1]

Figure 3-7 illustrates the interaction of the various components. Consider a relying party Alice that needs to use Bob's public key. Alice must first obtain in a reliable, secure fashion a copy of the public key of the CA. This can be done in a number of ways, depending on the particular PKI architecture and enterprise policy. If Alice wishes the send encrypted data to Bob, Alice checks with the repository to determine whether Bob's certificate has been revoked, and if not, she obtains a copy of Bob's certificate. Alice can then use Bob's public key to encrypt data sent to Bob. Bob can also send to Alice a document signed with Bob's private key. Bob may include his certificate with the document or assume that Alice already has or can obtain the certificate. In either case, Alice first uses the CA's public key to validate that the certificate´s

validity and then uses Bob's public key (obtained from the certificate) to validate Bob's signature.

Rather than using a single CA, an enterprise may need to rely on multiple CAs and multiple repositories. CAs can be organized in a hierarchical fashion, with a root CA that is widely trusted signing the public-key certificates of subordinate CAs.

3.3 Key management

This section presents a short introduction to key establishment and key management for secure keys distribution for cryptographic applications and purposes. Key establishment is the procedure that allows a secret key to be shared with two communication parties or more, for cryptographic usage. On the other side, key management is the set of processes and mechanisms which support key establishment and the maintenance of ongoing keying relationships between parties, including replacing older keys with new keys as necessary.

3.3.1 Key establishment

In general, key establishment can be subdivided into key agreement and key transport processes. When considering a network of entities, any two of which may wish to communicate, as shown in Figure 3-9, it is evident that when using symmetric-key techniques a major concern is the establishment of pairwise secret keys to all participating entities. In Figure 3-9, we can observe a network consisting of 6 communicating parties with the arrowed edges to show the 15 possible two-party communications which could possibly take place. Assuming that each pair of entities wish to communicate, this small network requires the secure exchange of $\binom{6}{2} = 15$ key pairs. This means that in a network with *n* entities, the number of secure key exchanges required is $\binom{n}{2} = \frac{n\,(n-1)}{2}$.

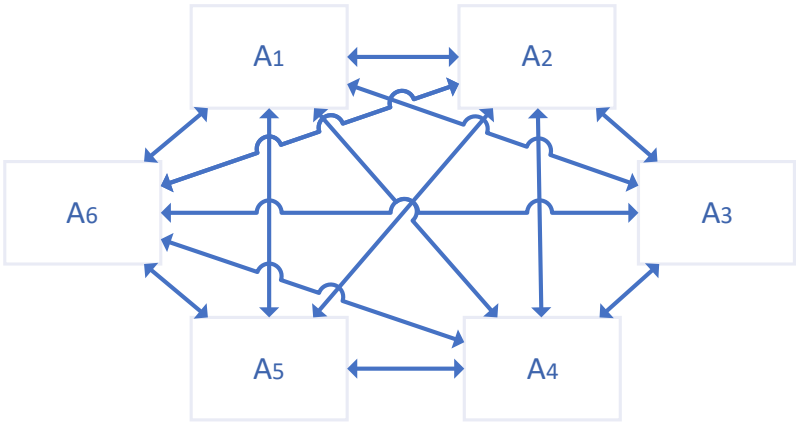


Figure 3-8: Keying relationships in a 6-party network [2]

The network diagram shown in Figure 3-8 is a simple network of 15 two-party communications. However, in practice, networks are very large, and as a consequence the key management is becoming a crucial and challenging issue. There are several ways to address the key management problem. Two simplistic techniques are examined in the rest of this section: (i) key management based on symmetric-key and (ii) the key management based on public-key techniques.

3.3.2 Key management using symmetric-key techniques

The first solution to address the complex key management problem employs symmetric-key techniques. This technique involves in the network an entity trusted by all communication parties, typically known as trusted third party (TTP), as depicted in Figure 3-10. Then, every communication party A_i shares a distinct symmetric key k_i with the TTP over a secured channel. When two communication parties subsequently wish to communicate, the TTP generates a key k, also referred to as session key, and sends it encrypted under each of the fixed keys.

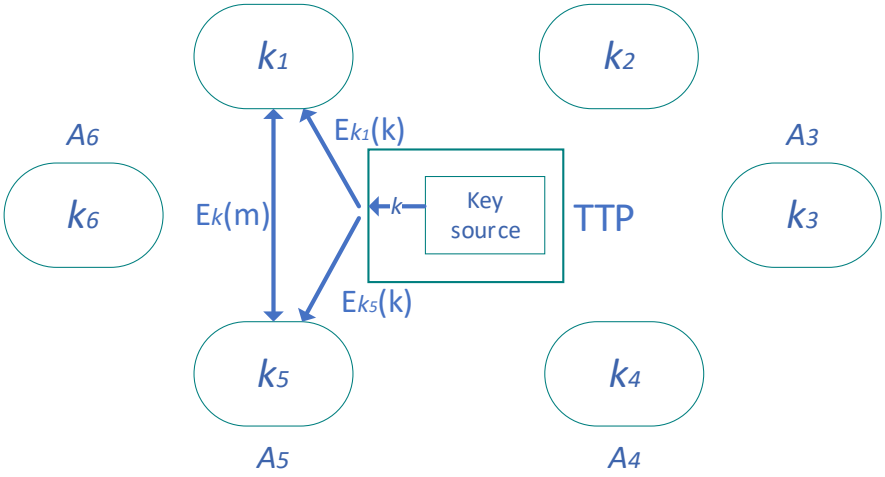


Figure 3-9: Key management using a TTP [2]

Advantages of this approach include the easiness to add and remove communication parties from the network, as well as that every communication party needs to store only one long-term secret key. On the other side, the main disadvantages of this approach include that all communications require an initial interaction with the TTP. In addition, the TTP must store all *n* long-term secret keys, while it can read all transmitted communications. Finally, in case that the TTP is compromised, then all communications – past and future - are insecure.

3.3.3 Key management using public-key techniques

Although there are several ways to address the key management problem through public-key techniques, for the purpose of this lecture we considered a very basic model.

Advantages of the key management using public-key techniques include that there is no need for a trusted third party in the network. In addition, the public file could reside with every communication entity. Finally, only *n* public keys need to be stored to allow secure communications between any pair of entities, assuming the only attack is that by a passive adversary. The key management problem becomes more complicated when the adversary is active. This is because an active adversary could compromise the key management scheme altering the public file containing the public keys.

To counteract active attacks, the communication parties may use a TTP for the certification of the public key of each communication party using a signing algorithm ST and a verification algorithm VT assumed to be known by all communication parties. The TTP carefully validates the identity of each entity, and signs a message consisting of an identifier and the entity's authentic public key. This is a simple example of a certificate, binding the identity of an entity to its public key.

Advantages of using a TTP to maintain the integrity of the public file include that it successfully prevents an active adversary from impersonation legitimate participants on the network. In addition, the TTP cannot monitor communications, while entities need trust the TTP only to bind identities to public keys properly. Finally, communication interactions between the communication parties and the public file can be eliminated if communication parties store certificates locally. However, similarly to key management using symmetric key techniques, if the signing key of the TTP is compromised, then all future and past communications become compromised and insecure. This happens as all trust is placed with one entity – the TTP.

3.4 Key management lifecycle

Key management life cycle is known as the sequence of states which keying material passes through over its lifetime. Figure 3-10 depicts the key management life cycle stages that generally may include:

1. user registration
2. user initialization
3. key generation
4. key installation
5. key registration
6. normal use
7. key backup
8. key update
9. key archival
10. key de-registration and destruction
11. key recovery

12. key revocation

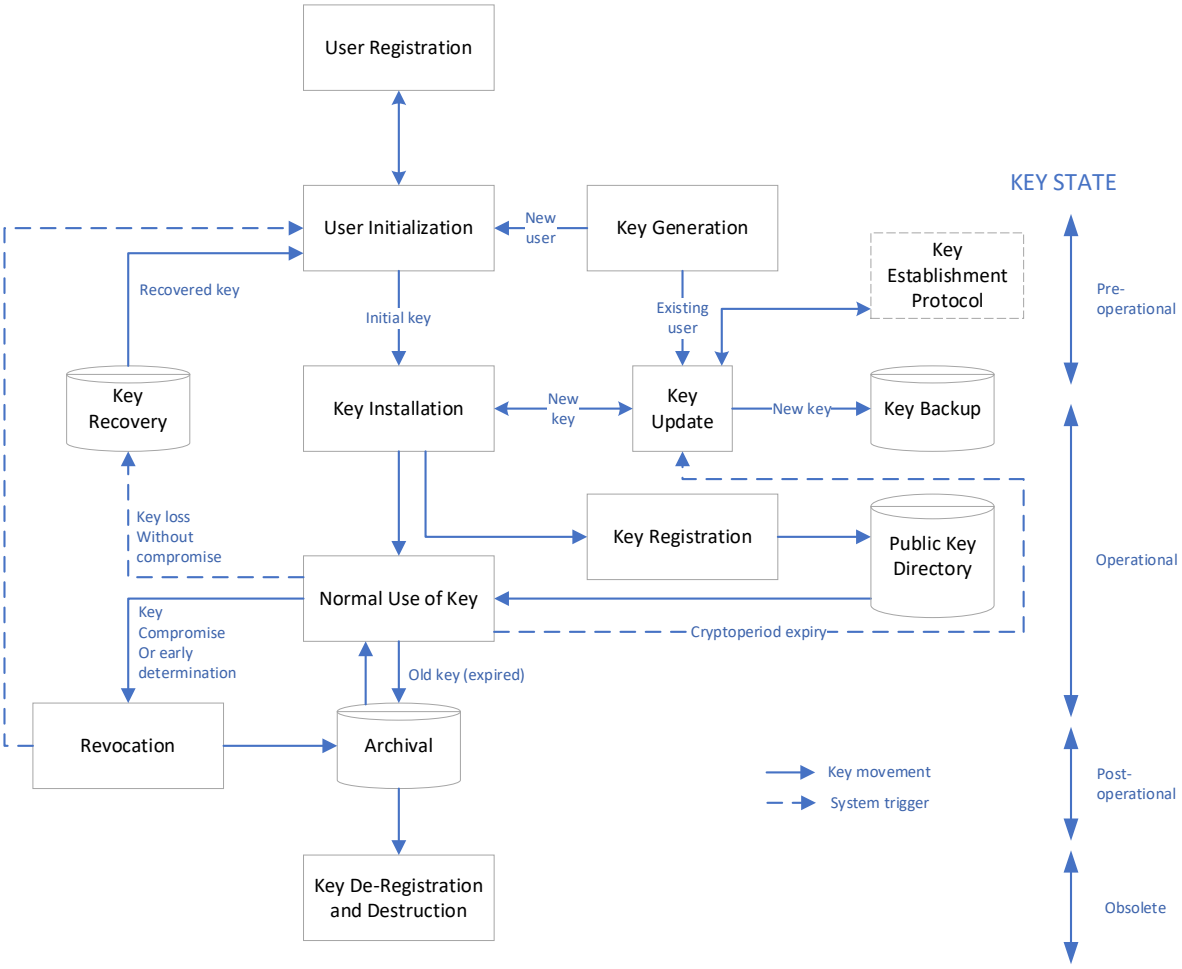


Figure 3-10: Key management life cycle [2]

It is worthwhile to mention that all the above stages are regularly scheduled and performed, except the points 11 and 12 about key recovery and key revocation which are only performed under special situations that require so.

3.5 Referencing

[1] “Computer Security: Principles and Practice”, 4/e, by William Stallings and Lawrie Brown

[2] “Computer Security: Art and Science”, 2/e, by Matt Bishop

[3] “Computer Security Fundamentals”, by Chuck Easttom, 2020

[13] “Information Privacy Engineering and Privacy by Design”, 1/e, by William Stallings

[15] “Cryptography and Network Security”, 8/e, by William Stallings

[14] “Handbook of applied cryptography”, by Menezes AJ, Katz J, Van Oorschot PC, Vanstone SA

4 Security Engineering: Cryptography Services

Author(s): Dr Eliana Stavrou

4.1 Introduction

This chapter will cover fundamental knowledge in Cryptography services. Specifically, the chapter introduces the following topics:

- Hashing
- Digital signatures
- Digital Rights Management (DRM) technologies
- Cryptanalytic and social engineering attacks

4.2 Hashing

In this section, the focus is on covering hashing fundamentals. Specifically, we will discuss what hashing is, its properties, and where it can be applied. Moreover, we will briefly discuss well-known cryptographic hash functions, such as MD5, SHA-1, SHA2, SHA-3, etc. The usage of hashing will be demonstrated through an example, to perform password verification. Then the usage of salting and its importance will be explained in the context of the password verification example.

4.2.1 What is hashing

Hashing is the transformation of a message of any length to a fixed-length value that is called the hash value or a message digest. The message digest serves as the fingerprint of the message that can uniquely identify the original message. The main objective of hashing is to achieve integrity verification. For example, when a message is exchanged between two communicating parties, the receiver must be able to confirm that the message received has not been altered, meaning that it retains its integrity.

Figure 4-1 presents an example to better understand the concept of hashing. As indicated in the figure, we have two communicating parties, A and B. A construct the following message to send to B: “You have to pay 100 Euros for the product”. Then A uses a hash algorithm on the message and generates the message digest 5409835. An attaches the message digest to the original message and sends it to B. During transmission, an attacker modifies the message

to "You have to pay **1000** Euros for the product". Once the message and attached message digest are received by B, he applies the same hash algorithm on the received message and produces the message digest 9827765. Then he compares the newly generated message digest with the one attached to the message and observes that the two values do not match. This means that the received message is not the original sent by A. B does not consider the received message as valid and notifies A to resend it.

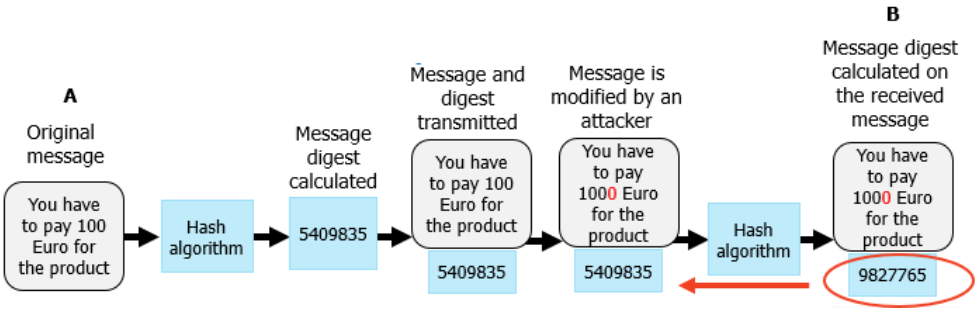


Figure 4-1: Hashing process

As mentioned before, a hash function takes as input a message of any length and produces a fixed-length message digest. Figure 4-2 presents the message digests generated when the SHA-1 hash function is applied on the respective input messages. As it can be observed, even the slightest variation in the input will create a totally different hash value. Another observation that can be made is that the input cannot be inferred back by analysing the message digest.

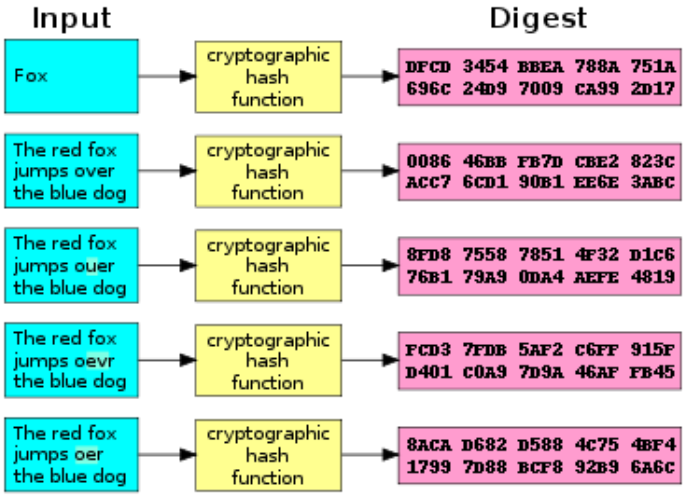


Figure 4-2: SHA-1 example
(https://en.wikipedia.org/wiki/Cryptographic_hash_function)

4.2.2 Hashing properties

Consider the following key properties that should be achieved by an ideal cryptographic hash function:

a) It is deterministic. This means that each time you hash the same message, the result (message digest) is the same.
b) It is one way. You can generate a hash given a message, but you cannot revert the process and generate a message given a hash.
c) Collision resistance. This means that you cannot find two different messages that when hashed, result to the same message digest.
d) A small change to a message should change the hash value to a great extend to prohibit someone to be able to correlate the hash values.

4.2.3 Application of hashing

Hashing finds applicability in different cases. Examples where hashing can be utilized are briefly explained below.

a) To verify the integrity of message and files

The main aim of hashing is to verity the integrity of messages and files. This can be achieved by comparing message digests of transmitted messages/files before and after transmission. The comparison can reveal if any alterations have occurred to the message/file. If the provided message digest before transmission does not match the digest created after the message is received, then this means that the received message is not the same as the original message.

Given that electronic resources, e.g., software, documents, etc., can be downloaded from the internet, we need a method to verify their integrity. To achieve this, hash digests are often published on websites to verify the integrity of the downloaded files. Users can download the hash digest, recompute it from the resource that they download and then compare the digests if they match.

b) Signature generation and verification

Digital signature schemes (discussed later in the lecture notes) utilize cryptographic hash functions to verify the message authenticity.

c) Password verification

Password verification relies on cryptographic hashes to verify that the provided password is the correct one.

4.2.4 Cryptographic hash functions

Examples of well-known hash functions include the following:

- MD5

MD5 was designed by Ronald Rivest in 1991. This is a hash algorithm that results to a message digest of 128 bits. It was found to be subject to collisions and therefore should not be used.

- RIPEMD-160

RIPEMD (RACE Integrity Primitives Evaluation Message Digest) was first published in 1996. It is based on the design principles of MD4. RIPEMD-160 results to a hash digest of 160 bits.

- SHA

SHA (Secure Hash Algorithms) is a family of cryptographic hash functions that includes the following: SHA-0, SHA-1, SHA-2, and SHA-3. SHA was published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS).

- SHA-0

SHA-0 is considered a 160-bit hash function (results to a 160-bit message digest). It was published in 1993 but it was withdrawn shortly after publication and replaced by SHA-1. SHA-0 is published as FIPS PUB 180.

- SHA-1

SHA-1 was designed by the United States National Security Agency (NSA) and it was published in 1995. This hash function was part of the Digital Signature Algorithm. It produces hash

outputs of 160 bits. This hash function should not be utilized as cryptographic weaknesses were discovered. SHA-1 is published as FIPS PUB 180-1.

- SHA-2

SHA-2 includes a set of cryptographic hash functions designed by the NSA that were first published in 2001. They basically consist of three hash algorithms: SHA-256, SHA-384 and SHA-512. They are published under FIPS PUB 180-2 that includes the following hash algorithms: SHA-1, SHA-256, SHA-384, and SHA-512.

- SHA-3

Secure Hash Algorithm 3 was released in 2015 by NIST. SHA-3 generates message digest sizes of: 224, 256, 384 and 512 bits.

4.2.5 Example: Password verification

As mentioned above, one application of hashing is to perform password verification.

We use passwords to get authenticated and have access to a system. The provided passwords are usually not stored on a system as clear text, rather they are stored in a coded form. The encoding is performed by a hash algorithm (also called one-way cipher). The hash algorithm is applied on the provided password and produces a fixed length string. The actual password can be of any length. The hash value or message digest of the provided password is stored on the system to be able to verify users when they login into the system.

Figure 4-3 presents how the password verification is performed, for example when logging into a computer. Initially, the user registers for an account to the system and provides his password. The password is hashed and stored on the system. In a subsequent login try, the user provides his username and password in order to access his account. The system computes the hash over the provided password and then compares it with the hash value stored on the system. If the two values match, it means that the user provided the correct password and he is granted access to the system. Otherwise, the system will notify the user that the password is not the correct and that it needs to be inserted again.

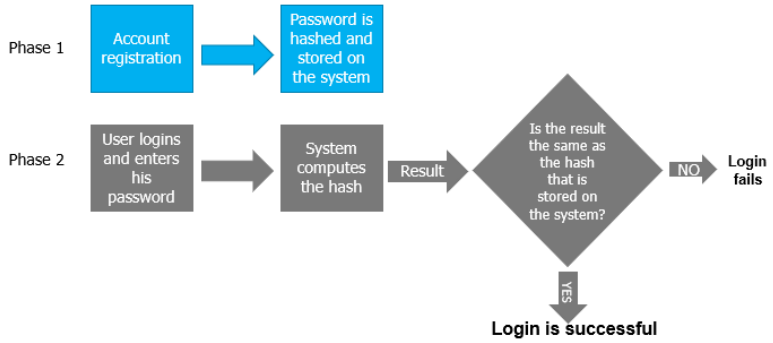


Figure 4-3: Password verification process

4.2.6 Salting

What happens if we have two users that choose the same password? If two users choose the same password, the respective hashed password will be the same. That would be an issue as it means that hackers can easily identify a user's password. In order to overcome this situation, a salt is being utilized. A salt is a random generated number which is added to the password before applying the hash algorithm. The objective of a salt is for the hashing output to be different even if passwords are identical. The salt is stored alongside the hashed password so it can be retrieved and utilized during password verification. The salt should not be reused though to maintain the uniqueness of the hash value.

Figure 4-4 demonstrates how the password verification is amended to include the usage of a salt. As mentioned earlier, initially, the user registers for an account to the system and provides his password. A salt is added to the password and then the (password + salt) value is hashed and stored on the system. In a subsequent login try, the user provides his username and password in order to access his account. The system retrieves the stored salt, appends it on the provided password and then it computes the hash over the provided password and salt. The generated message digest is then compared with the hash value stored on the system. If the two values match, it means that the user provided the correct password and he is granted access to the system. Otherwise, the system will notify the user that the password is not the correct and that it needs to be inserted again.

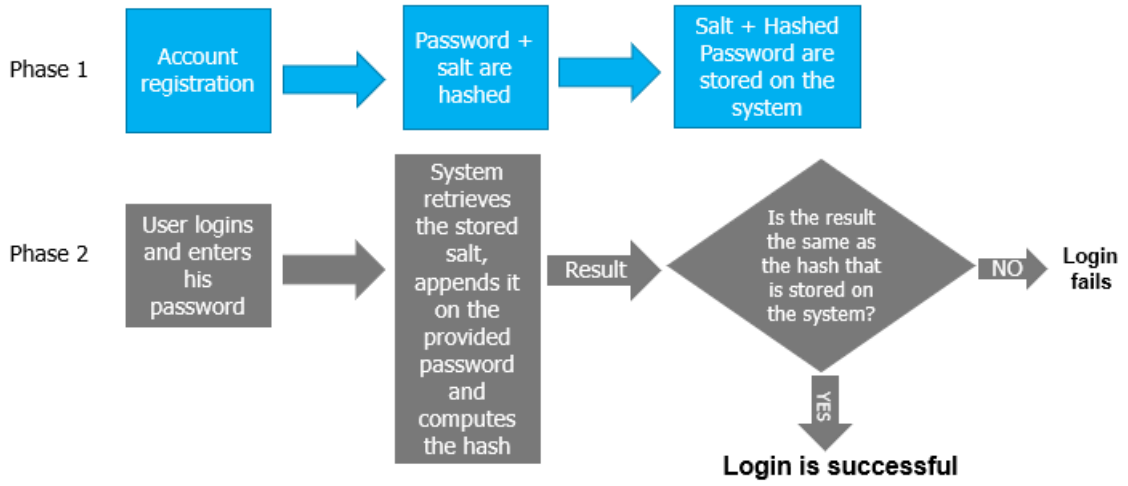


Figure 4-4: Password verification process with salt

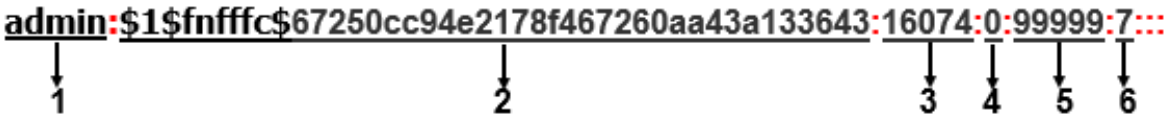


Figure 4-5: Linux hashed password file format

The numbering included on Figure 4-5 should be referenced alongside the following explanation:

1. Username: Login name of user
2. Password "idsalt$hash(password+salt)", where "$id" is the algorithm used ($1 stands for MD5)
3. This indicates when the password was last changed, starting from January 1st, 1970
4. This indicates the minimum number of days the user is allowed to change his/her password since the previous time he/she has performed a password change
5. Once the user has changed his/her password, this indicates the maximum number of days the password is valid
6. Indicates the number of days, before password is expected to expire, where a warning will be issued to the user to change his/her password
7. Indicates the number of days after password expires that account is disabled (here is not set)

8. When the account gets disabled, starting from January 1st, 1970 (here is not set)

4.3 Digital signatures

In this section, the topic of digital signatures is discussed. Specifically, this section explains the asymmetric encryption process and the security properties that can be achieved. Then it briefly discusses typical examples where digital signatures are utilized. Finally, the signing and verification process are explained in detail.

4.3.1 Introduction

NIST FIPS 186-4 (Digital Signature Standard) defines a digital signature as follows: "The result of a cryptographic transformation of data that, when properly implemented, provides a mechanism for verifying origin authentication, data integrity, and signatory non-repudiation".

The original data and the associated digital signature are utilized to verify that: 1) The data have been signed by the owner of the digital signature, and 2) The data have not been altered after the data signing. Furthermore, the signer cannot repudiate digitally signing the data, given that the secret key utilized to create the digital signature is known only by the signer.

Digital signatures are implemented through asymmetric cryptography. Asymmetric cryptography uses two mathematically related keys. Data encrypted with the public key can only be decrypted with the private key. The public key can be shared with anyone that would like to communicate securely with the owner of the corresponding private key. The private key is a secret key known only by its owner. In the case that the secrecy of the private key is compromised, the security of the communication is at risk as anyone that knowns the private key can digitally sign messages pretending to be the key owner. Asymmetric cryptography is also called public key cryptography (PKI). For digital signatures, NIST recommends three alternative digital signature algorithms: Digital Signature Algorithm (DSA) with length of 2048 bits, RSA algorithm with 2048 bits or Elliptic-Curve Digital Signature Algorithm with length of 224 bits.

Digital signatures are equivalent to traditional handwritten signatures. They are utilized to verify the authenticity of digital messages or documents, their integrity and non-repudiation of origin.

4.3.2 Asymmetric encryption process

As mentioned earlier, the digital signatures are implemented through asymmetric encryption. First, the asymmetric encryption process is explained to enhance the understanding of the readers on the fundamental asymmetric encryption concepts. The following scenario (Figure 4-6) demonstrates the asymmetric encryption process:

1. Bob wants to communicate with Alice and send her a message
2. He uses Alice's public key to encrypt the message and transmits it to Alice
3. Once Alice receives the encrypted message, she uses her private key to decrypt it and read the plaintext message from Bob

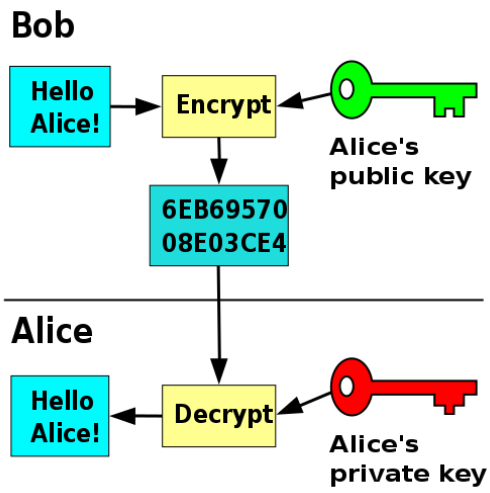


Figure 4-6: Asymmetric encryption process
https://en.wikipedia.org/wiki/Digital_signature

4.3.3 Security properties

Digital signatures support two security goals:

a) Non-repudiation

The receiver of a digitally signed message can verify that the signed message originated from the claimed owner of the private key utilized for the signing. The sender cannot later deny that he did not send the message.

b) Integrity

A message can be altered by an attacker. Also, unintentional modifications can occur, for example, due to communication errors while a message is in transit. With digital signatures, the recipient can verify the integrity of a message.

4.3.4 Typical usage

Digital signatures find applicability in different cases. Some typical examples of digital signatures usage include the following:

a) Software distribution

Software is digitally signed by the vendor to protect from modifications and also as a proof of the software's origin.

b) Financial transactions

A contract is digitally signed and the recipient party can verify its integrity and the sender's identity.

c) Email communication

An email is digitally signed so the recipient can verify that the sender is a trusted entity.

4.3.5 Signing and verification process

There are two key tasks related to the digital signatures, signing and verification.

In terms of the signing process, let's consider that Alice wants to communicate with Bob and send him a digitally signed message. To do so, she performs the following actions (Figure 4-7):

1. Alice applies a hashing algorithm on the plaintext message to send to Bob and generates the corresponding message digest
2. Then she encrypts the message digest with her private key and generates the digital signature which is appended to the original message
3. Alice transmits the digital signature and the original message to Bob

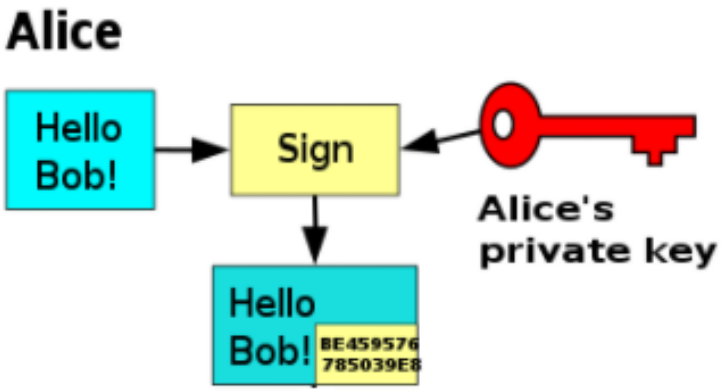


Figure 4-7: Digitally signing a message
https://en.wikipedia.org/wiki/Digital_signature

Bob received the digitally signed message. Then he must verify the authenticity and integrity of the message (Figure 4-8). He can achieve this by doing the following:

1. Bob decrypts the digital signature using Alice's public key. He gets back the decrypted message digest that was originally generated by Alice
2. Bob applies the same hashing algorithm (utilized by Alice) on the received plaintext and generates the corresponding message digest
3. The newly generated message digest is compared with the one Bob has decrypted. If the digests match, then this means that the message integrity can be confirmed and that it was sent by Alice

Note that a public key can only decrypt a message encrypted with the corresponding private key.

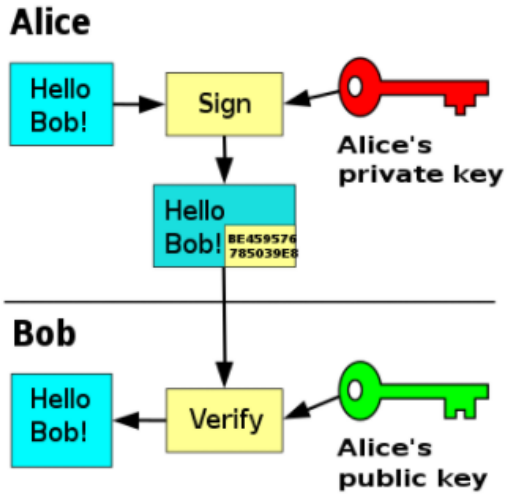


Figure 4-8: Verifying validity of digital signature
https://en.wikipedia.org/wiki/Digital_signature

Figure 4-9 presents the overall signing and verification process.

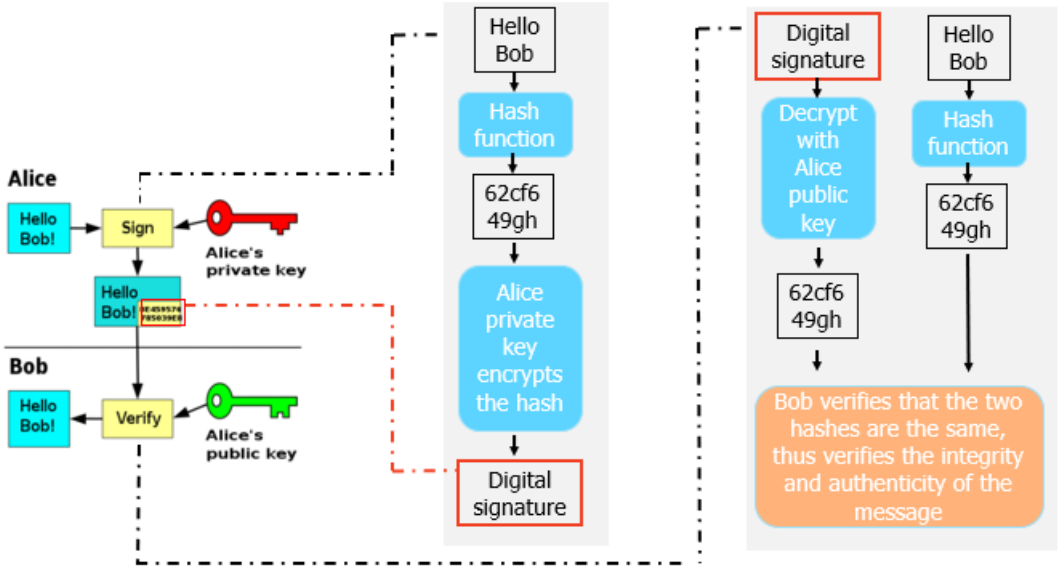


Figure 4-9: Digital signatures - signing and verification process
https://commons.wikimedia.org/wiki/File:Private_key_signing.svg

4.3.6 Digitally signed certificates

A digital certificate is an electronic document that holds information on the public key and also other PKI-related information, and is utilized to prove the ownership of the relevant public-private key pair. Digital certificates are typically utilized to:

- Protect the server-client communication by encrypting the communication channel.
- Protect e-mail communication by encrypting email messages.
- Verify the source and integrity of signed executable code.

The X.509 digital certificates is a widely accepted standard specifying the format for digital certificates, including fields such as:

- Version Number
- Serial Number
- Signature Algorithm
- Issuer
- Validity period
    - Not Before
    - Not After
- Subject
- Subject Public Key Info
    - Public Key Algorithm
    - Public Key
- Extensions (optional)
- Certificate Signature Algorithm
- **Certificate Signature**

The certificate signature is the digital signature of an entity that has verified the certificate's contents. This can be utilized to confirm the authenticity and integrity of the certificate.

Figure 4-10 indicates the digital certificate retrieved from amazon.co.uk.

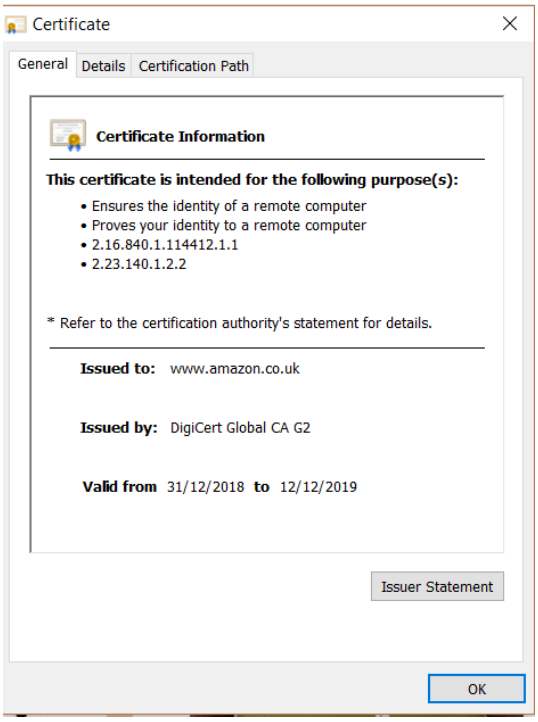


Figure 4-10: Digital certificate example

The certificate includes a field called Thumbprint (can be found under Details tab). This is the hash of the certificate, computed over all certificate data and its signature. To check the certificate's status, the certification path needs to be validated.

The certification path includes a chain of digital certificates (Figure 4-11) and each one is digitally signed by a trusted Certification Authority to indicate who the owner of the digital certificate is.

During validation, the digital signature of each certificate is validated to check whether its authentic and also to check the integrity of the data. Given that all validations are successful (starting from the end-entity certificate to the root certificate), the result is indicated at the bottom of the certificate (Figure 4-12).

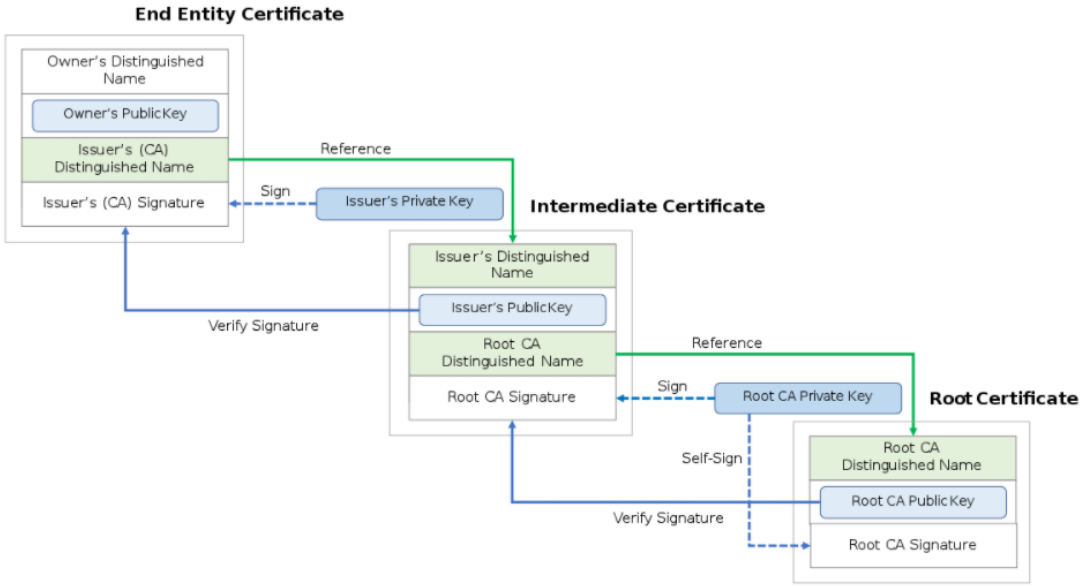


Figure 4-11: Chain of trust
https://commons.wikimedia.org/wiki/File:Chain_Of_Trust.svg

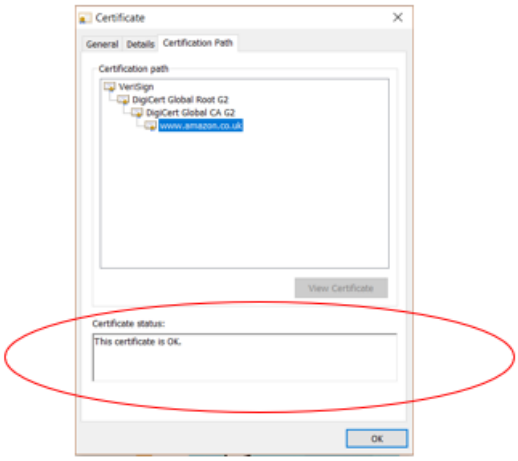


Figure 4-12: Digital certificate validity status

4.3.7 Asymmetric cryptography activity

To confirm understanding of the asymmetric cryptography concepts, consider the following actions and decide whether a private or public key is utilized and also who is the owner of the key:

1. Bob wants to communicate with Alice and protect the messages he sends her
2. Alice wants to decrypt the encrypted message sent by Bob

3. Bob wants to send a copy to himself of the encrypted message that he sent to Alice
4. Bob wants to read an encrypted message sent by Alice
5. Bob sends to Susan the message from Alice
6. Bob digitally signs a message and sends it to Alice
7. Alice confirms Bob's digital signature

Then study the following table and confirm your answers.

Table 4-1: Asymmetric cryptography activity

Action	Whose key to use	Which key to use (public, private)	Explanation
Bob wants to communicate with Alice and protect the messages he sends her	Alice's key	Public key	When an encrypted message is to be sent the recipient's key is used and not the sender's keys
Alice wants to decrypt the encrypted message sent by Bob	Alice's key	Private key	An encrypted message can only be read by using the recipient's private key
Bob wants to send a copy to himself of the encrypted message that he sent to Alice	Bob's key	Public key to encrypt Private key to decrypt	An encrypted message can only be read by the recipient's private key; Bob would need to encrypt it with his own public key and then use his private key to decrypt it
Bob wants to read an encrypted message sent by Alice	Bob's key	Private key	The recipient's private key is used to decrypt received messages
Bob sends to Susan the message from Alice	Susan's key	Public key	The message should be encrypted with Susan's key for her to decrypt and read it with her private key
Bob digitally signs a message and sends it to Alice	Bob's key	Private key	Bob's private key is used to encrypt the hash
Alice confirms Bob's digital signature	Bob's key	Public key	Because Bob's public and private keys work in both directions. Alice can use his public key to decrypt the hash

4.4 Digital Rights Management (DRM)

In this section, the focus is on Digital Right Management (DRM) technologies. Specifically, we will understand what intellectual property consists of and the need to protect it. Moreover, the main DRM technologies that can be implemented to safeguard intellectual property will be discussed.

4.4.1 Intellectual property & piracy

Intellectual property (IP) refers to the creation of the mind, such as inventions, literary and artistic work, designs, images, etc. Intellectual property rights specify how the IP can be utilized, and they include patents, copyright, industrial design rights, trademarks, etc. Resources are easily shared on the Internet where people can download material and use it in a way that compromises the intellectual property rights. Violation of intellectual property rights is called "infringement" with respect to patents, copyright, and trademarks. For example, copyright infringement involves reproducing, distributing or making derivative work without the permission from the copyright holder, also known as piracy. Content protection is a critical issue and the work created by someone needs to be respected and appropriate attribution needs to be given when using and/or referencing someone else's work.

4.4.2 What is DRM

Digital Rights Management (DRM) technologies are utilized to protect content, specifically by exchanging in a secure way intellectual property over the Internet or other electronic media. For example, exchange copyright-protected music, video, books, etc. DRM tools are utilized to control access, limit the usage, modification and distribution of copyrighted work.

4.4.3 DRM vendors

There are many DRM vendors that offer content protection solutions such as:

- Microsoft
- IBM
- Adobe
- Sony
- Intertrust
- Contentguard
- etc.

4.4.4 DRM technologies

A variety of DRM technologies exist to protect access to content. Following, the key technologies are briefly discussed.

1) Verifications

a) Product keys

A popular method to verify access to a resource is through product keys. User purchases a product key and utilizes it to activate a product. The key represents a license to use the product. This technique is often combined with other methods, e.g. an online activation, to address the issue of cracking a product key.

b) Limited install activations

Another way to control access to a resource, e.g. software, is by limiting the installations a user can activate. Usually, an online server is utilized to enforce this control.

c) Persistent online authentication

Game platforms often use online authentication to control access to a game (or parts of a game). Players have to connect to a server to get authenticated and then be able to play the game.

2) Tracking

a) Watermarks

Digital watermarks are steganographically embedded within the content during production or distribution. For example, they can record the copyright owner, who purchased a book, etc. They cannot enforce a technological restriction but can be used for copyright enforcement alongside other technologies.

b) Metadata

Metadata includes information about the actual data. Embedding metadata within a resource can track ownership of a licensee. Purchased media can include metadata, e.g. purchaser's name, email address, creation data, etc.

3) Encryption

Another mechanism to protect access to a resource includes encrypting content to protect unauthorised people from accessing it. A user needs a decryption key to be able to access the content, without it even if the user has acquired the content, he/she will not be able to read it.

4) Copying restriction

It is also essential to be able to control people from copying, printing, or sharing electronic documents. Restrictions should also apply when it comes to keeping backups of resources. A typical method to achieve this kind of restriction is by integrating with content management system software to enforce restrictions. There are four main e-book DRM schemes in common use today, from Adobe, Amazon, Apple, and Marlin Trust Management Organization (MTMO).

4.5 Cryptanalytic and social engineering attacks

In this section, a range of cryptanalytic and social engineering attacks will be discussed so that the readers can understand the methodology that is implemented in each case. Specifically, the following attacks will be presented:

- Ciphertext only attack
- Known plaintext
- Chosen plaintext
- Chosen ciphertext
- Dictionary attack
- Brute force attack
- Rainbow table
- Man-in-the-middle
- Replay attack
- Birthday attack

- Social engineering

One of the key objectives of attackers is to analyse the ciphertext and try to break the cipher, retrieve the original plaintext message and obtain the key so they can decrypt future messages. Cryptanalysis helps attackers to compromise encryption and decode an encrypted message, even if the cryptographic key is unknown. Usually, the encryption and decryption algorithms are public knowledge and anyone can study their operation. With that in mind, an attack can be executed against: a) Cryptographic algorithms to identify a weakness in their implementation, b) Encrypted messages to gain us much information as possible about the unencrypted data. To do so, a range of cryptanalytic attacks can be implemented. The success of a cryptanalytic attack is often depended on the amount of information that is available to the attacker.

4.5.1 Ciphertext only attack

It is the most common type of cryptanalytic attack. The attacker is assumed to have obtained the ciphertext of several messages, for example, by eavesdropping on the communication. All messages are encrypted with the same encrypted algorithm and the plaintexts of the encrypted messages are not known. With a ciphertext only attack, the attacker analyses the encrypted messages and tries to identify repeating patterns, e.g. of words that occur frequently such as “the”, “or”, etc., to be able to reconstruct the original message. Typically, this attack requires a large sample of encrypted messages for the analysis to take place. Modern ciphers are required to be very resistant to this type of attack.

4.5.2 Known plaintext

In this type of attack, the attacker has a number of pairs of plaintext and corresponding ciphertext and he analyses them with the goal to find the cryptographic key that was utilized to encrypt the messages.

4.5.3 Chosen plaintext

The chosen plaintext attack considers an attacker that encrypts selected plaintext messages and then analyses the encrypted messages. The objective is to obtain a better understanding of the encryption process and gather more information to identify the cryptographic key utilized.

4.5.4 Chosen ciphertext

In this case, the attacker can choose the ciphertext to be decrypted and obtain the corresponding plaintext to do the analysis and investigate whether he can infer any useful information. This attack requires the attacker to have access to the communication channel or the recipient and to be able to capture the ciphertext messages.

4.5.5 Dictionary attack

Dictionary attack is considered the most commonly attack against password files. The idea behind this attack is that users have the bad habit to choose simple passwords that can be easily compromised. Considering the situation where the attacker has compromised a system and has obtained the hashed passwords, he can attempt to crack them through a dictionary attack. To do so, the attacker constructs a dictionary, or reuses compromised dictionaries publicly available on the Internet, hashes each word listed in the dictionary and then checks whether the resulting hash matches one of the hashed passwords already obtained. If there is a match, then the password is discovered.

4.5.6 Brute-force attack

Brute-force attack is also called an exhaustive key search. The attacker tries every possible key or password until the correct one is found. This means that if the key has N bits, there are 2^N possible keys to try. Due to the exhaustive search that is performed, this is a time and resource intensive attack, depending on the length of the key. Of course, these days the

increased amount of processing power can increase the chances of an attacker to succeed in his attempt to crack a password.

4.5.7 Rainbow tables

The rainbow tables are utilized to enhance the effectiveness of brute-force attacks. They include precomputed values for cryptographic hashes which can be utilized to crack hashed passwords. Given that hashing a plaintext password is omitted and the precomputed value can be utilized over and over again in different password cracking attempts, the password cracking time is optimized.

4.5.8 Man-in-the-middle attack

Figure 4-13 presents the man-in-the-middle attack:

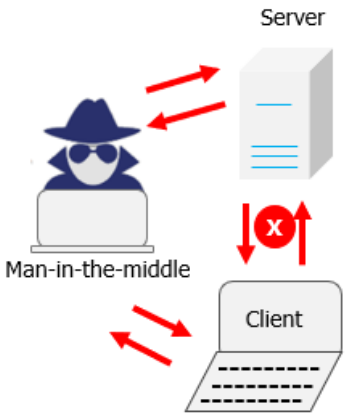


Figure 4-13: Man-in-the-middle attack

As the figure indicates:

1) An attacker is placed between two communicating parties and intercepts all communication.

2) The attacker intercepts the setup of the cryptographic session and establishes a session with each communicating party instead of letting the two parties to communicate directly.

3) Initially, the attacker responds to the initiator of the communication and establishes a secure session with him.

4) Then the attacker establishes a secure session with the intended recipient, masquerading as the initiator. This results to the attacker sitting in the middle, thus the term “man-in-the-middle”.

5) The attacker is then able to read the traffic exchanged between the two parties.

4.5.9 Replay attack

In a replay attack the attacker intercepts a data transmission, for example it captures a session key, and retransmits the data to establish an authenticated session. It is also known as a playback attack. *Figure 4-14* presents an example of how the attack works:

1. Alice logins into the server
2. Eve is eavesdropping and captures the login information
3. At a later stage, Eve sends the captured data to the server to login as Alice

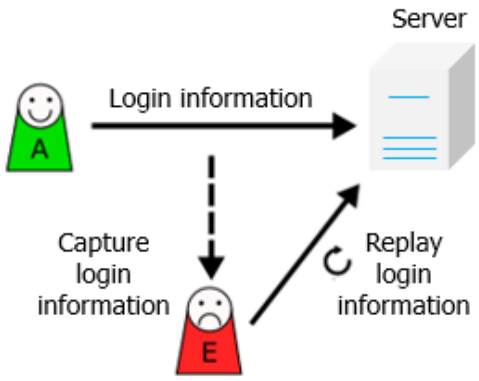


Figure 4-14: Replay attack

https://commons.wikimedia.org/wiki/File:Replay_attack_on_hash.svg

4.5.10 Birthday attack

The birthday attack is an attack against hashing functions. The principle here is based on the birthday paradox that specifies that in a group of 23 people, there is at least a 50% probability that at least two people will share the same birthday.

Consider that a hash function results into a specific digest (or hash) for a specific message. In a birthday attack, the attacker tries to find two messages with the same hash value. The attacker's objective with this attack is to substitute a message with another that will lead to the same hash value, therefore, maintain the validity of the hash even if the message is not the original. The birthday attack is also known as the collision attack or reverse hash matching.

4.5.11 Social engineering

Social engineering is considered the most common attack and typically the most successful as it targets to exploit human trust. The attacker tries to manipulate people through various means to reveal sensitive information. Social engineers' mentality is "Why try to crack a secret when you can just ask for it? People are trusting and helpful, these are characteristics that attackers are trying to exploit. Often ignorance is a driving factor for the attack to succeed, thus the need for a continuous security education of people.

Typically, social engineers utilize three means to deliver the attack: over the phone, using digital means or in-person. Given that there is a range of attacks that can be executed under social engineering, the attacker has increased chances to succeed. Some common social engineering attacks are the following:

- Phishing – Usually phishing is delivered through email. The attacker distinguishes as a well-known service/organization and tries to trick the user to provide sensitive information, e.g., credentials, open an infected attachment and/or download a malware. Usually, the attacker utilizes different baits to make his victims to take an action and get infected. For example, he advertises miles bonus, expired passwords, holiday discounts, etc.
- Vishing – Individuals are targeted over the phone. For example, the attacker calls an employee in an organizing pretending that he is from the IT department. He informs

the employee of an urgency, e.g., an update that needs to be done and the employee must provide his credentials for the update task to be completed. If the employee provides his credentials, the attacker can later on use them to gain access to the employee's account.

- Smishing – In this case, the attack is delivered over SMS. The attacker sends a message to his victim including a malicious URL. When the victim visits the URL the site asks him to create an account. If the victim creates an account providing personal information, the attacker steals the information and uses it for his own malicious purposes.
- Shoulder surfing – The attacker visits the victim organization. He tries to go near-by personnel that are working on their computer and tries to read when the user types in information using the keyboard. For example, it targets to read user's credentials when he tries to login in the system.
- Water-holing – Attacker compromises a website and uploads an infected file with malware. Afterwards, he sends an email to the victim including a link to the compromised website. When the victim visits the website and downloads the infected file, a malware is installed on his system that grants access to the attacker.
- Dumpster diving – Often people throw away things that include valuable information, e.g., invoices, post-it with listed credentials, corporate phone books, computer manuals, etc. Attacker collects sensitive information that is found in the trash and then he uses the collected information to strategize his attack.

4.5.12 Countermeasures

To protect from cryptanalytic and social engineering attacks, different control measures can be taken such as the following:

- Encryption. If the communication channel is encrypted, then eavesdropping will be prohibited.
- Session ID. If a session ID is utilized, then the replay attack can be detected.

- User education. It is essential to educate end users about best practices in cybersecurity so they can create stronger passwords and be able to identify phishing attacks.
- Multi-factor authentication. This control requires two or more authentication channels, for example to get authenticated the system requires a user-specified password and a one-time password sent to the user's mobile phone. By using multi-factor authentication, compromising an account is less likely to happen even if the attacker retrieves a user's password. If he does not have access to the one-time password, then he will not be able to login to the account.
- Use salt in passwords. This approach eliminates the case where the exact same passwords (without using a salt) are hashed and saved on a system. By using a salt when hashing a password, password guessing gets difficult.
- Use digital certificates. Digital certificates promote confidentiality, integrity and non-repudiation and they are a key countermeasure that needs to be applied in digital transactions and communications to protect from cyber threats.

4.6 Summary

Summarizing, below are the main points to take away:

- Hashing is useful to verify the integrity of a message
- Digital signatures are utilized to verify the integrity and origin of a message
- Given that a private key is kept secret, a user digitally signing a message cannot later deny his action (non-repudiation property)
- DRM technologies are useful to control access to intellectual property
- A range of cryptographic attacks exist; therefore, it is essential to deploy best practices to decrease the possibility for an attacker to successfully launching an attack

4.7 Referencing

[4] 2015. Official (ISC)2 Guide to the CISSP CBK (4th. ed.). Auerbach Publications, USA. [Reference Domain 3: Security Architecture and Engineering]

[15] 2016. Cryptography and Network Security: Principles and Practice (7th Edition). William Stallings. Pearson, USA. [Reference chapters 11 & 13]

[16] Cryptographic hash function: https://en.wikipedia.org/wiki/Cryptographic_hash_function

[17] Secure hash algorithms: https://en.wikipedia.org/wiki/Secure_Hash_Algorithms

[18] Digital signatura: https://en.wikipedia.org/wiki/Digital_signature

[19] Chain of trust: https://en.wikipedia.org/wiki/Chain_of_trust

[20] Digital rights management: https://en.wikipedia.org/wiki/Digital_rights_management

[21] Cryptanalysis: https://en.wikipedia.org/wiki/Cryptanalysis

[22] Attack model: https://en.wikipedia.org/wiki/Attack_model

[23] Replay attack: https://en.wikipedia.org/wiki/Replay_attack

[24] MD5: https://en.wikipedia.org/wiki/MD5

[25] SHA-1: https://en.wikipedia.org/wiki/SHA-1

[26] SHA-2: https://en.wikipedia.org/wiki/SHA-2

[27] SHA-3: https://en.wikipedia.org/wiki/SHA-3

5 Communications & Network Security: Introduction

Author(s): Filippos Pelekoudas Oikonomou
Georgios Mantas
Claudia Barbosa
Jonathan Rodriguez

5.1 Secure Design Principles

Although research and development has continued for many years, the development of security design and implementation techniques which can consistently reduce security flaws and inhibit all unauthorized actions. The lack of these fool-proof techniques means that there should be a set of agreed design fundamentals that will assist in guiding how the protection mechanisms are developed. The National Centres of Academic Excellence in Information Assurance/Cyber Defence, that is funded by the U.S. Department of Homeland Security together with the U.S. National Security Agency, note the security design principles mentioned below:

1. Open Design
2. Least Common Mechanism
3. Fail-Safe Defaults
4. Complete Mediation
5. Economy of Mechanism
6. Separation of Privilege
7. Least privilege
8. Least Astonishment

These principles are expressing classic practices of simplicity and restriction in the case of IT systems.

5.1.1 Principle of Least Privilege

According to *Computer security: Art and Science by Matt Bishop, "The principle of least privilege states that a subject should be given only those privileges that it needs in order to complete the task."*

This privilege specifies how privileges are granted and to which subject. According to this, an individual should be provided only those privileges that are necessary to complete a task. The function of the subject has to have the control of the privileges and rights that are assigned to it. In case of more privileges are needed for a further action, these privileges must be

withdrawn after the action is completed. This principle advises that every process has to be constrained to the minimum domain protection is possible.

5.1.1.1 Principle of Least Authority (POLA)

This principle is closely related to the Principle of Least Privilege and often is considered the same. In order to understand the difference, we have to distinct the terms of permission and authority:

- Permissions control what the subject can do to an object directly.
- Authority controls what influence a subject has over and object (directly, indirectly or through other subjects).

5.1.2 Principle of Fail-Safe Defaults

This principle constrains the starting principles of a subject or an object when this is created.

Matt Bishop *stated that "The principle of fail-safe defaults states that, unless a subject is given explicit access to an object it should be denied access to that object."*

In other words, the default action is to grant no access, privileges or other attributes related to security. If the subject is not able to complete an action, the system will be safe, due to the fact that, this will undo all the security-related alternations before the system terminates.

5.1.3 Principle of Economy Mechanism

Matt Bishop defined this principle as *"The principle of economy of mechanism stats that security mechanisms should be as simple as possible."*

In this case the interest relies on making a system as simple as possible. This way, there are fewer possibilities that something will go wrong and in the case of an error, it will be easier to understand and fix it.

Some modules and interfaces perform actions with wrong parameters and the result is often an error. With limited permissions as a default, these problems can be avoided.

5.1.4 Principle of Complete Mediation

The purpose of this principle is to have a complete and ongoing check in every access and privileges is given.

According to Computer security: Art and Science by Matt Bishop, *"The principle of complete mediation requires that all accesses to objects be checked to ensure that they are allowed."*

Every time a subject performs an action, the system should observe the action for verification that is a legal one. In case of repeated and ongoing actions, the system needs to perform this control every time. Some systems do not control more than the first time and base the rest actions to the first check. An example of this case is the UNIX operating system in which, access is checked upon the first time but not checked thereafter. As a result, if a permission is changed after, the subject may get an unauthorized access.

5.1.5 Principle of Open Design

This principle specifies that security should not rely on the secrecy of implementation or design details.

Popularly it is misunderstood that this principle means that source code should be public but this is not the case. What it really means is that the strength of the security of a program should not be based on the ignorance of the user, because if that is true then a user that has sufficient information about the program will be able to overpass the security mechanism. This concept can be defined by the term "security through obscurity". Again, that does not mean that passwords, cryptographic keys should be a public information.

Based on the open design principle, the secrets regarding a security mechanism should be kept to the minimum by the creator. That is because the leak of secrets is inevitable most of the times. Therefore, it is better to keep what it matters, confidential, rather than keeping everything in secrecy.

5.1.6 Principle of Separation of Privilege

Computer security: Art and Science by Matt Bishop defines, *"The principle of separation of Privilege states that a system should not grant permission based on a single condition."*

By separating of duties and giving access and rights only to the subject responsible for a certain task and rights sufficient only to complete the task then a good control over the resources and the access that each subject has on them.

5.1.7 Principle of Least Common Mechanism

The least common mechanism principle is a restrictive one. The idea behind it is that the mechanisms, that are used to reach assets, should not be shared.

Distributing resources grants a channel along which information can be broadcasted, and for this reason this distribution ought to be minimized to the necessary. By reducing the number of common mechanisms also minimizes the possibility of an attack that jeopardizes such a mechanism. If all version of a system uses the same mechanism, then compromising the sole mechanism gives the opportunity to adversaries to compromise any other mechanisms of the same kind. By changing the version of each system, slightly, then compromising becomes a more complicated process.

With the use of virtual machines and sandboxes the operating system will enforce this privilege and provide isolation between resources and information.

5.1.8 Principle of Least Astonishment

This principle states that security mechanisms should be designed so users understand why the mechanism works the way it does, and the use of the mechanism to be simple. The logic behind this, is that if the mechanism's mental model is different than the one of the target user group, then it can weaken the security mechanism. The mechanisms must be easy to install, configure and use.

Human factor plays a major role in this principle. For this reason, the design must keep a balance between the comfort of the user and the security requirements. With all the previous,

another thing that has to be taken into account during the designing process is the environment that the security mechanism is used.

5.1.8.1 Psychological Acceptability

According to this principle the security mechanisms should not add difficulty of accessing a resource, like if the security mechanism did not exist. The difference between Psychological Acceptability and the Principle of Least Astonishment is that the former states an ideal situation while Psychological Acceptability recognises the fact that more difficulty is added on a system because of a security mechanism, to access resources.

5.2 Cryptography used to maintain communication security

In this chapter there are going to be analysed two security mechanisms that are broadly used in Internet applications to maintain security. These mechanisms are the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) which are applied in the Transport layer and IP security (IPsec) which is applied on the Network Layer (IP protocol).

5.2.1 Secure Sockets Layer (SSL)

The Secure Sockets Layer (SSL) and its associated Internet standard called Transport Layer Security (TLS) defined in RFC 5246, is one of the widely utilized security services nowadays. SSL consists a general-purpose service and can be implemented as a set of protocols that employ the TCP protocol. There are two implementation alternatives. The first choice aims at full generality and refers to SSL (or TLS) to be included as part of the underlying protocol suite and thus, it will be transparent to applications. The second option is to embed SSL into specific packages.

5.2.1.1 SSL protocol stack

SSL utilizes TCP to support a reliable end-to-end secure service providing basic security services to various higher layer protocols. The Hypertext Transfer Protocol (HTTP), offers the transfer service for the interaction between the Web client and Web server and it can be deployed on top of SSL. Three higher-layer protocols are defined as part of SSL consists of three higher-layer protocols. Those are the Handshake Protocol, the Alert Protocol and the Change Cipher Spec Protocol. The SSL exchanges are managed using these SSL-specific protocols.

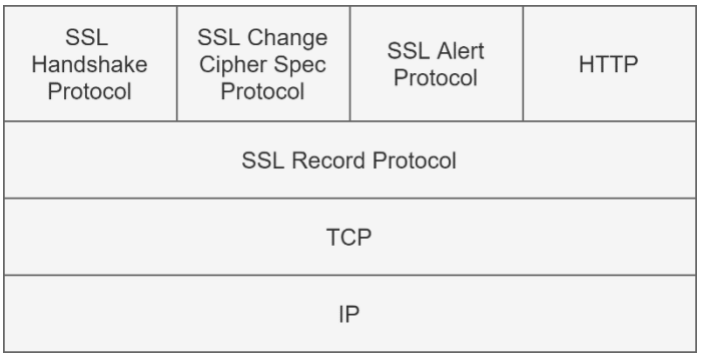


Figure 5-1: SSL Protocol Stack [1]

5.2.1.2 SSL architecture

The SSL session together with the SSL connection constitute two significant SSL concepts and they are described inside the specification as mentioned below.

- **Connection**: constitutes a transport (based on the OSI layering model description) which offers an appropriate type of service. In the case of SSL, these connections appear as peer-to-peer relationships and are transient. Each connection corresponds to one session.
- **Session**: A connection between a server and a client is described as a SSL session. The Handshake Protocol initializes the Sessions. Sessions produce a collection of cryptographic security parameters and multiple connections can share these parameters. Using Sessions, the costly negotiation of new security parameters required for every connection can be avoided.

Two entities (applications such as HTTP on client and server) may share more than one secure connection, as well as a great number of states relating to each session. After the establishment of a session, both read and write (i.e., receive and send) have a currently operating state. Additionally, while the Handshake Protocol is in effect, pending write and read states are generated. After the Handshake Protocol concludes successfully, the current states are the previously pending states.

5.2.1.3 Session State

The parameters mentioned below describe the state of a session.

- The **Session identifier:** an arbitrary byte sequence which the server selects.
- The **Peer certificate:** an X509.v3 certificate of the peer.
- The **Compression method:** algorithm that compresses the data before the encryption.
- The **Cipher spec:** the bulk data encryption algorithm (e.g., null, AES, etc.) and a hash algorithm (such as MD5 or SHA-1)
- The **Master secret**: a 48-byte secret key which the client and the server share between them.
- The **Is resumable** flag shows if new connections can be established by the session.

5.2.1.4 Connection State

The parameters, that are presented below, show the state of a connection:

- **Client and Server random**
- **Client write MAC secret**
- **Server write MAC secret**
- **Client write key**
- **Server write key**
- **Initialization vectors**
- **Sequence numbers**

5.2.1.5 SSL Record Protocol

Two services regarding SSL connections are defined in the SSL Record Protocol:

- **Confidentiality**: The Handshake Protocol sets a shared secret key. This shared secret key is employed during the encryption of the SSL payloads. The message is initially compressed and after that, it is linked together with the MAC and encrypted. Various ciphers are supported as shown.
- **Message Integrity**: Another shared secret key is also defined in the Handshake Protocol. This shared secret key is utilized to produce a message authentication code (MAC), that is comparable to HMAC.

Figure 5-2 shows how the SSL Record Protocol generally functions. The Record Protocol receives an application message that requires transmission. The data of the message are fragmented into manageable blocks and they can optionally be compressed. After that, the Record Protocol insert a MAC and encryption is performed. Then, a header is added and the resulting message is sent, using a TCP segment. At the destination, decryption, verification, decompression and reassembly is performed on the received data and then, the data are provided to higher-level users.

The algorithm initially performs fragmentation. Every upper-layer message is divided into blocks whose length is 2^{14} bytes or less than that. After that, optionally, the data can be compressed and this process must be lossless and the content may not be increased by more than 2^{10} (1024) bytes. In the case of SSLv3 (as well as the current version of TLS), there is not a specified compression algorithm and thus, the default algorithm, used for compression, is null.

The next process involves the computation of a message authentication code based on the compressed data. In this case, the algorithm uses a shared secret key.

Next, symmetric encryption is employed to encrypt the compressed message and the MAC. The encryption process must not increase the length of the content by more than 1024 bytes, in order that the total size does not surpass 2^{14} + 2048 bytes.

In the case of stream encryption, the message that was compressed is encrypted with the MAC. It is important to notice that the MAC is calculated before the encryption of the plaintext or compressed plaintext plus the MAC occurs.

In the case of block encryption, padding may be included after the MAC before encryption occurs. The padding consists of a specific amount of padding bytes and one byte that describes the size of padding. The total size of padding is the minimum possible size in order that the total length of data for encryption (plaintext and MAC and padding) is a multiple of the length of the cipher's block. One instance is a plaintext (or compressed text) of 58 bytes and a MAC of 20 bytes (using SHA-1), that are supposed to be encrypted utilizing a block size of 8 bytes (e.g., DES). Considering the padding-length byte, this sums up to 79 bytes. Since the total length needs to be an integer multiple of 8, it is necessary to add one byte of padding.

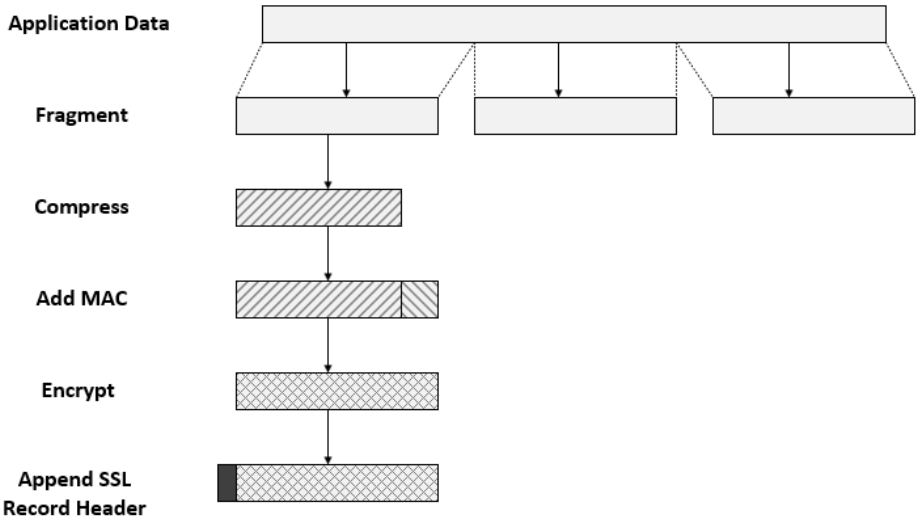


Figure 5-2: SSL Record Protocol Operations [1]

Finally, the SSL Record Protocol needs to construct a header that includes the fields mentioned below:

- The **Content Type (8 bits)**
- The **Major Version (8 bits)**
- The **Minor Version (8 bits)**
- The **Compressed Length (16 bits)**

The defined content types are change_cipher_spec, alert, handshake and application_data. The first three are SSL-specific protocols and they are discussed next. Note that SSL does not distinguish between the content of data which is created by the different applications, such as HTTP.

Figure 5-3 illustrates the SSL record format.

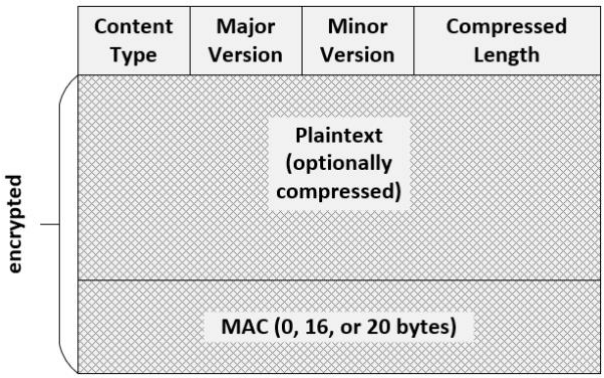


Figure 5-3: SSL Record Format [1]

The Change Cipher Spec Protocol is the simplest of the three SSL-specific protocols which utilize the SSL Record Protocol. This protocol uses a single message (figure 5-4a) with a single byte whose value is 1. This message is used in order that the pending state gets copied into the current state. This, in turn, causes the update of the cipher suite which is utilized on this connection.

The purpose of the Alert Protocol is the transmission of SSL-related alerts to the peer entity. Compression and encryption are performed on the alert messages before they are sent, as the current state specifies.

In the Alert Protocol, every message uses two bytes (figure 5-4b). The value of the first byte can indicate either warning (1) or fatal (2) and this refers to the severity of the message. In case that the level is fatal (2), SSL urgently ends the connection. Other connections that are part of the same session may continue, but new connections cannot be created on this session. The second byte includes a code which shows the particular alert.

The most complicated part of SSL is the Handshake Protocol which enables mutual authentication between the server and client, and negotiation about the encryption and MAC

algorithm and cryptographic keys to be used to protect information in an SSL record. The Handshake Protocol consists of a series of messages exchanged by client and server. All of these have the format shown in figure 5-4c.

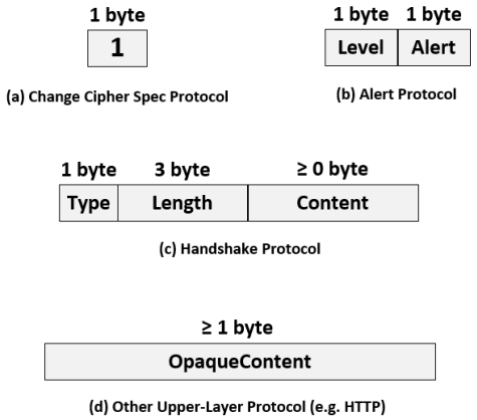


Figure 5-4: SSL Record Protocol Payload [1]

Figure 5-5 depicts the initial exchange that is required for the establishment of a logical association between server and client. The exchange constists of four stages.

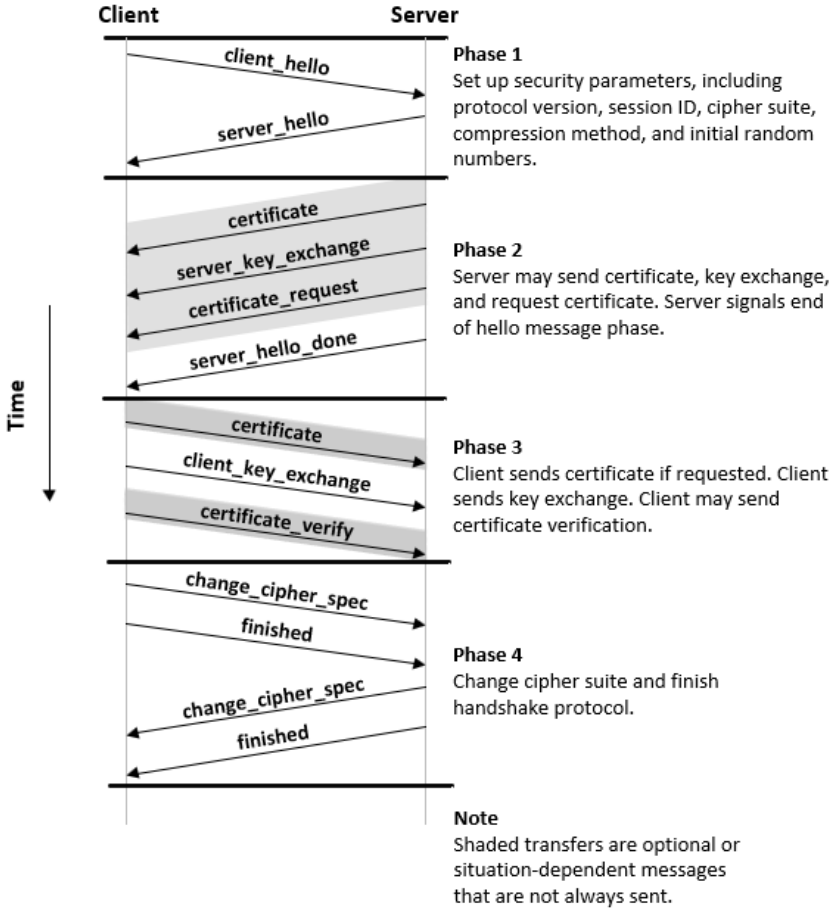


Figure 5-5: Handshake Protocol Action [1]

5.2.1.6 Cryptographic Computations

The interest revolves around two additional items: (1) the generation of a shared master secret key using key exchange means and (2) the formulation of cryptographic parameters utilizing the master secret.

The shared master secret key constitutes a one-time 48-byte value (384 bits) that is created for this session using secure key exchange means. There are two stages for the creation of the key. Initially, a pre_master_secret key is exchanged. After that, both parties compute the master_secret key. The pre_master_secret exchange can be performed in two possible ways:

- **RSA**: The client creates a 48-byte pre_maste_secret key, that is encrypted with the server's public RSA key, and then sends the key to the server. Then, the ciphertext is decrypted by the server utilizing its own private key to recover the pre_master_secret key.
- **Diffie-Hellman**: A Diffie-Hellman public key is created by both the client and server.

5.2.2 Transport Layer Security (TLS)

TLS is an IETF standardization initiative whose objective lies on the establishment of an Internet standard version of SSL. TLS exists as a Proposed Internet Standard in RFC 5246, which is comparable to SSLv3.

- **Version Number**

The TLS Record Format and the SSL Record Format are the same and the respective fields of the header have equal meanings. Only the version values are different. The current TLS version uses the number 3 as both the major and the minor version number.

- **Message Authentication Code**

The SSLv3 and the TLS MAC schemes differ in two points: their algorithms and their scopes of the MAC calculation. The HMAC algorithm, which is described in RFC 2104, is employed in TLS.

The MAC calculation relates to all of the fields of the SSLv3 calculation, in addition to the field TLSCompressed.version. This field is the version of the employed protocol.

- **Pseudorandom Function**

In TLS, a pseudorandom function named as PRF is used to extend secrets into blocks of data for the cases of key validation or generation. The goal is the use of a comparatively small shared secret value and also, the computation of longer blocks of data which will be protected from attacks targeting MACs and hash functions.

TLS covers all of the alert codes described in SSLv3 except from the alert code of no_certificate. TLS defines an amount of additional codes.

- **Cipher Suites**

The cipher suites used in SSLv3 and in TLS share several small differences between them:

  - **Key Exchange**: All key exchange techniques of SSLv3, except from Fortezza, are included in TLS.
  - **Symmetric Encryption Algorithms**: All symmetric encryption algorithms of SSLv3, except from Fortezza, are supported in TLS.

- **Client Certificate Types**:

TLS describes four certificate types that are requested in the case of a certificate_request message: rsa_sign, dss_sign, rsa_fixed_dh, and dss_fixed_dh.

- **certificate_verify and Finished Messages**

In the TLS certificate_verify message, the MD5 and SHA-1 hashes are computed only over handshake_messages. In SSLv3, the computation of the hash also involves the master secret and the pads. No new security is added by these extra fields.

- **Cryptographic Computations**

In TLS, the pre_master_secret is computed in the same way as in SSLv3. As in SSLv3, TLS calculates the master_secret using a hash function of the pre_master_secret and the two hello random numbers.

- **Padding**

In SSL, the padding that is inserted before the encryption of the user data is the smallest amount needed. This amount should simultaneously be of a magnitude that the total size of the encrypted data is a multiplier of the cipher's block size. In TLS, the amount of padding can

be any number as long as it yields a total that is a multiple of the cipher's block size, with an upper limit of 255 bytes.

5.2.3 HTTPS

HTTPS (HTTP over SSL) concerns the fusion of both HTTP and SSL to achieve secure communication among a Web browser and a Web server. HTTPS (HTTP over SSL) related to the usage of both HTTP and SSL to establish secure communication between a Web server and a Web browser.

All modern Web browsers are equipped with the HTTPS feature. Its use relies on the Web server implementing HTTPS communication.

The main change, that a user sees in a Web browser, is that URL (uniform resource locator) addresses start with https:// instead of http://. A normal HTTP connection uses port 80. If HTTPS is specified, it uses port 443, invoking SSL.

5.2.3.1 Connection Initiation

In the case of HTTPS, the entity posing as the HTTP client is also the TLS client. The client sets up a connection to the server on a suitable port and after that, the TLS ClientHello is sent by the client in order to initiate the TLS handshake. After the TLS handshake is completed, the client may transmit the first HTTP request. The entirety of the HTTP data has to be transmitted as TLS application data. It is necessary to follow typical HTTP behavior, including retained connections.

A connection in HTTPS contains three layers of awareness. At the HTTP layer, the HTTP client requests a connection to the HTTP server. This happens by transmitting a connection request to the next lowest level. Normally, the next lowest level is the TCP level, but it could also be the TLS/SSL level. In the case of TLS, a session is created among the TLS client and the TLS server. During this session, one or more connections can be maintained at any time. As it was mentioned, a TLS request for the establishment of a connection initiates with the creation of a TCP connection among the TCP client entity and the TCP server entity.

5.2.3.2 Connection Closure

An HTTP server or client can signal the end of a connection by inserting a specific line in an HTTP record: *Connection: close*. This signifies the closing of the connection after the delivery of this record.

The closure of an HTTPS connection means that TLS has to end the connection with the peer remote TLS entity. This includes ending the underlying TCP connection. The proper way to close a TLS connection involves for each side the use of the TLS alert protocol in order to transmit a close_notify alert. In TLS implementations, an exchange of closure alerts must be executed before ending a connection.

It is possible that after sending a closure alert, a TLS implementation may end the connection before the peer manages to transmit its closure alert and thus, an “incomplete close” occurs. Note that an implementation, doing this, could opt to reuse the session.

This should only happen if the application can perceive (typically through the detection of HTTP message boundaries) that all the necessary message data has been received.

HTTP clients should also be capable of dealing with a situation where the underlying TCP connection is closed while there was neither a prior close_notify alert nor a *Connection: close* indicator. The reason for such an occurence may be a programming error on the server side or a communication error that forces the end of the TCP connection. Nevertheless, an unannounced TCP closure could also be a sign that some type of attack may be happening. So, in case that this happens, it is advised that the HTTPS client produces some type of security warning.

5.2.4 Secure Shell (SSH)

Secure Shell (SSH) refers to a protocol about secure network communications. This protocol was developed with the aim of being comparatively simple and inexpensive to implement. The first version, SSH1 was centered on offering a secure remote logon facility to take the place of TELNET and other remote logon schemes which included no security. SSH also contains a more generic client/server capability and can be utilized for network functions, such as e-mail and file transfer. A new version, SSH2, deals with numerous security flaws that

existed in the first scheme. SSH2 is a proposed standard that is documented in IETF RFCs 4250 through 4256.

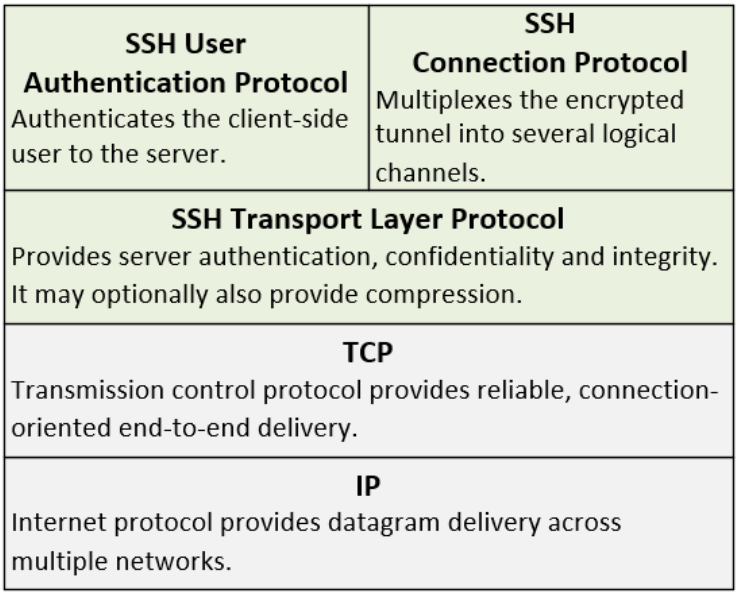


Figure 5-6: SSH Protocol Stack [1]

For most operating systems, SSH server and client applications can be easily obtained. It is the method of choice for both remote login and X tunneling and its use is continuously increasing as one of the most pervasive applications for encryption technology, with the exception of embedded systems.

SSH is typically implemented on top of TCP and it consists of three protocols (Figure 5-6):

- **Transport Layer Protocol**
- **User Authentication Protocol**
- **Connection Protocol**

5.2.4.1 Transport Layer Protocol

The transport layer is responsible for server authentication, by utilizing the public/private key pair of the server. A server may possess a number of host keys for the use of a multitude of different asymmetric encryption algorithms. The same host key may be shared by more than one host. Nevertheless, during key exchange, the host is authenticated using the server host

key. This can only occur if the client already knows the server's public host key. RFC 4251 describes two different trust models that can be employed:

1. The client retains a local database that stores the corresponding public host key for each host name (as typed by the user). This method does not need any centrally administered infrastructure or third-party coordination. The disadvantage of this method relates to the case where it becomes burdensome to maintain the database of name-to-key associations.

2. The host name-to-key association is certified by a trusted certification authority (CA). The client only possesses the knowledge of the CA root key and can validate all host keys certified by accepted CAs. This method mitigates the maintenance problem, since ideally, the client is required to store securely only a single CA key. On the other hand, a central authority must certify each host key, before authorization is possible.

5.2.4.2 SSH Transport Layer Protocol Packet

The format of each packet is presented below (Figure 5-7):

- **Packet length**: This consists of the size of packet in bytes, without the fields of packet length and MAC.
- **Padding length**: This constitutes the size of the random padding field.
- **Payload**: Valuable content of the packet. Prior to algorithm negotiation, this is an uncompressed field. In case that there is a negotiation about compression, then this becomes a compressed field in subsequent packets.
- **Random padding**: After the end of the negotiation for the encryption algorithm, this field is included. It consists of random bytes of padding in a way that the total size of the packet (not including the MAC field) is a multiple of the length of the cipher block, or 8 bytes if a stream cipher is used.
- **Message authentication code (MAC)**: In case that there was a negotiation about message authentication, the MAC value is included in this field. The MAC value is calculated over the entire packet together with a sequence number, excluding the MAC field. The sequence number consists of an implicit 32-bit packet sequence which

is set to zero for the first packet and increases for every packet. The packet transmitted over the TCP connection does not contain the sequence number.

Once the negotiation for the encryption algorithm has ended, and after the MAC value is calculated, encryption is performed to the whole packet (not including the MAC field).

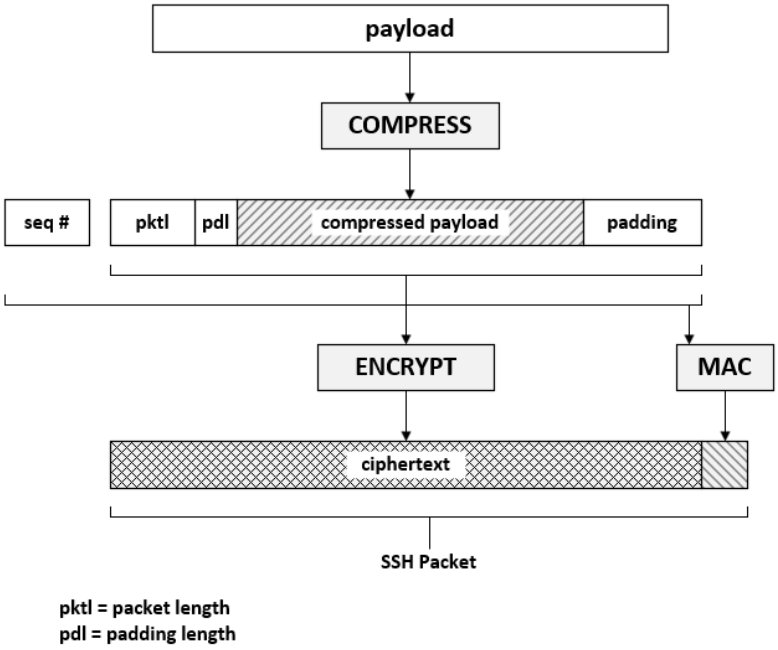


Figure 5-7: SSH Transport Layer Protocol Packet Formation [1]

5.2.4.3 Connection Protocol

The SSH Connection Protocol is implemented over the SSH Transport Layer Protocol with the assumption that a secure authentication connection is currently used. The Connection Protocol uses that secure authentication connection, otherwise named as tunnel, in order to multiplex numerous logical channels. Utilizing separate channels, all types of communication employing SSH are supported. Any side may create a channel. Each side attributes a unique channel number for each channel. Both sides do not need to use the same number for a specific channel. The data flow in channels is controlled by utilizing a window mechanism. Before sending data to a channel, it is required to receive a message which states that there is available window space.

There are three stages regarding the life of a channel. These stages are the establishment of the channel, the data transfer, and the termination of the channel.

5.2.4.4 Connection Protocol

Port forwarding consists one of the most helpful features of SSH. In essence, utilizing port forwarding, it is feasible to transform any insecure TCP connection into a secure SSH connection. This is also named as SSH tunneling. In this context, it is needed to be known what a port is. A user of TCP is identified based on the port. So, each application, that uses TCP, is characterized by a port number. The port number is used to send the incoming TCP traffic to the associated application. It is possible for an application to utilize more than one port numbers.

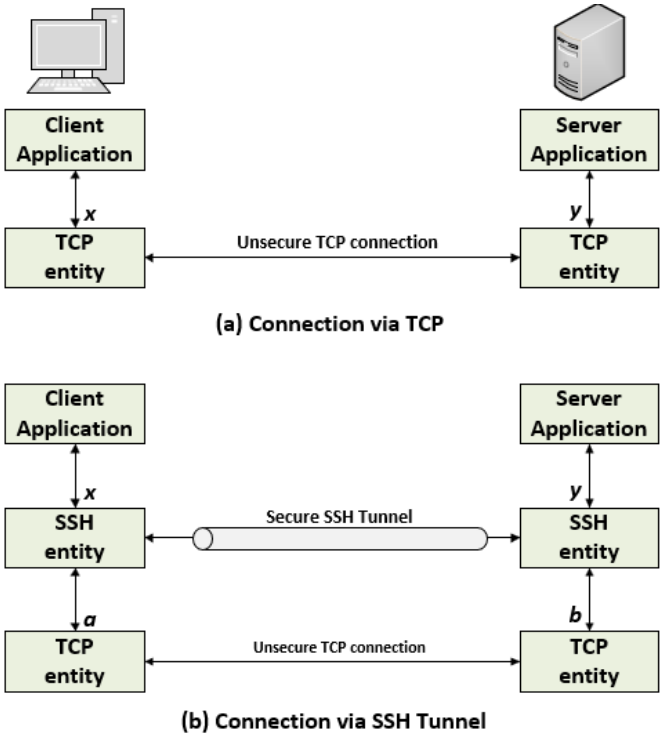


Figure 5-8: SSH Transport Layer Packet Exchanges [1]

Figure 5-8 shows the basic idea about port forwarding. A client application employs the port number x and a server application uses the port number y. The client application signals the local TCP entity to establish a connection with the remote server on port y. Then, the local

TCP entity initiates the negotiation about the TCP connection with the remote TCP entity. The goal is the creation of a connection between local port x and remote port y. SSH can also be used to secure this connection. The SSH Transport Layer Protocol initiates a TCP connection that links the SSH client entity with TCP port number a and the SSH server entity with TCP port number b. A secure SSH tunnel is set up over this TCP connection. The traffic data originated from the client at port x is sent to the local SSH entity and through the tunnel, it arrives at the point where the remote SSH entity hands over the data to the server application on port y. A similar process occurs regarding the traffic in the other direction.

In SSH, two types of port forwarding are supported: local forwarding and remote forwarding. Local forwarding permits the client to start a "hijacker" process. In this case, specific application-level traffic can be intercepted and redirected from an unsecured TCP connection to a secure SSH tunnel. SSH is set up to pay attention to selected ports. SSH picks all outgoing traffic that concerns a particular port and transmits it through an SSH tunnel. On the other side, the SSH server directs the incoming traffic to the destination port that the client application dictates.

In the case of remote forwarding, the user's SSH client takes action on behalf of the server. On the client side, traffic regarding a given destination port number is received, placed on the correct port and transmitted to the destination that the user selects.

5.2.5 IP Security

In 1994, the Internet Architecture Board (IAB) released a report named as "Security in the Internet Architecture" (RFC 1636). The report labeled key areas concerning security mechanisms. Among those mentioned, there was the necessity to guard the network infrastructure against unauthorized monitoring and control of the network traffic as well as the need for secure end-user- to-end-user traffic by employing encryption and authentication mechanisms.

The IAB incorporated authentication and encryption as essential security capabilities in the next-generation IP, named as IPv6, in order to enhance security. Fortunately, these security mechanisms were developed in a way that they can be employed by both the current IPv4

and the future IPv6. This leads to vendors being able to provide these mechanisms now. A number of vendors now include some IPsec capabilities in their products. The IPsec specification constitutes a set of Internet standards.

5.2.5.1 Benefits of IPsec

Some benefits of IPsec are shown below:

- The implementation of IPsec in a firewall or router leads to strong security relating to all traffic that crosses the perimeter. There is no overhead caused by security processes when it concerns the traffic within a company or workgroup.
- It is not easy to bypass IPsec in a firewall in case that all outside traffic need to use IP and the firewall consists the only means to access the organization network from the Internet.
- Since IPsec is one layer below the transport layer (TCP, UDP), applications do not perceive its existence. Even if the firewall or router uses IPsec, it is not required to modify the software on a user or server system. Even if the end systems utilize IPsec, there is no impact on the software of the upper layers, including applications.
- The end users do not need to know about the existence of IPsec. It is not necessary to educate users on security mechanisms, or provide keying material based on the user's case, or withdraw this keying material in case that users leave the organization.
- If it is necessary, IPsec can offer security for individual users. This is helpful in the cases of offsite workers and when it is required to create a secure virtual subnetwork for sensitive applications within an organization.

5.2.5.2 Routing Applications

Apart from the ability to support end users and secure the premises of their systems and networks, IPsec can additionally assist tremendously in the routing architecture that is needed for internetworking. IPsec can ensure that:

- A router advertisement (a new router advertises itself) originates from a router that is authorized.

- A neighbor advertisement (a router tries to start or sustain a neighbor relationship with a router existing in a different routing domain) originates from a router that is authorized.
- A redirect message originates from the router who received the initial IP packet.
- A routing update is not forged.

In case that such security measures are not in place, an adversary can intervene with communications or redirect some of the network traffic. Routing protocols, like the Open Shortest Path First (OSPF) protocol should be implemented to operate on top of security associations between routers that IPsec defines.

5.2.5.3 IPsec Architecture

The operation of IPsec depends heavily on the existence of a security policy used on each IP packet that is transferred from a source to a destination. The IPsec policy is defined mainly by the interaction of two databases. These are the security association database (SAD) and the security policy database (SPD). An outline of the two databases is presented below and then follows a summary of their use during IPsec operation.

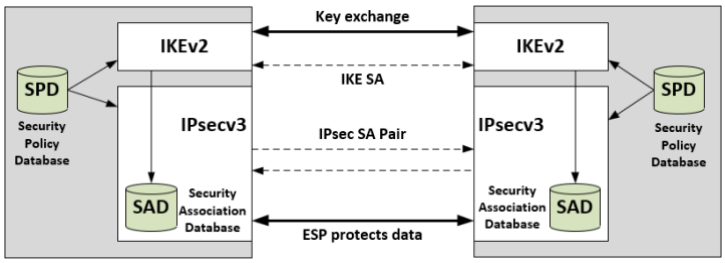


Figure 5-9: IPsec Architecture [1]

5.2.5.3.1 Security Association

The authentication and confidentiality mechanisms regarding IP share the same key concept of the security association (SA). An association constitutes a one-way logical connection between a receiver and a sender. Security services are provided by this connection to the

corresponding traffic. In case of two-way secure exchange that requires a peer relationship, it is necessary to have two security associations.

5.2.5.3.2 Security Association Database

Every IPsec implementation involves a specific Security Association Database whose role is the definition of the parameters related to each SA. The following parameters normally describe a security association in an SAD entry.

- **Security Parameter Index**: This constitutes a 32-bit value designated by the receiving end of an SA to uniquely recognize the SA. In an SAD entry for an outbound SA, the SPI is needed to create the packet's AH or ESP header. In an SAD entry for an inbound SA, the SPI is employed to map traffic to the relevant SA.
- **Sequence Number Counter**: This constitutes a 32-bit value utilized for the construction of the Sequence Number field in AH or ESP headers.
- **Sequence Counter Overflow**: This constitutes a flag that shows if the overflow of the Sequence Number Counter should create an auditable event and stop further sending of packets on this SA (necessary for all implementations).
- **Anti-Replay Window**: This is used to perform a check about an inbound AH or ESP packet being a replay or not.
- **AH Information**: This involves authentication algorithm, keys and key lifespans, and other parameters being utilized with AH (needed for AH deployments).
- **ESP Information**: This involves encryption and authentication algorithm, initialization values, keys and key lifespans, and other parameters being utilized with ESP (needed for ESP deployments).
- **Lifetime of this Security Association**: This involves a time period or byte count (needed for all implementations). After this time period has passed, an SA needs to be changed with a new SA (and new SPI) or terminated. The time period also serves as a sign regarding the type of action (change or termination) that should happen.
- **IPsec Protocol Mode**: Tunnel, transport, or wildcard.

- **Path MTU**: Any observed path maximum transmission unit (maximum length of packet that can be sent without fragmentation) and aging variables (needed for all implementations).

The key management operations, that are employed to deliver the keys, are associated with the authentication and privacy operations only through the Security Parameters Index (SPI). Therefore, privacy and authentication have been characterized as independent of any particular key management mechanism.

IPsec supports the user with substantial flexibility regarding how IPsec services are utilized on IP traffic. As it will mention later, it is possible to combine SAs in numerous ways in order to achieve the preferred user configuration. Moreover, IPsec offers a high degree of granularity in distinguishing traffic with IPsec protection from traffic that is permitted to ignore IPsec, with the former case concerning IP traffic to particular SAs.

5.2.5.3.3 Security Policy Database

The Security Policy Database (SPD) records the means that are used to associate IP traffic with specific SAs (or no SA in the case of traffic allowed to bypass IPsec). In its simplest form, an SPD includes entries. Each entry describes a subset of IP traffic and associates with a specific SA for that traffic. In more complicated implementations, multiple entries may point to one SA or multiple SAs related to a single SPD entry. The reader is advised to use the relevant IPsec documents for a full discussion.

Each of the entries in the SPD includes a set of IP and upper-layer protocol field values, that are named selectors. The use of these selectors is to filter outgoing traffic and map it into a specific SA. The processes regarding outgoing traffic follows the subsequent general three steps for each IP packet.

1. The values of the relevant fields in the packet (the selector fields) are matched against the SPD to find a fitting SPD entry. The entry may associate with zero or more SAs.
2. If this packet relates to a particular SA, it is found along with its respective SPI.
3. The necessary IPsec processing operations are performed (i.e., AH or ESP processing).

5.2.5.3.4 SPD Entries

An SPD entry is determined by the selectors mentioned below:

- **Remote IP Address**: This could be a single IP address, a catalogued list or range of addresses, or a wildcard (mask) address.
- **Local IP Address**: This could be a single IP address, a catalogued list or range of addresses, or a wildcard (mask) address.
- **Next Layer Protocol**: The IP protocol header (IPv4, IPv6, or IPv6 Extension) contains a field (Protocol for IPv4, Next Header for IPv6 or IPv6 Extension) that specifies the protocol employed over IP.
- **Name**: This is used by the operating system as a user identifier.
- **Local and Remote Ports**: These could be individual TCP or UDP port values, an catalogued list of ports, or a wildcard port.

5.2.5.3.5 Encapsulating Security Payload

The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service. If the algorithm, that encrypts the payload, needs cryptographic synchronization data, such as an initialization vector (IV), then these data may be placed explicitly at the start of the Payload Data field. The IV, when included, is usually not encrypted, although it consists part of the ciphertext.

The ICV field is not obligatory. It exists only if the integrity service is chosen and either a separate integrity algorithm or a combined mode algorithm, that uses the ICV field, calculates it. The encryption occurs before the calculation of the ICV field. This processing order enables rapid detection and rejection of replayed or bogus packets by the receiver before the decryption of the packet. Thus, it may diminish the impact of denial of service (DoS) attacks. It additionally encourages parallel processing of packets at the receiver meaning that decryption and integrity checking can occur at the same time. Also, encryption does not protect the ICV, it must be calculated employing a keyed integrity algorithm.

The Padding field serves several purposes:

- An encryption algorithm may need the size of the plaintext to be a multiple of a specific number of bytes (e.g., the multiple of a single block for a block cipher). In this case, the Padding field is employed to assist the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to reach the necessary size.
- The ESP format demands that the Pad Length and Next Header fields are right aligned within a 32-bit word. In the same way, the ciphertext must be an integer multiple of 32 bits. This alignment is assured through the usage of the Padding Field.
- Additional padding could be inserted to ensure partial traffic-flow confidentiality by hiding the real size of the payload.

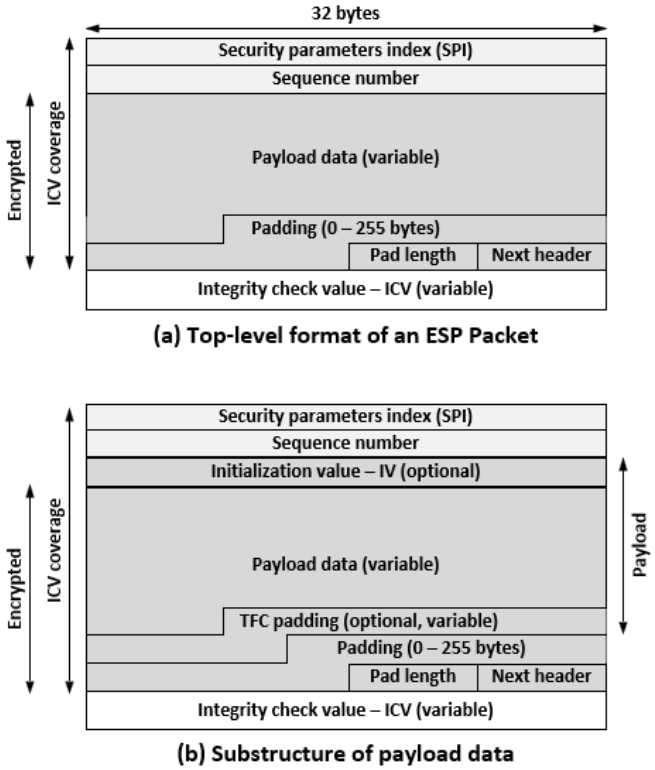


Figure 5-10: ESP Packet Format [1]

Figure 5-10a presents the top-level format of an ESP packet. It involves the fields mentioned below.

- **Security Parameters Index (32 bits)**: This is used to identify a security association.

- **Sequence Number (32 bits**): This constitutes a monotonically increasing counter value which operates as an anti-replay mechanism in AH.
- **Payload Data (variable):** This is a transport-level segment (in transport mode) or an IP packet (in tunnel mode) which is secured using encryption.
- **Padding (0–255 bytes):** The function of this field is elaborated later.
- **Pad Length (8 bits):** This is a number of the pad bytes that exist before this field.
- **Next Header (8 bits)**: This designates the type of data involved in the payload data field by describing the first header in that payload (e.g., an upper-layer protocol such as TCP, or an extension header in IPv6).
- **Integrity Check Value (variable)**: This constitutes a variable-length field (integral number of 32-bit words) that includes the Integrity Check Value that is calculated based on the ESP packet excluding the Authentication Data field.

In case that any mixed mode algorithm is utilized, it is expected that the algorithm will return both the decrypted plaintext and a pass/fail signal concerning the result of the integrity check. For combined mode algorithms, the ICV, which typically exists at the end of the ESP packet (when integrity is chosen) can be omitted. In case that the ICV is omitted and integrity is selected, the combined mode algorithm is supposed to encode inside the Payload Data an ICV-equal way of checking the integrity of the packet.

The payload may include two additional fields (Figure 5-10b). If the ESP employs an encryption or authentication algorithm that needs an initialization value (IV), or nonce, then this IV or nonce is included. If tunnel mode is active, then the IPsec mechanism may insert traffic flow confidentiality (TFC) padding after the Payload Data and before the Padding field and after the Payload Data, as explained subsequently.

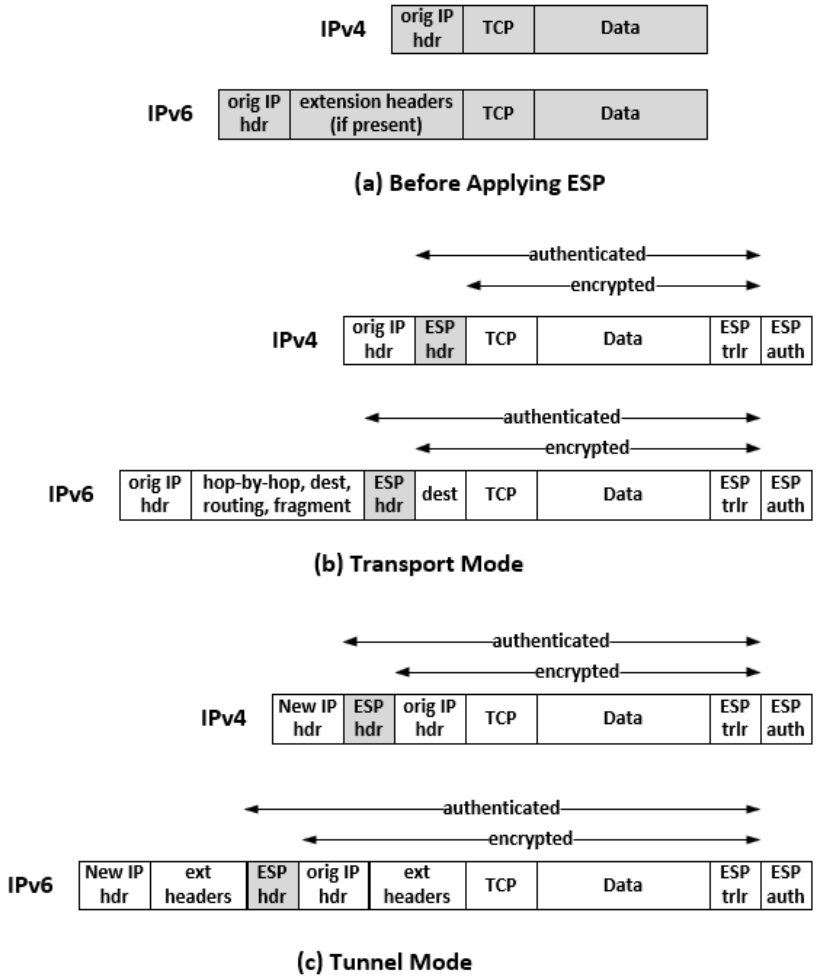


Figure 5-11: Scope of ESP Encryption and Authentication [1]

The packet formats of Figure 5-11a work as a starting point to discuss the scope of ESP for Transport and Tunnel Mode.

Transport mode ESP encrypts and optionally authenticates the data transported by IP (e.g., a TCP segment), as depicted in Figure 5-11b. In case of this mode employing IPv4, the ESP header is added into the IP packet immediately before the transport-layer header (e.g., TCP, UDP, ICMP), and an ESP trailer (Padding, Pad Length, and Next Header fields) is inserted subsequently to the IP packet.

When authentication is chosen, the ESP Authentication Data field is appended after the ESP trailer. Encryption is performed to the whole transport-level segment together with the ESP trailer. Authentication encompasses the entirety of the ciphertext and the ESP header. In IPv6, ESP is visualized as an end-to-end payload; it is not checked or processed by in-between

routers. This result in the ESP header emerging after the IPv6 base header and the hop-by-hop, routing, and fragment extension headers. The destination options extension header could be present before or after the ESP header, based on the preferred semantics. In IPv6, encryption is provided to the entire transport-level segment plus the ESP trailer plus the destination options extension header in case it happens after the ESP header. Again, the ciphertext in addition to the ESP header are covered by authentication.

The synopsis of the transport mode operation is the following.

1. At the source, the block of data comprised by the ESP trailer and the entire transport-layer segment is encrypted and the ciphertext of this block replaces its plaintext to construct the IP packet for transmission. If the option is active, authentication is included.
2. Then, the packet is transferred to the destination. Each in-between router is required to check and process the IP header and any plaintext IP extension headers but the examination the ciphertext is not required.
3. In the destination node, the IP header plus any plaintext is checked and examined.

IP extension headers. Then, based on the value of the SPI in the ESP header, the destination node performs decryption to the rest of the packet in order that the plaintext transport-layer segment can be recovered.

In transport mode operation, confidentiality is provided for any application that uses it. Thus, it is not necessary to apply confidentiality in every individual application. One downside to this mode refers to the possibility of performing traffic analysis on the sent packets.

Tunnel mode ESP is utilized to encrypt one whole IP packet (Figure 5-11c). In this mode, the ESP header is inserted in the beginning of the packet and then the packet followed by the ESP trailer are encrypted. This is a suitable method to prevent traffic analysis.

As the IP header includes the destination address and possibly source routing directives and hop-by-hop option data, the encrypted IP packet prefixed by the ESP header cannot be just sent to the destination. It would be impossible for in-between routers to process such a packet. Thus, the entire block (ESP header plus ciphertext plus Authentication Data, if

present) must be encapsulated with a new IP header that includes enough data for appropriate routing but not for traffic analysis.

Although, the transport mode can protect connections between hosts that implement the ESP feature, the tunnel mode is practical in a configuration that involves a firewall or another type of security gateway that safeguards a trusted network from external networks. In the latter case, encryption is performed either between the security gateway and an external host or between two security gateways. This lessens the burden on the host of the internal network, as it does not need to perform the processing of encryption and the key distribution operation is simplified by decreasing the amount of required keys. Moreover, it counters traffic analysis relating to ultimate destination.

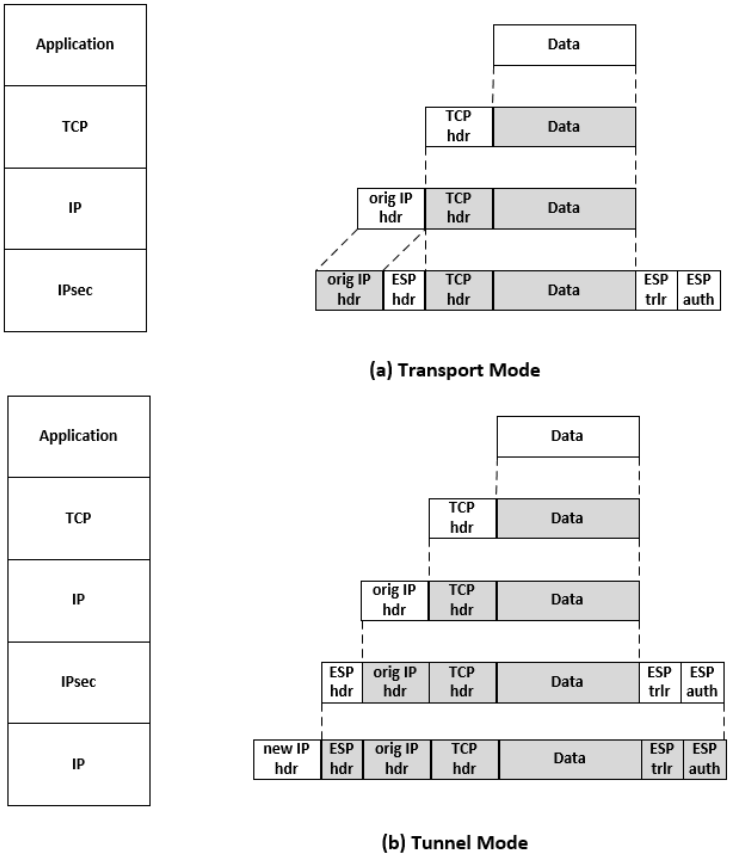


Figure 5-12: Protocol Operation for ESP - Protocol Architecture of the two modes [1]

5.3 Denial of Service (DoS) attacks

Availability is the service that refers to a system that is accessible and usable on demand by authorized users. A denial-of-service attack (DoS) aims to compromise availability by inhibiting or stopping entirely the provision of some service. The purpose of this attack is the exhaustion of a critical resource that is required to provide a specific service.

5.3.1 Description of Attacks

Denial of service is a type of attack impacting the availability of some service. In the case of computer and communications security, the attacks are focused on the network connection of network services. This form is distinguished of attack on availability from other attacks, such as the classic acts of god, that cause damage or destruction of IT infrastructure and consequent loss of service.

NIST SP 800-61 categorizes several types of resources that could be targeted by a **Denial-of-Service (DoS) attack,** including the System resources, the Network bandwidth, and the Application resources.

Network bandwidth is relevant to the capacity of the network links that are used to connect a server with the wider Internet, as depicted in the example network of Figure 5-13. In most cases, the capacity of this connection will be smaller compared to the capacity of the links within and between ISP routers. Thus, more traffic can reach the ISP's routers over these higher-capacity links and it can be transported over the link to the organization. If this happens, the router is forced to discard some packets and keep on delivering only as many as the link can afford to handle. During normal network activity, these types of high loads could happen to a popular server that needs to deal with traffic from a large amount of legitimate users. Some of these users will experience a nonexistent or degraded service as a result. In case of an overloaded TCP/IP network link, this behavior is expected.

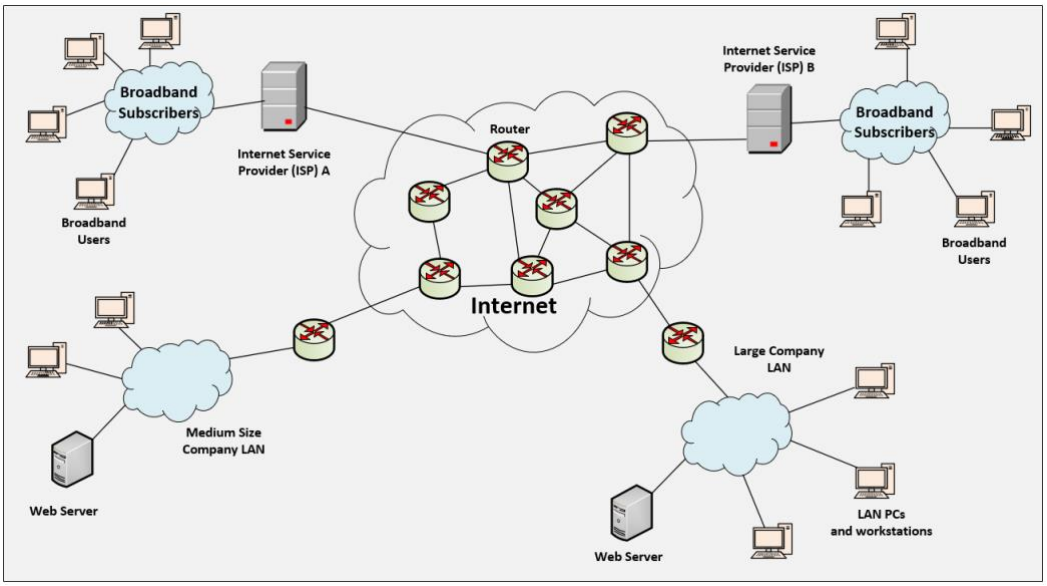


Figure 5-13: Example Network to Illustrate DoS Attacks [1]

In a DoS attack, most of the traffic directed at the target server is malicious and the attacker produces it either directly or indirectly. This traffic surpasses any legitimate traffic, effectively preventing legitimate users from gaining access to the server. Some recent high-volume attacks have even been targeting the ISP network that supports the target organization, with the aim of interfering with its connections to other networks.

A DoS attack, which targets system resources, normally intends to overwhelm or crash its network handling software. Instead of consuming bandwidth by sending a lot of traffic, particular types of packets, that consume the constrained available system resources, are transmitted. These resources may involve temporary buffers that are utilized to store tables of open connections, arriving packets, and other memory data structures.

The simplest and most typical DoS attack is a flooding attack that targets an organization. The goal of this attack is to cause the overload of the capacity of the network link to the targeted organization. Another type is the **distributed denial-of-service** (DDoS) attacks where the attacker uses multiple systems to generate attacks. These systems normally consist of compromised user PCs or workstations. Using malware, the attacker can subvert the system and set up an attack agent under their control. A large number of these systems, that are controlled by one attacker, collectively constitute a botnet.

5.3.2 DoS Attack Defenses

Numerous measures can be taken in order to lessen the consequences of being targeted by a DoS attack and to reduce the chances of systems being compromised and afterwards utilized in DoS attacks. It is necessary to understand that it is impossible to stop these attacks entirely. Particularly, if an attacker is capable of directing a large volume of legitimate traffic to the system, then it is highly probable that the system's network connection will be overloaded and as a result, traffic requests from other legitimate users will be restricted. This can also sometimes happen by accident due to the high publicity of a particular site.

Generally, four lines of defense can be used to counter DDoS attacks:

- **Attack prevention and preemption (before the attack)** mechanisms enable the victim to endure attack attempts without denying service to legitimate clients.
- **Attack detection and filtering (during the attack)** mechanisms attempt to detect the attack as it begins and respond immediately.
- **Attack source traceback and identification (during and after the attack)** mechanisms attempt to identify the source of the attack as a first step in preventing future attacks.
- **Attack reaction (after the attack)** mechanisms attempt to eliminate or curtail the effects of an attack.

5.3.3 DoS Attack Prevention

A crucial component of numerous DoS attacks refers to the usage of spoofed source addresses. The purpose of these is either to hide the source system that creates the direct and distributed DoS attacks or to lead the reflected or amplified traffic towards the target system. Therefore, one of the fundamental, and longest standing, recommendations regarding the defense against these attacks constitutes the limitation of the capability of systems to transmit packets that have spoofed source addresses.

It is necessary to perform this filtering as close to the source as possible, meaning that routers or gateways should know the valid address ranges of the incoming packets. Normally, this relates to the ISP that offers the network connection for a home user or an organization. An ISP possesses knowledge over which addresses are distributed to each of its customers and

thus, it is set up in the appropriate way to assure that all packets from its customers utilize valid source addresses. This way of filtering can be established employing explicit access control rules in a router to make sure that the source address used on any customer packet was distributed by the ISP. Another way to use filters is to ensure that the current packet indeed utilizes the path back to the claimed source address.

Defending specifically against the SYN spoofing attack is possible if a modified version of the TCP connection handling code is employed. Instead of storing the connection details on the server, critical information regarding the requested connection can be cryptographically encoded in a cookie that is transmitted as the server's initial sequence number. This is transmitted in the SYN-ACK packet from the server back to the client. In case that a legitimate client sends a response with an ACK packet that includes the incremented sequence number cookie, the server can then reconstruct the information about the connection that normally would have been stored in the known TCP connections table.

Alternatively, it is possible to modify the system's TCP/IP network code in order that an entry for an incomplete connection can specifically dropped from the TCP connections table if it overflows. This helps a new connection attempt to proceed normally. This approach is also named as *selective drop or random drop*. Assuming that most entries in an overflowing table originate from the attack, it is highly probable that the dropped entry associates with an attack packet. Thus, no consequence comes from its removal. If not, then a legitimate connection attempt will fail, and there will be a need for retry. However, this approach means that new connection attempts have a chance of succeeding instead of being dropped urgently if the table has overflowed.

Another way of defending against SYN spoofing attacks involves the change of parameters that are utilized in a system's TCP/IP network code. These parameters involve the size of the TCP connections table and the timeout period that is set up in order to delete entries from the table if there is no received response. It is possible to combine the modifications to the parameters with appropriate rate limits on the organization's network link to determine the maximum allowable rate of connection requests. It is important to note that these changes cannot prevent SYN spoofing attacks, but they will increase the difficulty of the task of the attacker.

The best defense against broadcast amplification attacks refers to the blocking of the use of IP-directed broadcasts. This is achievable either through the ISP or through any organization whose systems could operate as an intermediary. As mentioned before, this and anti-spoofing filters are long-standing security recommendations that should be implemented by all organizations. In other words, constraining or cutting off traffic to suspicious services, or combinations of source and destination ports, can lessen the types of reflection attacks that target an organization.

The defense against attacks that aim at the application resources typically necessitates changes to the targeted applications, such as Web servers. Defenses may include attempts to discriminate between legitimate, human-initiated interactions and interactions stemming from automated DoS attacks. These often means the use of a graphical puzzle, a captcha. The graphical puzzle is easily solved by most humans, but difficult to automate.

Apart from implementing these defense mechanisms to prevent DoS attack, it is important to maintain proper system security practices. The purpose of these practices is to protect the systems from being compromised and turned to zombie systems. Also, potential intermediary servers such as high performance, well-connected servers should be properly configured and monitored in order that they do not contribute to the problem.

Lastly, in case that an organization relies on network services, mirroring and replicating these servers over multiple sites with multiple network connections should be considered. High-performance servers can benefit from this type of general practice, which offers increased reliability and fault tolerance and not just a response to these attacks.

5.3.4 Responding to DoS Attacks

It is necessary to have a good incident response plan in case of a DoS attack. This plan should involve details regarding the ways to communicate with technical personnel from your Internet service provider(s). This contact must be feasible with nonnetworked means, because when under attack there is a high possibility that your network connection cannot be used. DoS attacks, specifically flooding attacks, can only be filtered upstream of your network connection. The plan also needs to involve details concerning the response to the

attack. The way that responsibilities will be divided between organizational personnel and the ISP will rely on the usable resources and the technical capabilities of the organization.

After the detection of a DoS attack, first of all, it is important to understand what is the type of attack and what is the best approach against it. Normally, this means that packets flowing into the organization should be captured and analyzed in order to check for common attack packet types. Organizational personnel can perform this task by utilizing appropriate network analysis tools. In case that the organization does not possess the resources or skill to achieve this, the capture and analysis needs to be done by its ISP. This analysis assists in recognizing the type of attack and subsequently, appropriate filters are employed to restrict the flow of attack packets. The ISP have to install these filters on its routers. If the target of the attack is a bug on a system or application, rather than high traffic volumes, then it is crucial to identify the bug and correct it in order to prevent future attacks.

After the immediate response to this attack, the organization's incident response policy may designate some additional measures that will be taken to respond to incidents like this. The measures should surely involve the analysis of the attack and response. This can assist in acquiring experience and improving future handling. In an ideal situation, the result can be the enhancement of the security of the organization.

5.4 Referencing

[1] Stallings, W., Brown, L., Bauer, M. D., & Bhattacharjee, A. K. (2012). *Computer security: principles and practice* (pp. 978-0). Upper Saddle River, NJ, USA: Pearson Education.

[2] MATT, Bishop. (2018). *Computer security: art and science*. 2nd Edition, Addison-Wesley Professional, ISBN: 9780134097145.

[3] Easttom, C. (2019). *Computer security fundamentals*. Pearson IT Certification.

[28] Stallings, W. (2003). Network Security Essentials: Applications and Standards, 4/e. Pearson Education India.

6 Communications & Network Security: Securing network components

Author(s): **Fabrizio Granelli**

6.1 Introduction to securing network components

The security of a networked computer system depends from the security of its components. In order to analyse the security of the single components of the system, an important concept to introduce is the attack surface. When security experts refer to the attack surface, they mean the cumulation of all of the vulnerabilities for a specific infrastructure item.

Security professionals analyse the attack surface of individual devices with the purpose of understanding potential threats that could endanger the system. For example, we can consider the attack surface of a gateway router. In this case, the attack surface is evaluated by considering how a potential attacker might interact with the router, including the possibility to alter its configuration (mis-configuration), gaining access to its software, or even physically stealing the device itself.

This section is aimed at defining and exploring the attack surface of typical network devices. In the case of typical networked devices, this includes hardware, software, connections, and possibly applications. All those components might be vulnerable to attacks and therefore insecure.

The next sections analyse the attack surfaces of individual system from multiple viewpoints, including hardware and software, operating systems as well as applications.

6.2 Types of System Attacks

An attacker can perform different actions to impact on the security of network components. Such attacks can be clustered in the following types:

- Attacking applications
- Attacking the OS
- Attacking the network stack
- Attacking drivers

Let's consider a network component. To function in a network, a device uses different functionalities:

- Networked applications (that offer services)

- Operating system with networking functionalities
- Network protocol stack (to manage, received and transmit packets)
- Software drivers (for operating the network interfaces)

Taken together, those functionalities represent the actual attack surface of the device.

The next sub-sections provide additional information on each of those aspects.

6.2.1 Attacking applications

It is extremely common for attackers to consider the applications. Indeed, networked applications represent a relevant component of the overall attack surface of a networked device.

This is due to the fact that the number of available applications is extremely high today, and that the lack of proper patches or support makes it easy to identify and exploit the related security issues.

The most common form of attack against applications is the man-in-the-middle. The concept of this attack is to typically target the authentication or encryption systems and to pretend to have the right to access or impersonate an authorized user.

6.2.2 Attacking the Operating System

If users fail to perform good and timely patch management, operating systems can represent an excellent attack surface. Indeed, users are typically not aware about the danger of leaving the operating system outdated or not properly patched.

The most common attack to the operating system is privilege escalation. Privilege escalation represents an attack consisting of using a low authentication level to access the system and then to find ways to upgrade the authentical to achieve more access rights and privileges.

Operating systems are generally designed to avoid this attack, however attackers might identify a bug through an application or an alternative login. Some operating systems, as an example, offer a “guest” login account, with limited privileges. However, in some cases,

attackers might enter the system using such guest account and then find weaknesses in the operating system that could allow privilege escalation.

6.2.3 Attacking System Services

Operating systems services represent another potential attack surface option. Indeed, attacking processes running in the operating system is like attacking applications, therefore the same concepts discussed in the corresponding section are valid for these cases, too

6.2.4 Attacking Network Stacks

A popular attack strategy is to attack the TCP/IP protocol stack. This strategy might be considered the second one within common attacks, typically after attacks to applications. Historically, TCP/IP was built on an utopia where all users of the networks are considered friendly and collaborative. In reality, this design principle represents a serious security problem, since almost all traditional TCP/IP suite protocols we not designed for secure communications.

Examples of potential attacks to the TCP/IP stack include:

- IP/MAC spoofing: Spoofing means “stealing, therefore this attack is implemented by pretending to have a different identity (e.g., address) at IP or MAC layers. In practice, the attacker “impersonates” a different host (i.e., “steals” an IP or MAC addr.).
- ARP poisoning: this attack exploits the fact that ARP protocol is a plug-and-play self-configuring facility. The device interested in solving a specific IP address into the corresponding MAC address sends an ARP broadcast request, and typically the owner of such IP address will answer and enable the originating device to populate its ARP table. A good ARP poisoning utility floods the network with ARP responses, filling other systems with spoofed ARP data in their caches, in order to generate false entries and saturate the ARP table with false/forged information.

6.2.5 Attacking Drivers

Every hardware component of a computer communicates with the operating system via device drivers. Modifications might be performed on such software components in order to implement different kind of attacks. The most popular driver attacks are:

- Refactoring: It represents a popular attack to the device drivers, implemented by modifying the drivers in order to maintain normal operation and generate the requested service, while enabling additional malicious output. E.g.: an attacker might refactor a printer driver in order to maintaining to work properly, but at the same time to sending all print jobs to a malicious remote server.
- Shimming: Shimming is similar to refactoring, but simpler: it involves modifying the driver to listening to inputs that the original device driver isn't written to handle. In practice, the driver still works properly, but it offers unknown interfaces to malicious devices.

6.2.6 Denial of Service

A Denial of Service (DoS) attack is aimed at disrupting service continuity, by denying the service provided by an application, the operating system itself, or a server. It is a very common and dangerous type of attacks.

Typical DoS attacks range from simple continuous ping packets to sophisticated DoS applications. The common idea behind a DoS attack is to drain the resources that a service or application needs in order to make it not usable by its legitimate users.

However, generating a DoS attack from a single position in the networks is easily detected and can be simply mitigated. For this reason, a recent evolution of the DoS attack is the Distributed DoS attack (DDoS). The Distributed DoS attack uses many geographically distributed attackers to increase the impact of the attack and to become hard to detect.

6.3 System Resiliency

Once we reviewed the typical types of attacks our network components might be subject to, it is time to move to identify which strategies to use in order to avoid or mitigate their effects. This leads us to the concept of system resiliency.

System resiliency represents a mitigation strategy focused on providing tools to the system to being able to recover from attacks easily and efficiently. Therefore, resilient systems do not eliminate risks (that's impossible), but they "handle risks better". Indeed, a resilient system might make the work of system administration easier.

The methodologies to make a system resilient can be categorized into the following categories: (1) non-persistence, (2) redundancy, and (3) automation.

6.3.1 Non-persistence

A typical individual system and its configuration are persistent. This means that the system has a fixed set of hardware, a fixed operating system, and a fixed configuration. In this framework, the attack will modify its persistent state or change some aspects.

Once the attack happens, the recovery of the original state requires costly and time-consuming processes.

Three options are available to implement non-persistence:

- Virtualization/snapshots
- Revert/rollback tools
- Live boot

6.3.1.1 Virtualization/snapshots

Virtualization represents a useful tool to implement non-persistence. Virtualization enables to isolate a subset of the system resources (memory, CPU, storage, etc.) and to organize them in order to host a virtual machine (VM). Several VMs can be hosted on the same hardware device, or in a cluster.

Virtual Machines brings several advantages, including flexibility and isolation. Using virtual machines enables the system to gain agility, since VMs can be shut down and rebooted easily without affecting the other ones nor the underlying hardware.

A key element for fast restoration of VMs is the snapshot. It represents an archive containing the difference between VM filesystem in one version and in another version. Snapshots represent a critical non-persistence tool for VMs, since they enable to recover the state of the VM before the attack in a simple manner and in short time.

For this reason, it is advisable to run a snapshot whenever a driver is changed, an application is updated, or an OS is patched within the VM.

6.3.1.2 Revert/roll back tools

Today, most Operating Systems offer at least one revert/rollback method, so that it is possible to restore the system to an earlier state. Indeed, this is a diffused and useful functionality.

In Microsoft Windows it is possible to create restore points, that enable to revert to earlier configurations of the OS. The user can restore one of the previously recorded state of the system.

Apple MacOS implements a specific tool for performing snapshots of the OS and enable to access previous machine states. It is called Time Machine, a program that records snapshots of the state of the computer over time on an external drive. The Time Machine has the capability to restore a previous state of the system, or to browse the filesystem and recover deleted or modified files.

The process of revert/roll back can be implemented by using VMs and automation functionalities. In many modern networks, the usage of virtualization allows to implement rollout and rollbacks for several systems at the same time. In those cases, a central server pushes a master image to every local computer/VMs.

6.3.1.3 Live boot

Live boots represent the ultimate solution in non-persistence on a non-VM system. By using live boot media it is possible to run complete operating systems without installing the OS on the machine. Indeed, such OSs exist only on such bootable media, therefore they can be executed on a generic hardware and they are capable of restoring the entire operating system and services at the desired state.

This solution might be appropriate for cases in such there is no need to store any persistent data, and it allows to run different combinations of OS and applications directly from the live boot media.

In the area of cybersecurity, an interesting and popular live boot Linux distribution is Kali Linux, which provides several already-installed software and services dedicated to IT security.

6.3.2 Redundancy

Resiliency can be achieved by means of non-persistence in several cases, but it does not work quite well in the case of data storage. In fact, snapshots/reverts are most appropriate for small or invisible modifications of the operating systems state, however they are not adequate in cases such as customer databases, inventories, users' file storage, where big amounts of data are recorded.

In this situation, the required system resiliency can be achieved through redundancy.

It is possible to create redundant mass storage, redundant systems and even redundant networks.

The major benefits of redundancy include:

- Fault tolerance: redundancy allows the secondary storage, the system and the network to be online and ready to go. In case of attacks, the system will experience a minimal amount of disruption as the secondary devices will be used as substitutes for the primary ones.

- High availability: through redundancy it is possible to increase the availability of data and services since it is possible to balance performance against risk. This is a crucial aspect in highly available systems.
- Minimal impact of offline services: attack on one of the services will have minimal impact on service delivery, since the system will be able to automatically react to anomalies and trigger redundant resources.

6.3.2.1 Storage redundancy

The most relevant technology to implement storage redundancy is RAID. The Redundant Array of Independent Disks (RAID) is a fault tolerance technology that distributes and duplicates data as needed across multiple hard drives.

RAID is implemented by deploying a set of physical hard drives. Those hard drives are used to define logical drives, typically in a way transparent to the users. Data and metadata are spanned across the different physical drives. If any one of the drives fails, the storage is service is still maintained by the other drives in the RAID without any service interruption. This means that users will not perceive any service degradation while the faulty drive is replaced.

The following figure provides a description of the different RAID configurations:

Table 6-1: RAID configurations

RAID level	Details	Minimum Number of Physical Drives Required
RAID 0	Disk striping; does not use mirroring or parity; provides for performance only with no redundancy	2
RAID 1	Disk mirroring; all data is completely duplicated on both disks; uses no striping or parity but provides for full redundancy at the expense of the loss of half the total available disk space for duplication	2
RAID 5	Disk striping with parity; parity information is spread across sll disks evenly; 1/n of the total disk space available is used for parity	3 to n
RAID 6	Disk striping with double distributed parity; this allows for failure of up to two drives	4
RAID 1+0 (or RAID 10)	Disk mirroring with striping; combines both RAID levels 0 and 1 for performance and redundancy; a stripe of two mirrored arrays	4
RAID 0+1	Disk striping with mirroring; combines both RAID levels 0 and 1 for performance and redundancy; a mirror of two striped arrays	4

6.3.2.2 System redundancy

RAID is certainly a suitable technology for mass storage fault tolerance. However, it does not apply to redundant systems.

In systems, fault tolerance is achieved by means of:

- Load balancing: multiple servers offering the same service are deployed. Load balancing is a high availability technology that shares the load among multiple servers by associating the service requests to multiple serves in a balanced way. In case of a fault, the anomalous system is simply excluded from the group.
- Clustering: multiple identically configured servers are deployed, as in the previous case. Clustering allows to run those servers in parallel, while still appearing as a single logical server. In this way, the failure of one server will not affect the overall service, as the others continue to operate in a transparent way.
- Virtualization: virtualization solutions provide all the benefits of redundancy, especially high availability, with extreme benefits in terms of costs and time. Major benefits of virtualization include load scalability, geographical scalability, and elasticity.

6.4 Securing Hardware

Secure hardware represents the last line of defence for physical attacks on network components. Securing hardware means limiting the possibility that physical attacks might generate security threats.

Today, virtually every piece of hardware comes pre-packaged with proper tools that enable the system to be locked down. In this way it is possible to secure individual computers from physical attacks.

The following subsections outline some potential attacks and technologies to prevent them.

6.4.1 Avoiding Interference

Energy sources might damage or interfere with our systems in different ways:

- Electromagnetic interference (EMI): in general, EMI prevents communication by interfering with signals passing through media. Electromagnetic interference can be generated by other communication systems in the radio-frequency domain or by other factors (e.g., a microwave oven, an electric engine, etc.).
- Radio-frequency interference (RFI): it is an EMI that transmits in the radio frequency range. If strong enough, it might even interfere with speakers and copper transmission lines.
- Electromagnetic pulse (EMP): EMP is the discharge of an electrical current through the air. EMP can lead to physical destruction of equipment. EMP can be caused by electrostatic discharge (ESD) or lightning. Most buildings are protected by lightning, while ESD might represent a "hidden" threat since it might happen due to someone touching some electronic component without the required attention.

6.4.2 Securing the Boot Process

Secure components require secure booting. Indeed, a potential attacker might reset or restart the system in order to attack it.

Securing components is achieved by:

- Securing the supply chain: e.g., see Open Trusted Technology Provider Standard (O-TTPS) – best practices for supply chain security.
- Using Trusted Computing Group (TCG) Technologies: TCG is responsible for all the most popular technologies that make our system very secure. Using TCG technologies supports physical system security.
- Using Trusted Platform Module (TPM) for Security: modern personal computers rely on Unified Extensible Firmware Interface (UEFI) to guarantee a secure boot process (e.g., to prevent malicious changes in the low-level configuration). This might limit boot process attacks.
- Disk encryption: it is now more useful in the presence of TPM. Every operating system supports full disk encryption (FDE) today, as this reduces the possibility of fast access to stored data on the disks.

- Incorporating Hardware Security Modules (HSMs): they are any type of hardware used to offload security-related operations to (e.g., key handling, encryption, etc.). HSMs can be introduced in case of low computational platforms, or to increase the robustness of the system.

6.5 Securing Operating Systems

In a computer network, we have different types of devices. Each device is equipped with an operating system in order to function properly, and for this reason each operating system should be secured.

The following represent the most popular systems that can be found in a computer network:

- Server: the servers (e.g., MS Windows Servers, or Linux servers) supports services and applications. The server operating systems have several features, as required in order to support multiple types of server applications.
- Workstation: it is a common type of computing equipment. Typically, it hosts the main version of the OS (e.g., Windows 10, macOS, etc.).
- Mobile: a mobile OS provides specific support for a limited amount of hardware and it is always pre-installed and pre-configured.
- Appliance: an appliance OS is the firmware powering the devices designed to perform a specific function (e.g., IEEE 802.11 WAPs).
- Kiosk: kiosks have open access to the public and are equipped with OSs that do not allow users to open a terminal or increase privileges, nor to execute non necessary applications.

Hardening an operating system means configuring the OS and setting security options appropriately. Moreover, it also means to maintain the OS updated and install all the important patches. In this way, the system security is increased.

The following list provides some points to consider to hardening an operating system:

- Using a Trusted OS: A Trusted OS is a specialized version of an operating system, created and configured for high-security environments (e.g., Trusted Solaris, SE Linux,

Trusted AIX, also Windows 10 and Windows Server with specific configuration settings).

- Disabling unnecessary ports and services: a host that is running unnecessary services might represent a security concern, therefore the operating system manager can disable ports and services that are not necessary for the system to work.
- Host-Based Firewalls and Intrusion Detection: firewall and intrusion detection systems allow into the host only the traffic necessary for the host to perform its function. Their basic concept is similar to disabling ports, but in this case, it is implemented with additional software components. Their presence on a host increase the security to external attacks.
- Application whitelisting and blacklisting: it is possible to define which applications are allowed or not allowed to install or in the network, in order to guarantee the presence of known software in the network or on the hosts. This makes it easier to maintain the system updated and safe.
- Disable accounts/passwords: unnecessary accounts (e.g., Windows guest) should be disabled or deleted, since they could be used to perform privilege escalation.

After all this discussion, you might be interested in checking which are the open TCP or UDP connections (network connections) in your system. This is possible by using the system command “netstat” (or “netstat –a”, to add listening ports).

```
fabrizio@Linux:~$ sudo netstat

Active Internet connections
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4       0      0  mbp-di-fabrizio..59181 40.114.211.99.https    ESTABLISHED
tcp4       0      0  mbp-di-fabrizio..59180 mrs08s05-in-f14..https ESTABLISHED
tcp4       0      0  mbp-di-fabrizio..59179 mrs08s05-in-f14..https ESTABLISHED
tcp4       0      0  mbp-di-fabrizio..59178 hpbce92f053349.f.http  ESTABLISHED
tcp4       0      0  mbp-di-fabrizio..59177 hpbce92f053349.f.http  ESTABLISHED
tcp4       0      0  mbp-di-fabrizio..59175 ec2-3-123-248-34.https ESTABLISHED
tcp4       0      0  mbp-di-fabrizio..59170 ec2-34-197-166-2.https ESTABLISHED
tcp4       0      0  mbp-di-fabrizio..59169 ec2-52-206-222-3.https ESTABLISHED
tcp4       0      0  mbp-di-fabrizio..59168 13.107.42.23.https     ESTABLISHED
tcp4      31      0  mbp-di-fabrizio..59164 ec2-52-202-62-25.https CLOSE_WAIT
tcp4       0      0  mbp-di-fabrizio..59163 13.89.202.241.https    ESTABLISHED
...
```

Listing 1-1: Example of operation of the netstat command.

6.6 Referencing

[29] Jernigan, M. M. S. (2018). Mike Meyers' CompTIA Security Certification Guide.

7 Communications & Network Security: Securing communication channels

Author(s): **Fabricio Granelli**

7.1 Introduction

This section is focused on securing communication channels. Security professionals use different tools to secure access to a TCP/IP network. Those will represent the first tools that will be analysed. In the following subsections, we will study how to improve security in data communication and how to secure virtualized environments.

7.2 Securing Network Access

Securing network access involves implementing solutions for avoiding malicious files and users to enter the network (typically the Intranet of a company).

Securing network access can be implemented by (a) avoiding that malicious traffic enters the network, and (b) detecting whether malicious files / executables are already present within the network.

Consequently, secure network access can be implemented by means of the tools introduced in the next sub-sections.

7.2.1 Anti-malware programs

Almost all anti-malware tools include a point scanner and a mass storage scanner.

Point scanners, also called real-time scanners, are used for real-time scanning of all incoming data, searching for malware signatures. The purpose of the point scanner is to identify and avoid malware from entering the system, by operating proactively.

Mass storage scanners perform a similar action, but on mass storage. Therefore, in those cases, malware might be already active within the network devices.

7.2.2 Data execution prevention

Data Execution Prevention (DEP), and in general executable space protection methods, prevents unauthorized software to be run on one system.

DEP is almost always on by default on Windows and Mac OSX operating systems. However, it is possible to disable it from the system settings.

In general, DEP should always be active, silently protecting your systems. The need to turn DEP off is quite rare.

7.2.3 File integrity check

File integrity check is used to verify whether some portions of the file system are modified by malware. Most file integrity systems use a hash of the file itself, but it is possible to include also date/time, version, digital signature and other meta data (e.g. modify date).

Typically, hash values are recorded at the installation of the operating system or applications, so that it is possible to easily verify whether the files were manipulated afterwards.

7.2.4 Data loss prevention

Data loss prevention software aim to keep the data integrity at 100%, but in several cases they also include additional functionalities, such as backups.

As an example, DLP might be implemented in email gateways, along with a spam filter to delete unsolicited messages and to secure integrity of emails.

7.2.5 Application whitelisting

It simply consists of a list of applications that the users might install or run on their system. Each operating system allows to install a policy to control which applications might be run.

For example, in Windows operating system it is possible to enable installation and execution only of applications indicated by the Group Policy Editor.

7.2.6 Firewalls

A firewall is a software or device that inspects IP packets and filters them based on preselected criteria. A network firewall operates a shield to defend the network from unwanted or potentially malicious traffic.

A host-based firewall assumes the presence of a network firewall, and it focuses on the input and output traffic from a specific host, to prevent unauthorized executables from accessing the network without the network administrator's permission.

A good example of host firewall is Windows Firewall. However, it is possible to find software firewalls for any operating system.

7.2.7 Intrusion detection

In general, intrusion detection systems allow to detect the moment someone/something is trying an unauthorized access to your network.

More in details, we can define Host-based Intrusion Detection Systems (HIDS) and Host-based Intrusion Prevention Systems (HIPS). The former (HIDS) detect intrusions and report those to the host manager. The latter (HIPS) actively work to stop an intrusion, hopefully preventing it.

Similarly, network-based intrusion detection systems (NIDS) and network-based intrusion prevention systems (NIPS) represent an extension of host-based intrusion detection and prevention systems for individual systems to protect an entire network.

7.3 Securing Data Communication

Network security is an important part of security, and it starts with a proper implementation of a secure LAN.

7.3.1 Organizing an Intranet

Any proper network infrastructure is based on an organization into specific zones or topologies. This partitioning is aimed at deploying the network functions and services in the most efficient and secure way.

A typical architecture of an Intranet is as presented in the following figure.

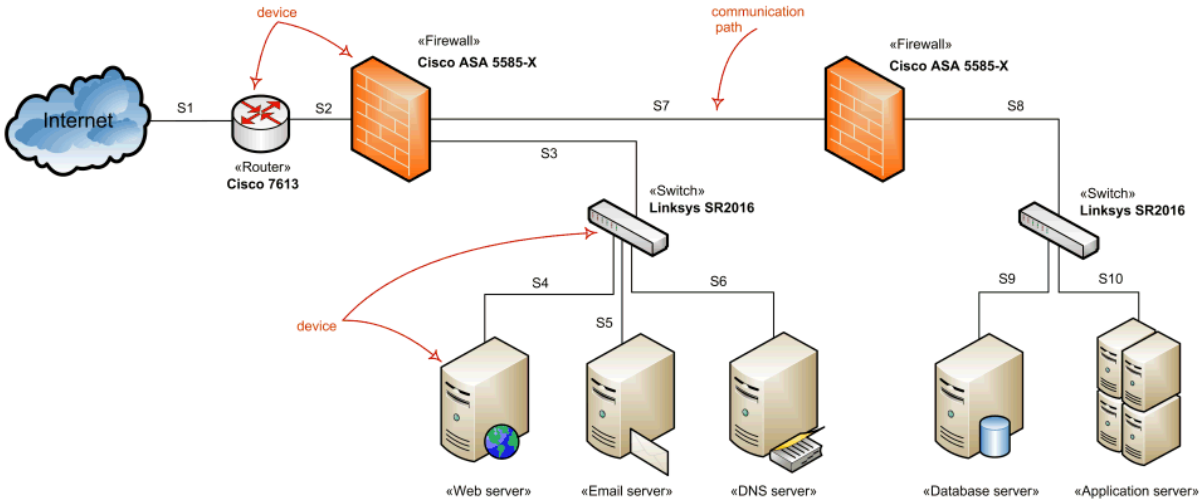


Figure 7-1: Sample Intranet Architecture (https://www.uml-diagrams.org/network-architecture-diagrams.html).

The Intranet will be composed by:

- Switches: a switch connects individual hosts to a broadcast domain. Simplest LANs have a single switch, however most typical deployments we can have one switch per floor, connecting all the hosts of the floor, and then an aggregation switch connects those switches together as well as a router for Internet connection. Generally, the physical LANs are re-organized by using Virtual LANs or (VLANs) to logically separate and re-group the hosts.
- Routers: a router is an IP network node. Its purpose is to switch traffic, by forwarding IP traffic from one LAN to another. Nevertheless, while routers are used to interconnect LANs, they also provide a way to separate them being layer 3 devices: (1) routers block all layer 2 broadcasts between LANs; (2) security experts use router's

separation of LANs as strategic locations to inspect and block some packets traversing different LANs.

- Network firewalls: a network firewall is a network device which purpose is to filter IP traffic. IP filtering is based on a set of rules generically defined access control lists (ACLs).
- DMZ: a demilitarized zone (DMZ) is a LAN where public servers are allocated. The DMZ is separate from the internal LANs that typically contain workstations and private servers. The DMZ is connected to the Internet as well as with the internal LANs. However, while the internal connectivity is implemented by means of an aggressive firewall protecting a router, access to the Internet is provided via a lightly firewalled router.
- NAT: from a security perspective, network address translation (NAT) provides the advantage that internal private addresses are protected and hidden from external networks and clients.
- VLANs: Virtual LAN (VLAN) creates a logical overlay network at layer 2, where hosts can be assigned. For a host assigned to a VLAN, the situation is exactly the same as if it were physically a part of an actual LAN. Different kinds of VLAN membership are possible: port-based VLAN, MAC-based VLAN and protocol-based VLAN (e.g. any client with an IP address on a subnet would be assigned to a VLAN by default). VLANs contribute to security since they allow the creation and isolation of separate logical networks on top of a single infrastructure.

7.3.2 TLS/SSL

Secure data transfer over an IP network can be provided by means of Transport Layer Security (TLS) or Secure Sockets Layer (SSL). Today TLS and SSL are widely spread on the Internet for several applications, including web browsing, email, instant messaging, and voice over IP (VoIP). The purpose of implementing TLS is to enable websites and other services to secure communications between their servers and web browsers.

Transport Layer Security protocol is an Internet Engineering Task Force (IETF) standard. Current protocol version is TLS 1.3, as defined in RFC 8446 (August 2018).

The security features of the TLS protocol are primarily focused to provide privacy and data integrity between two or more communicating computer applications. The TLS connections between a client (e.g., a web browser) and a server are characterized by the following properties:

- Symmetric cryptography is employed to encrypt the data in order to make the connection private (or secure). Symmetric encryption is obtained by generating unique keys for each connection. Uniqueness is ensured by a shared secret negotiated at TLS Handshake level at the beginning of the TLS session. The TLS Handshake procedure is implemented before starting the actual transfer of data, and it allows the server and client to negotiate which encryption algorithm and cryptographic keys to operate. TLS implements a secure and reliable shared secret negotiation, since (1) the negotiated secret cannot be intercepted by eavesdroppers and cannot be eavesdropped even in the middle of the connection, and (2) no attacker can alter the negotiation messages without being detected.
- At least one of the involved parties is authenticated (most likely the server). However, authentication is optional. In case identity authentication is enabled, the process is performed through public-key cryptography.
- The connection is reliable (i.e. data is not lost or altered during the session). Message integrity check is implemented by using a message authentication code which is include in each transmitted message.

Moreover, it is possible to configure TLS in other to implement additional privacy-oriented properties, such as forward secrecy – that does not allow future disclosure of encryption keys to enable decryption of past TLS transactions, as well as selecting different methods for exchanging keys, encrypting data, and verifying message integrity.

TLS was subject to several refinements aimed at removing vulnerabilities and strengthening security. Along this line, TLS 1.3 implements a shortened TLS/SSL handshake and 0-RRT session resumption in order to reduce latency, thus resulting a relevant performance increase in terms of communication latency.

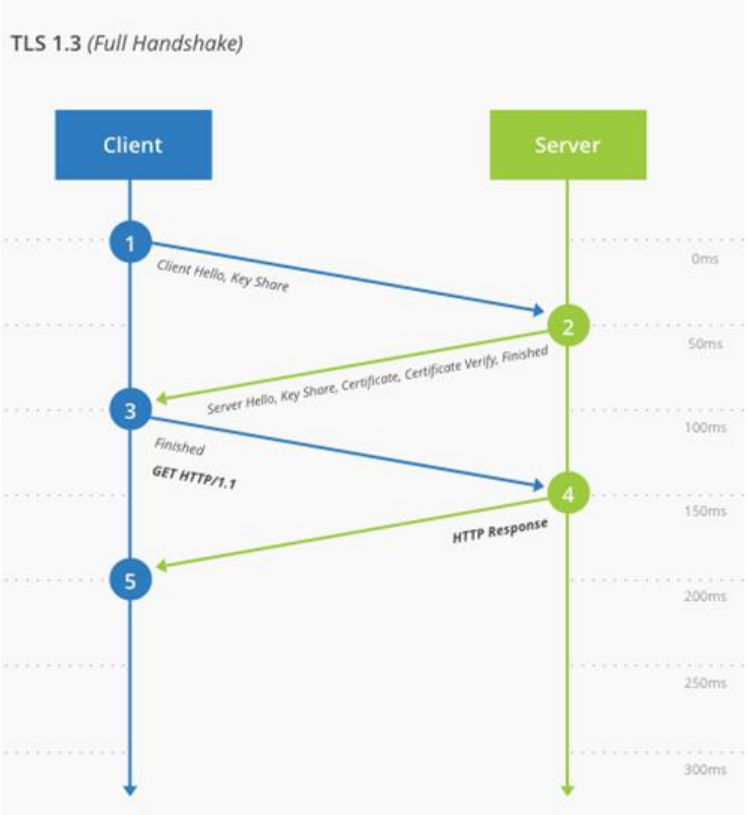


Figure 7-2: TLS 1.3 Handshake (https://www.cloudflare.com/it-it/learning-resources/tls-1-3/).

7.3.3 Virtual Private Networks

A virtual private network (VPN) is implemented to connect two local private networks across a public network. Users are enabled to send and receive data across shared or public networks as if their computing devices were directly connected to the private network. To ensure privacy and data protection, typically encryption is also integrated with the VPN connection. However, this is not always required. By using a VPN connection, connected applications and services can benefit from the functionality, security, and management of the private network.

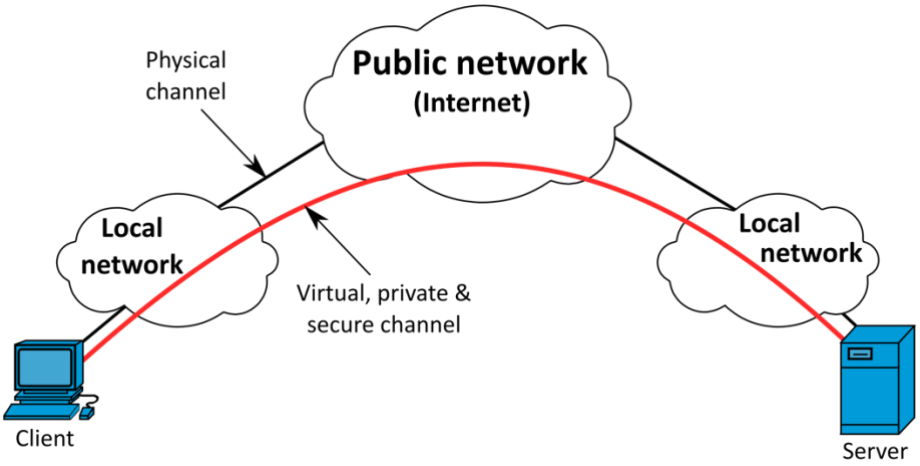


Figure 7-3: VPN example (Wikipedia)

VPN technology was developed to provide secure access to corporate intranets from the public Internet, and to connect geographically remote offices. VPN connection is implemented by means of an encrypted tunneling protocol. VPN supports different methods to implement users' authentication.

VPN is also used in different application scenarios, for example when Internet users may want to secure their Internet connections while being mobile or out of office. Indeed, the VPN allows to avoid geo-blocking and censorship policies, and by using proxy servers a VPN can be used also to protect personal identity and location in order to anonymously browse the Internet.

The usage of VPNs to avoid geographical restrictions can be limited by blocking access to known IP addresses used by VPNs. However, some VPN providers developed alternative strategies to get around such limitations.

From the technical viewpoint, the VPN is created by deploying a virtual point-to-point connection. Such connection can be implemented by using tunnelling protocols over existing networks, such as the Internet, or by exploiting dedicated circuits. The major advantage of a VPN is that it provides secure access to resources within a private network, as if the user would be physically inside it.

Given the above variety, the figure below provides a summary of the different ways to implement VPNs.

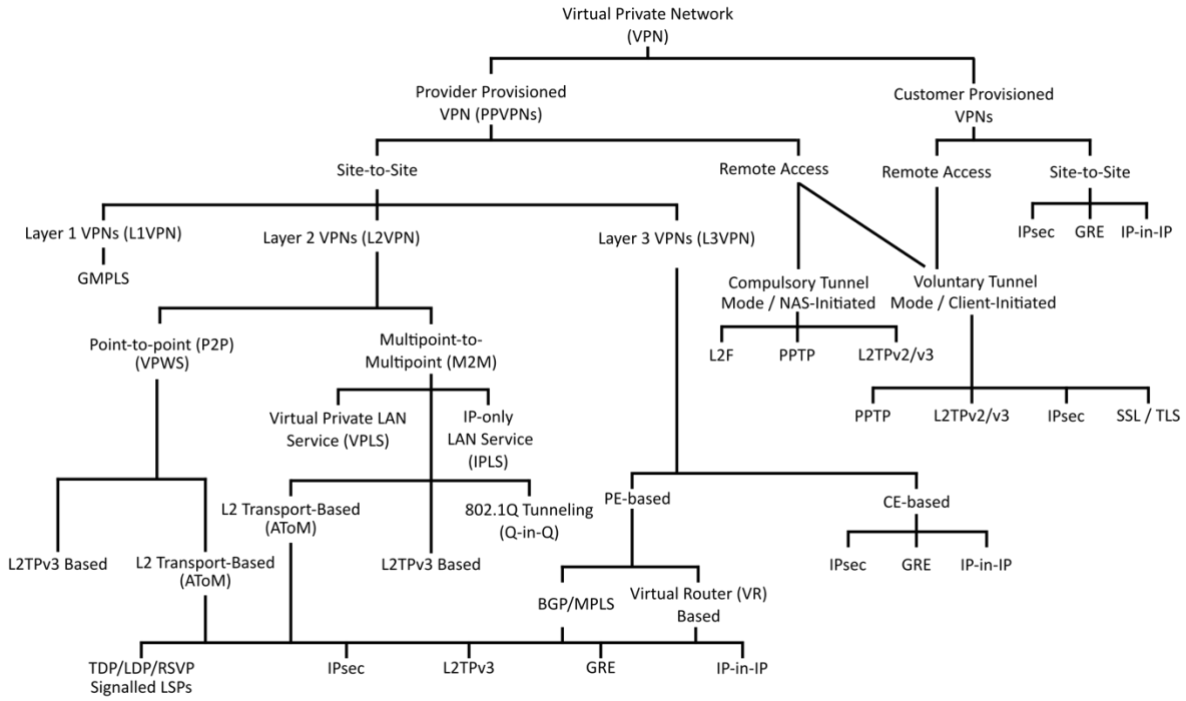


Figure 7-4: Different types of VPNs (Wikipedia)

As discussed above, VPNs can be used to increase privacy and security in accessing private resources.

In terms of security, a VPN provides:

- Confidentiality: data flows are encrypted, so that even sniffing packets would result in acquiring only encrypted data;
- Sender authentication: VPN implements authentication to avoid non-authorized users to access private resources;
- Message integrity: VPN packets include a mechanism to identify modifications of message content as packets travel from source to destination.

To implement a secure VPN it is possible to use other commonly used protocols, such as:

- Internet Protocol Security (IPsec, presented in the next subsection) was developed by the Internet Engineering Task Force (IETF) and proposed for all standards-compliant implementations of IPv6. However, RFC 6434 made this feature only a recommendation. Today, IPsec is broadly used in modern IPv4 networks and to implement Layer 2 Tunneling Protocol. IPsec is designed to support the following security goals: availability, integrity, and confidentiality. Practically speaking, IPsec encrypts the original IP packet and encapsulates it inside an IPsec packet at the beginning of the encrypted tunnel, and then it de-encapsulates and decrypts the packet at the end of the tunnel in order to forward it to its final destination.
- Transport Layer Security (SSL/TLS, described in the previous sub-section) provides security at the transport layer. For this reason, TLS can be either used to “tunnel“ the traffic of an entire network or to secure an individual connection. VPN is commonly implemented using SSL/TLS in order to provide remote access to private facilities. An SSL VPN can be used to connect in scenario where IPsec cannot successfully operate due to Network Address Translation and firewall rules.

7.3.4 IPSec

Internet Protocol Security (IPsec) is a protocol to enable secure encrypted communication between two hosts on an IP network. It represents a security enhancement of the IP protocol. IPsec supports different protocols for authentication and exchange of cryptographic keys, at the beginning and during each session.

It is possible to use IPsec in different scenarios: host-to-host (between to hosts), network-to-network (between two security gateways), or network-to-host (between a gateway and a host). For example, it is possible to use IPsec for implementing a VPN.

Among the services offered by IPsec we can mention: data-origin and network-level peer authentication, data integrity and confidentiality, replay protection.

IPSec was designed as an IPv4 enhancement operating at the layer 3 OSI or TCP/IP protocol model, to provide an end-to-end security scheme. Indeed, IPsec is used to automatically secure applications at the IP layer. This represents a different approach with respect to most

of the other Internet security approaches which are widespread, which operate above layer 3. As an example, we already introduced the Transport Layer Security (TLS) protocol, that operates at the Transport Layer. Another example is the Secure Shell (SSH), that operates at the Application layer.

The different functionalities of IPsec are performed by the following components:

- Data integrity and data origin authentication are provided by Authentication Headers (AH)
- Confidentiality, data integrity and data origin authentication are provided by Encapsulating Security Payloads (ESP)
- The algorithms necessary to implement ESP and/or AH operations are provided by the Security Associations (SA)

Both IPsec AH and ESP can use to connect two hosts or two gateways, in the so-called “host-to-host” transport mode and “network tunnelling” mode, respectively.

When operating in transport mode, routing is enabled in standard IPv4 network. For this reason, the IP header is not encrypted, but only the payload of the datagram. In case the authentication header is employed, IPsec cannot traverse a NAT procedure, since that would modify the corresponding hash value. Indeed, IPsec always secure transport ports and application layer, therefore it is not possible to modify those in anyway, e.g. modifying port numbers. If NAT is employed, IPsec messages must be encapsulated following the indication of RFCs related to NAT-T.

On the opposite with respect to the transport mode, in the tunnel mode the original IP datagram is encapsulated in a new datagram with a new IP header. This allows to encrypt and authenticate the entire IP datagram. As the name recalls, this solution is used to enable network-to-network secure communication or enable secure remote user access to company intranets. When operating in tunnel mode, IPsec does not have problems related to NAT traversal.

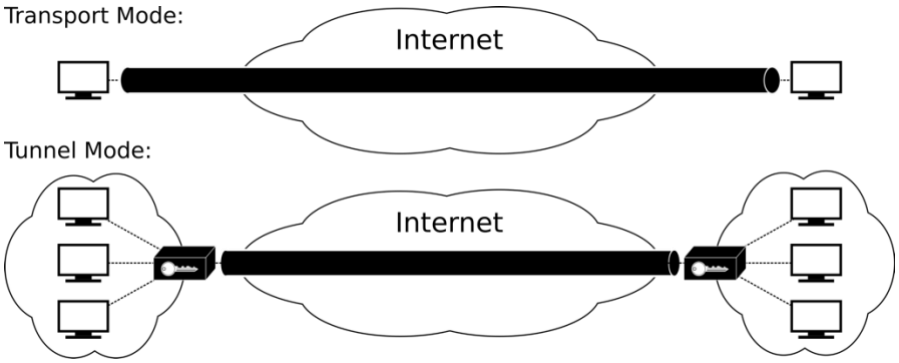


Figure 7-4: IPSec modes of operation (Wikipedia)

7.4 Securing Virtualized Environments

With the advent of virtualization in computing and in networking, it became necessary to properly secure also those specific environments. On the other side, virtualization technologies can be exploited for increasing system and network security. The next subsections briefly illustrate some basic concepts related to virtualization and their associated risks and then present some effective strategies to use such concepts in the framework of cybersecurity.

7.4.1 Virtualization Architectures and Associated Risks

Virtualization is commonly achieved by means of a hypervisor. The purpose of the hypervisor is to act as a mediator between the host operating system and the guest operating system, enabling/controlling access to both physical and logical resources, such as network cards, storage, and other peripherals.

A hypervisor can be of two types: Type 1 and Type 2. Type 1 hypervisors, or bare-metal hypervisors, are operating systems themselves, but dedicated to the single task of managing hosted virtual machines. A type 2 hypervisor instead is an application that runs on the top of a host operating system. Type 2 hypervisors are probably most known, including VMware Workstation, Oracle VirtualBox, etc.

Virtualization technologies also have risks. Most typical risks in virtualized computational environments include:

- VM Sprawl: out-of-control creation of VMs outside security control. To counteract this, every hypervisor implements authentication tools to limit the number of people allowed to build and activate VMs.
- VM Escape: it takes place when a user finds a way to break out (escape) the VM and somehow reach the hosting machine. VM escape protection modules can be deployed to reduce this risk.

7.4.2 Using Virtualization for Security

Software Defined Networking extend the above concept to the networking infrastructure. Indeed, SDN provides separation between data and control planes, in order to enable programmability and reconfigurability of network infrastructures. In the SDN world, security needs to be applied to the SDN architecture itself, as well as delivered as a service.

The former requires to secure all the three layers defining the SDN architecture, which are typically indicated as application layer, controller layer and physical infrastructure layer.

Of course, the SDN controller represents the most important module of the architecture to secure. Indeed, the SDN controller is responsible for enabling the network to correct operate, therefore a successful attack on it might completely bring the network down. The main approach used to limit those attacks is to implement role-based authentication, in order to enable only a limited number of people to have access to the SDN controller.

In general, the SDN controller should run on a trusted platform. This would enable it to act as a security tool by itself (or at least to contribute to the overall system security) by blocking or detecting requests that might be non-valid or insecure.

To reduce the network attack surface, micro-segmentation might represent a useful tool. Micro-segmentation enables to create “zones” within the networking infrastructure in order to isolate the corresponding workloads. In this way, it becomes possible by system administrators to implement policies to reduce the traffic among such segments of the networks and apply a Zero Trust approach.

The reader should note that the above represent specific issues and solutions related to the SDN scenario. However, as in the general case, it is still necessary for a secure SDN network

to (i) protect data confidentiality; (ii) protect system integrity, and (iii) ensure network services availability.

7.4.3 Software Defined Networking

As we discussed above, virtualization might introduce relevant security risks. On the other hand, the definition of isolated and stable environments to securely test applications, services and malicious code is possible only through virtualization.

This represents a clear advantage of applying virtualization in IT security, which includes also the possibility for virtualization to contribute to system and data availability by maintaining and facilitating the deployment of multiple redundant copies of identical virtual machines.

Some relevant scenarios of virtualization applied to security include the following:

- Patch compatibility: a virtual environment can provide a complete test environment that mirrors the production environment. This allows to avoid unintentional issues when applying patches to production machines.
- Increased availability of hosts and services: by using virtualization it is possible to increase the availability and elasticity of the architecture. Indeed, via virtualization it becomes easily and effective to duplicate hosts in the virtual environment almost in real time. The same holds for the associated resources and scaling of services.
- Virtual testing: security expert might test systems' security in virtual environments to facilitate security control before deploying those systems in production.
- Sandboxing: virtualization allows isolation of virtual machines. Therefore, it becomes possible to easily and affordably define a closed environment for testing of potentially dangerous applications without affecting the production system.

7.5 Referencing

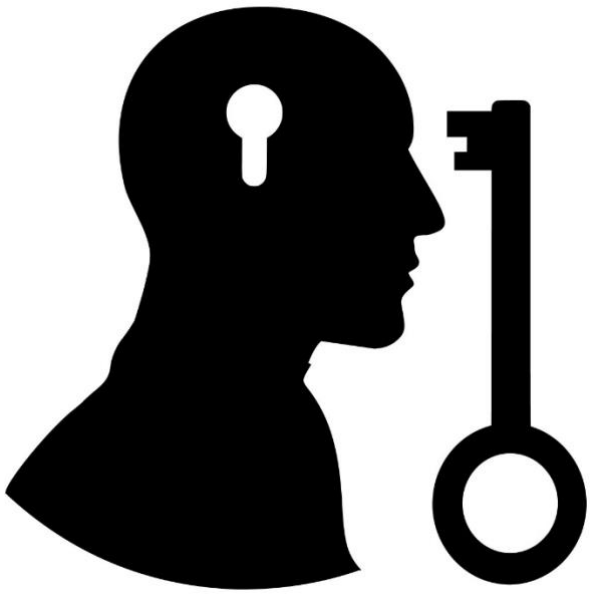
[29] Jernigan, M. M. S. (2018). *Mike Meyers' CompTIA Security Certification Guide*, Mc. Graw-Hill.

[30] https://www.uml-diagrams.org/network-architecture-diagrams.html

[31] https://www.cloudflare.com/it-it/learning-resources/tls-1-3/

8 Security Operations: Login, Monitoring & Access Control

Author(s): Maria Papaioannou
Filippos Pelekoudas Oikonomou
Georgios Mantas
Claudia Barbosa
Jonathan Rodriguez



8.1 Foundational Security Operations Concepts

8.1.1 Foundational Security Operations Concepts

Security operations are those practices that are aiming to detect, assess, respond, monitor, and prevent cybersecurity threats and incidents.

The main security operations include:

The Logging	The Monitoring	The Access Control
• includes the recording of the events or statistics with the purpose to provide information about system use and performance	• is the continual checking, supervising, critically observing or determining the status in order to identify change from the performance level required or expected	• is the process of granting or denying specific requests to: a) obtain and use information and related information processing services; and b) enter specific physical facilities (e.g., federal buildings, military establishments, border crossing entrances)

Institutions may set up this monitoring and defending capability in a facility dedicated to security operations called a **security operations center**, or SOC.

8.1.2 Security Audit Terminology

In this chapter, the discussion focuses on the collection, storage, and analysis of data about IT security. Initially, an overall look is presented about the security auditing architecture and it is discussed how this architecture associates with the activity of intrusion detection. After that, the different aspects of audit trails, also known as audit logs, are discussed. Next, an analysis of audit data is presented.

We start the discussion about security auditing by observing the components that make up a security audit architecture. In the beginning, a model, showing security auditing in its broader context, is examined. Next, a functional breakdown of security auditing is presented. ITU-T

Recommendation X.816 defines a model that describes the components of the security auditing function and their association with security alarms. Figure 8-1 shows the model. The key components are the following:

- **Event discriminator:** This logic is designed into the system software that checks the activity of the system and identifies security-related incidents that it has been programmed to detect.
- **Audit recorder**: For every identified incident, the event discriminator sends the data to an audit recorder. The model shows that this transmission can be virtualized as a message. The audit can also be accomplished by storing the information of the event in a shared memory area.
- **Alarm processor**: A number of the incidents identified by the event discriminator are evaluated as alarm events. In the case of such an event, an alarm is sent to an alarm processor. The alarm processor performs a specific action based on the alarm. This action is considered an auditable event and so the relevant information is sent to the audit recorder.
- **Security audit trail:** The audit recorder generates a formatted log of each event and saves it in the security audit trail.
- **Audit analyzer**: The audit analyzer utilizes the security audit trail. If a pattern of activity is detected, the analyzer may generate a new auditable event that is transmitted to the audit recorder and may issue an alarm.
- **Audit archiver**: This constitutes a software module whose purpose is to periodically obtain records from the audit trail. Using the records, the audit archiver constructs a permanent archive of auditable events.
- **Archives:** Security-related events, that occurs on this system, are permanently stored in the audit archives.
- **Audit provider:** The audit provider serves as an application and/or a user interface (UI) associated with the audit trail.
- **Audit trail examiner:** The audit trail examiner can be either an application or user. The examiner inspects the audit trail and the audit archives for computer forensic purposes, historical trends, and for other analysis.

- **Security reports:** Human-readable security reports are created by the audit trail examiner.

This model depicts the connection between the audit functions and the alarm functions. The audit function generates a record of events that the security administrator considers to be security related. A part of these events could really be security violations or suspected security violations. Such events are provided to an intrusion detection or a firewall function in the form of alarms.

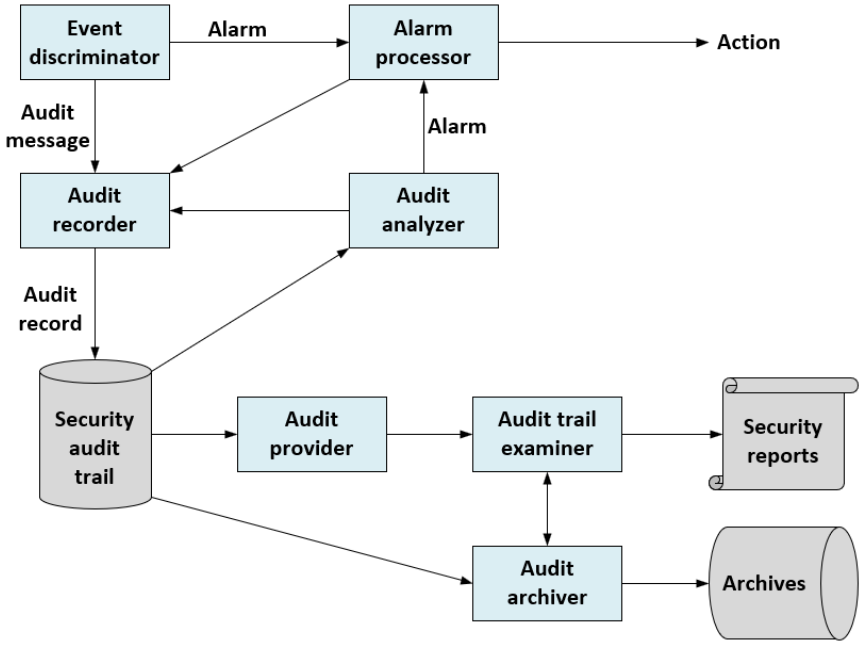


Figure 8-1: Security Audit and Alarms Model
https://csrc.nist.gov/glossary/term/monitoring

Analogously to the case of intrusion detection, a distributed auditing function with a centralized repository can be beneficial for distributed systems. A distributed auditing service (Figure 8-2) requires the following two supplementary logical components:

- **Audit trail collector:** A module on a centralized system that gathers audit trail records from other systems and constructs a joint audit trail.
- **Audit dispatcher**: A module on a local system that sends the audit trail logs from its local system to the centralized audit trail collector.

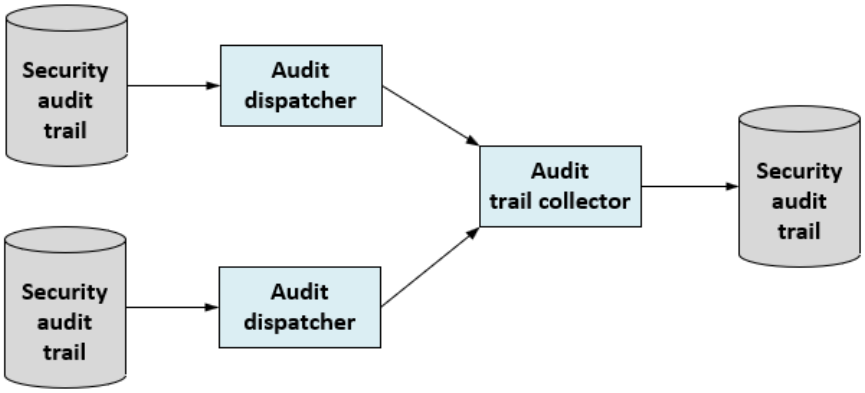


Figure 8-2: Distributed Audit Trail Model
https://csrc.nist.gov/glossary/term/monitoring

It is beneficial to look at a different analysis of the security auditing function, created as part of the Common Criteria specification. Figure 8-3 presents how the security auditing can be split into six major elements, each of which comprises of one or more particular functions:

- **Data generation:** Detects the level of auditing, counts the types of auditable events, and recognizes the minimum set of audit-related data provided. This function is also responsible to handle the conflict between security and privacy and to determine the events for which the identity of the user, correlated to an action, is incorporated in the data produced as a result of an event.
- **Event selection**: Determines if events will be included or excluded from the auditable set. This enables the system to be set up at different levels of granularity to evade the generation of an unwieldy audit trail.
- **Event storage:** Generation and maintenance of the secure audit trail. The storage function involves measures that aim to offer availability and to thwart loss of data from the audit trail.
- **Automatic response:** Implements reactions taken in case that events, that are similar to a possible security violation, are identified.
- **Audit analysis:** Implemented via automated mechanisms to investigate system activity and audit data. Its purpose is to check for potential security violations. This component recognizes the set of auditable events whose appearance or accumulated appearance points to a possible security violation. Such events are analyzed in order to ascertain if a security violation has taken place; this analysis entails the use of anomaly detection and attack heuristics methods.
- **Audit review:** Can be performed by authorized users in order to provide assistance in audit data review. The audit review element may contain a selectable review function that allows to perform searches related to a one or more criteria with logical (i.e.,

and/or) associations, sort audit data, and filter audit data before this data is reviewed. The audit review functionality may be available only to authorized users.

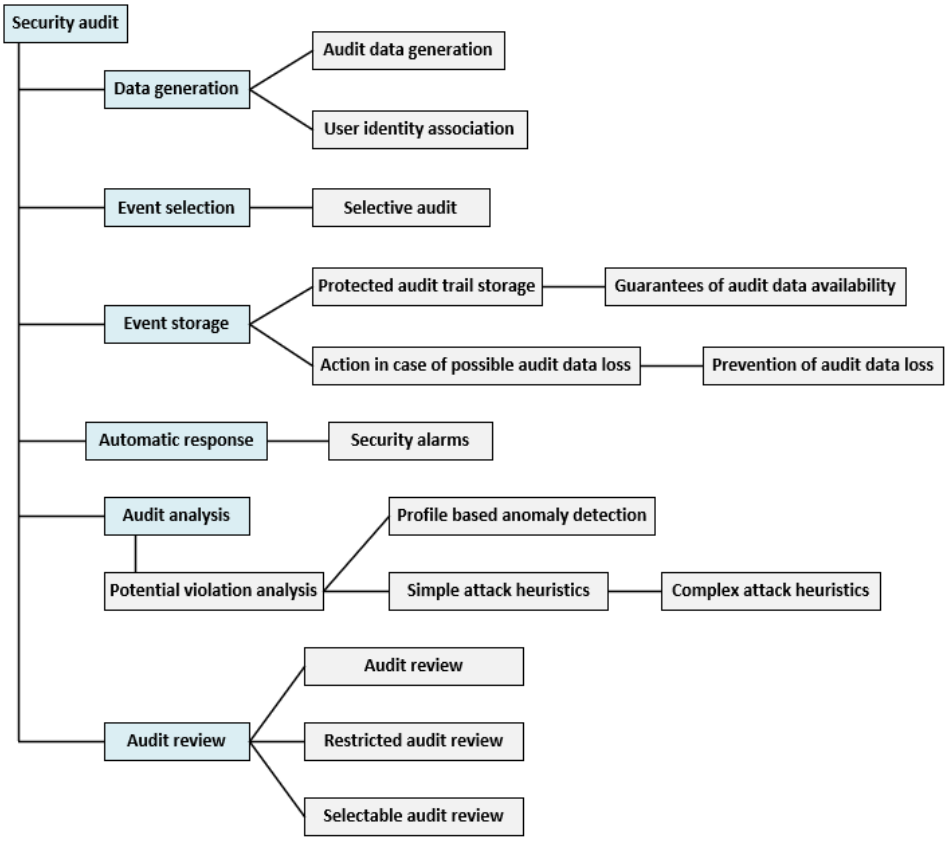


Figure 8-3: Common criteria security audit class decomposition
https://csrc.nist.gov/glossary/term/monitoring

8.1.2.1 Event Definition

Reviewing the functionality suggested by Figures 8-1 and 8-3, we can define a group of requirements for security auditing. **Event definition** constitutes the first requirement**.**

The security administrator is responsible to describe the group of events that are subject to audit.

In the next section more details are presented and below, a list, proposed by the bibliography is included:

- Addition of objects within the security-related part of the software into a subject's address space

- Deletion of objects
- Distribution or revocation of access rights or capabilities
- Changes to subject or object security attributes
- Policy checks performed by the security software as a result of a request by a subject
- The utilization of access rights to circumvent a policy check
- Use of identification and authentication functions
- Security-related actions performed by an operator and/or authorized user (e.g., suppression of a protection mechanism)
- Import/export of data from/to removable media (e.g., printed output, magnetic or optical disks, portable USB storage devices)

8.1.2.2 Event Detection

Another requirement states that the appropriate hooks should exist in the application and system software to permit **event detection.** Monitoring software is required to be integrated in the system and in proper places to record relevant activity. After that, an **event recording** function is required, and it also has to account for a secure storage that possesses resistance against tampering or deletion. **Event and audit trail analysis software, tools, and interfaces** may assist in the analysis of collected data as well as the investigation of data trends and anomalies.

One additional requirement refers to the **security of the auditing function.** Not just the audit trail, but it is necessary that all of the auditing software and intermediate storage are protected from tampering or bypass. Finally, the effect of the auditing system on functionality should be **minimal.**

8.1.2.3 Protecting Audit Trail Data

RFC 2196 (Site Security Handbook, 1997) contains three alternatives for the storage of audit records:

- Read/write file on a host

- Write-once/read-many device (e.g., CD-ROM or DVD-ROM)
- Write-only device (e.g., a line printer)

File system logging is the least resource intensive and is comparatively simple to set up. Logs can be accessed instantly, and this is helpful in case of countering an attack in progress. Nevertheless, this method entails a high degree of vulnerability. If an attacker manages to obtain privileged access to a system, then the audit trail becomes susceptible to modification or deletion attempts.

A DVD-ROM or similar storage method may be a lot more secure but it is also more inconvenient. A stable stock of recordable media is required. Furthermore, access to the logs may involve a lot of delays and the logs may not be immediately available.

Printed records offer a paper trail. However, they are not practical to use in capturing meticulous audit data on large or networked systems. RFC 2196 proposes that the paper log can be practical in case that a permanent, immediately available record is needed even if a system crash is ongoing.

The protection of the audit trail refers to matters of both integrity and confidentiality. Integrity is especially important since an adversary may attempt to erase proof of the intrusion by modifying the audit trail. In the case of file system logging, the best way to provide integrity may be the use of the digital signature. Write-once devices, such as DVD-ROM or paper, inherently include integrity. Another measure to ensure integrity is strong access control.

Confidentiality is essential in case that the audit trail involves sensitive user information that must not be revealed to all users. Examples of such information are the changes in a salary or the pay grade status. Strong access control assists in this case. A valid measure relates to symmetric encryption (e.g., using AES [Advanced Encryption Standard] or triple DES [Data Encryption Standard]). The secret key must be secured and only accessible by the audit trail software and subsequently by the audit analysis software.

Note that integrity and confidentiality mechanisms safeguard audit trail data while it is stored locally and also while it is transferred to a central repository.

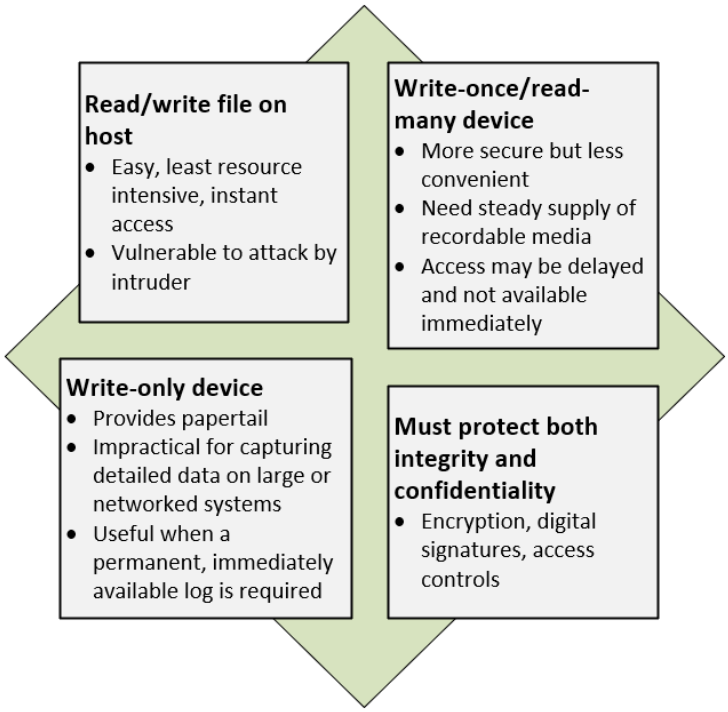


Figure 8-4: Protecting Audit Trail Data
https://csrc.nist.gov/glossary/term/monitoring

8.2 Authentication and Authorization

8.2.1 User Authentication

NIST SP 800-63-3 describes digital user authentication as the establishment of confidence between two entities in the user identities which are introduced in electronic form to an information system. Afterwards, systems utilizing the authenticated identity may decide if the authenticated entity is permitted to perform certain operations. In numerous cases, the authentication and subsequent authorization may happen locally (e.g., across a local area network).

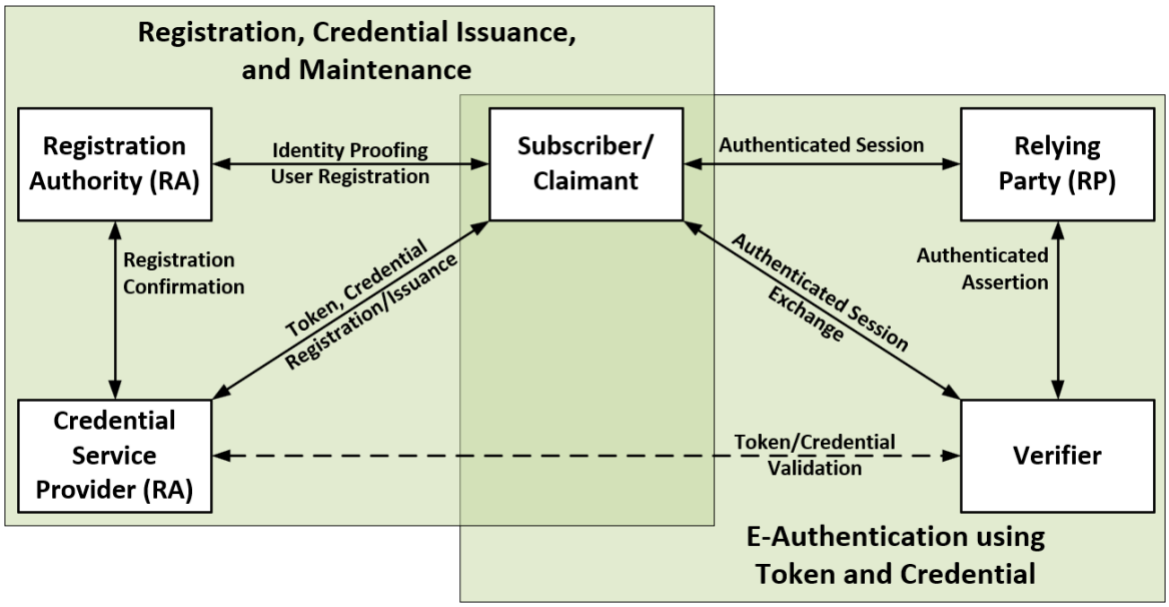


Figure 8-5: E-Authentication based on NIST SP 800-63-2
https://csrc.nist.gov/glossary/term/monitoring

NIST SP 800-63-3 develops a general model for user authentication that contains several entities and procedures. This model is discussed using the Figure 8-5 as reference.

The first requirement to perform user authentication relates to the user being registered with the system. A typical process for registration follows. An applicant makes a request to a registration authority (RA) to subscribe in a credential service provider (CSP). In this case, the RA is a trusted entity that generates and vouches for an applicant ’s identity to a CSP. After that, the CSP engages in an exchange with the subscriber. Based on the technicalities of the overall authentication system, the CSP generates some sort of electronic credential to the subscriber. The credential consists of a data structure that authoritatively associates an identity and additional attributes with a token of a subscriber. The credential can be verified when provided to the verifier in case of an authentication transaction. The token may be an encryption key or an encrypted password from which the subscriber can be identified. The token could be generated by the CSP or directly by the subscriber or supplied by a third party. The token and credential can be used in ensuing authentication events.

After a user registers as a subscriber, the actual authentication process may occur between the subscriber and one or multiple systems which perform authentication and authorization. The party that requires authentication, is named claimant and the party which validates that

identity is named verifier. If a claimant successfully shows possession and control of a token to a verifier by using an authentication protocol, the verifier can ensure that the claimant is the subscriber mentioned in the respective credential. The verifier transmits an assertion concerning the identity of the subscriber to the relying party (RP). The assertion contains identity information of a subscriber, such as an identifier assigned at registration, the name of the subscriber or other verified subscriber attributes from the registration process. The RP can utilize the authenticated information, that the verifier sent, in order to make decisions regarding access control or authorization.

The general means of authenticating a user's identity are four and can be utilized alone or in combination:

Something the individual knows

- Instances involve a personal identification number (PIN), a password or answers to a preconfigured set of questions.

Something the individual possesses

- Instances involve smart cards, electronic keycards and physical keys. This type of authenticator is also named as *token*.

Something the individual is (static biometrics)

- Instances involve recognition using fingerprint, face and retina biometrics.

Something the individual does (dynamic biometrics)

- Instances involve recognition using biometrics like voice pattern, handwriting characteristics and typing rhythm.

All these methods can provide secure user authentication. Nevertheless, each method has drawbacks. It is possible that an attacker steals or guesses a password. In a similar way, an adversary could potentially steal or forge a token. Also, a user could fail to remember a password or lose a token. Moreover, there is a considerable administrative overhead in order to manage password and token information on systems and to ensure the protection of such information on systems. In the case of biometric authenticators, existing problems involve handling false positives and false negatives, cost, user acceptance and convenience.

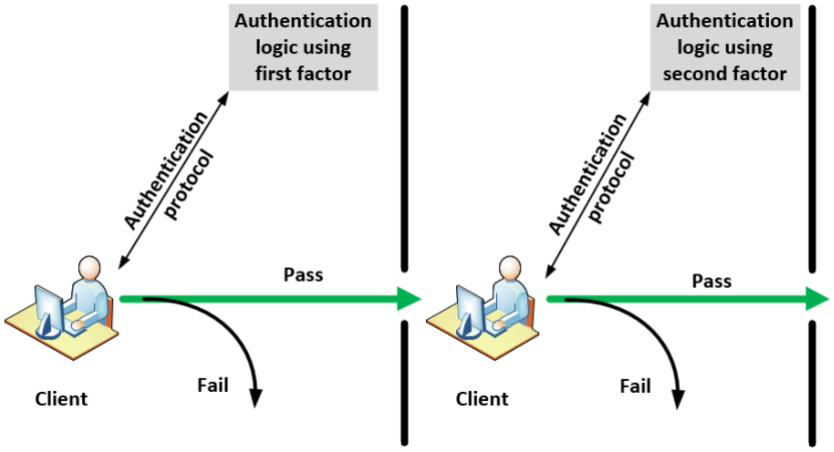


Figure 8-6: Multifactor Authentication
https://csrc.nist.gov/glossary/term/monitoring

Multifactor authentication refers to the development and integration of more than one of the authentication means including all four general categories in combination (see Figure 8-5). The strength of authentication mechanisms is depending on the different authentication means that have been incorporated by the authentication system. Usually, two-factor authentication is considered more secure and rigorous than one-factor authentication, and so on.

8.2.1.1 Event Definition

The password system constitutes a widely way of defending against attackers. Almost all multi-user systems, Web-based e-commerce sites, network-based servers, and other such services oblige the user to insert not only a name or identifier (ID) but additionally a password. The system performs a comparison of the provided password to a previously saved password that corresponds to that user ID and is stored in a password file of the system. The purpose of the password is to assist in the authentication of the ID of the individual that attempts to log on to the system. In turn, the ID enhances security in the ways mentioned below:

- The ID dictates if the user receives authorization to access a system. In some cases, system access is only permitted to those who already possess an ID that is registered on the system.

The ID designates the user privileges. A small amount of users may possess supervisory or “superuser” status that allows them to read files and perform operations which the operating system specifically protects. Some systems provide guest or anonymous accounts. The users of such accounts are given more reduced privileges than others.

- The ID is employed in discretionary access control. For instance, it is possible for other users to read files owned by another user if the latter one grants permission to the former users by listing their IDs.

8.2.1.1.1 Password vulnerabilities

The major forms of attacks against password-based authentication are outlined and for each form of attack, a countermeasure strategy is briefly presented in this subsection. The following attack strategies can be specified:

- **Electronic monitoring:** A password, that is transmitted through a network to log on to a remote system, is simultaneously exposed to eavesdropping. This problem cannot be solved using simple encryption, since the encrypted password is, in reality, the password and an attacker can inspect and reuse it.
- **Exploiting multiple password use:** In the case that multiple network devices have the same or a similar password assigned to a specific user, attacks can become more potent or harmful. One countermeasure is the establishment of a policy that prohibits the same or similar password on specific network devices.
- **Popular password attack:** Another method of attack entails the use of a popular password against a large number of user IDs. One amongst the trends of the users is the selection of a password that is simple to memorize. However, this means that the password becomes easy to guess. Countermeasures involve the establishment of policies to limit the selection of frequently used passwords by users and the examination of the IP addresses about authentication requests and client cookies for submission patterns.
- **Workstation hijacking:** The adversary awaits until a logged-in workstation is unattended. A standard countermeasure involves the automatic logout of the

workstation if there is inactivity for a specified period of time. Intrusion detection mechanisms can also be employed to identify changes in user behavior.

- **Password guessing against single user:** The adversary tries to obtain information regarding the account holder and system password policies and exploits that information to infer the password
- **Exploiting user mistakes:** If a password is provided by the system, then it is likely to write it down since it may be challenging to memorize. This means that an adversary could possibly read the written password if such a circumstance arises. Additionally, a user may intentionally distribute a password in order to allow another person to exchange files, for instance. A lot of times, attackers succeed in acquiring passwords by employing social engineering techniques which deceive the user or account manager into disclosing a password. Countermeasures involve intrusion detection, user training, and simpler passwords in conjunction with an additional authentication mechanism.
- **Specific account attack:** The adversary focuses on a particular account and sends password guesses until the correct password is found. In this case, a standard countermeasure can be an account lockout mechanism, that refuses access to the account if a specific number of failed login attempts has been surpassed. In common practice, the max number corresponds to five failed attempts.
- **Offline dictionary attack:** Although strong access controls are typically employed for the protection of the system's password file, the adversary may gain access to the system password file and performs a comparison between the password hashes and the hashes of typically used passwords. If there is a match, the adversary may obtain access using that ID/password combination. Countermeasures involve intrusion detection mechanisms to recognize a compromise, controls to inhibit unauthorized access to the password file, and quick reissuance of passwords in the case that the password file becomes compromised.

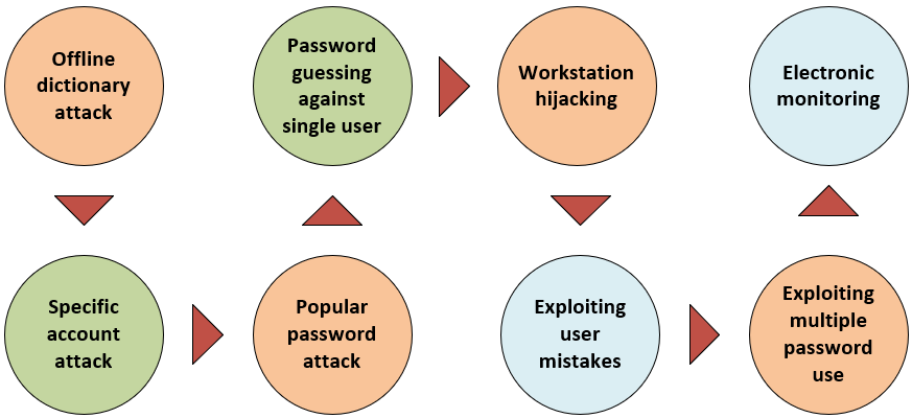


Figure 8-7: Password Vulnerabilities
https://csrc.nist.gov/glossary/term/monitoring

8.2.1.2 Token-Based Authentication

The tokens are objects that are in possession of a user for the purpose of user authentication. In this section, two categories of tokens, that are widely utilized, are examined; these tokens are similar to bank cards both in size and appearance.

8.2.1.2.1 Memory cards

Memory cards are capable of storing but they cannot perform processing on data such as the bank card with a magnetic stripe on its back. In the magnetic stripe, only a simple security code can be stored, and an inexpensive card reader can read (and unfortunately reprogram) this security code. Also, an internal electronic memory may be included in some memory cards.

Memory cards can be utilized alone for physical access. In the case of computer user authentication, memory cards are normally utilized in conjunction with some type of personal identification number (PIN) or password. An automatic teller machine (ATM) is a classic application. The memory card, if it is employed together with a password or PIN, offers considerably more security compared to just a password. An adversary needs to acquire the card (or be able to duplicate it) and in addition, know the PIN. Among the potential drawbacks, NIST SP 800-12 notes the following:

- **Requires special reader**

- **Case of token loss**
- **User dissatisfaction**, as they should care an extra card only for authentication purposes.

8.2.1.2.2 Smart tokens

Various devices can be considered as smart tokens. They can be distributed along three non-mutually exclusive dimensions:

- **User interface**: Manual interfaces contain a display for human/token interaction and a keypad.
- **Physical characteristics**: The smart tokens contain an integrated microprocessor. Smart tokens can be the bank cards, as well as resemble keys, calculators, or other small portable objects.
- Electronic interface: A smart card or other token needs an electronic interface to interoperate with a compatible reader/writer.

It is possible that a card possesses one or both of the types of interface mentioned below:

- Contact: A contact smart card requires to be placed into a smart card reader. The reader needs to connect directly to a conductive contact plate that exists on the surface of the card (typically gold plated). The transfer of commands, data, and card status occurs over these physical contact points.
- Contactless: A contactless card needs only to be in close proximity to a reader. Both the card and the reader possess an embedded antenna, and the two use radio frequencies for communication.

➢ **Authentication protocol**: A smart token has the purpose of acting a means for user authentication. The authentication protocols employed with smart tokens can be classified into three types:

Static	✓In a static protocol, the user is initially authenticated to the token and then the token is used to authenticate the user to the computer.
Dynamic password generator	• In this case, a unique password is created periodically (e.g., every minute). This password is used into the computer system as an authentication means, either manually by the user or electronically via the token.
Challenge-response	• In this case, the computer system creates a challenge. This can be a random string of numbers. The smart token constructs a response according to the content of the challenge.

8.2.1.2.3 Smart cards

For user authentication, the smart card is the most fundamental type of smart token. The smart card appears as a credit card, has an electronic interface, and may utilize any type of the protocols just mentioned. In the rest of this section, smart cards are discussed.

A smart card includes inside it an entire microprocessor, with its processor, memory, and I/O ports. Certain versions integrate a special co-processing unit for cryptographic operation to assist in the encoding and decoding of messages or in the creation of digital signatures to verify the transferred data. In some cases, the I/O ports of certain cards can be directly accessed by a compatible reader through exposed electrical contacts. In other cards, an embedded antenna is employed to ensure wireless communication with the card reader.

An ordinary smart card contains three types of memory. The Read-only memory (ROM) holds data that are not modified during the card's life. Examples are the card number and the cardholder's name. The Electrically erasable programmable ROM (EEPROM) stores application information and programs, such as the protocols which the card can operate. It

also stores information which may change with time. For instance, a telephone card has an EEPROM which stores the remaining talk time. The Random Access Memory (RAM) saves temporary data that are created during the execution of applications.

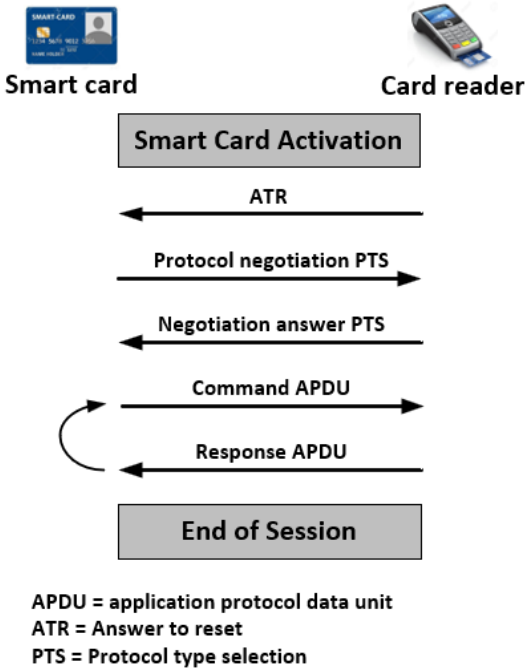


Figure 8-8: Smart Card/Reader Exchange
https://csrc.nist.gov/glossary/term/monitoring

Figure 8-8 depicts the typical way that a smart card and a reader or computer system communicate between each other. Each time the card is inserted into a reader, the reader performs a reset in order to set up parameters like the clock value. After the end of the reset function, the card sends an answer to reset (ATR) message. This message determines the parameters and the protocols that the card can execute and the other operations it can perform. It is possible for the terminal to modify the utilized protocol and other parameters by transmitting a protocol type selection (PTS) command. The card transmits a PTS response and confirms the protocols and parameters that will be used. Then, the card and terminal can start using the protocol to execute the desired application.

8.2.1.3 Biometric Authentication

A biometric authentication system receives the physical characteristics of an individual for authentication purposes. These characteristics can be static or dynamic. Static physical characteristics include facial, iris and retinal patterns, hand geometry, and fingerprints, while dynamic characteristics may be the individual´s signature and voiceprint. In reality, biometrics relates a lot to pattern recognition.

It is considered as secure and accurate authentication method given the fact that physical characteristics are unique for every person. However, biometric authentication is much more expensive and technically complicated compared to password-based and token authentication. On top of that, although it is already employed in several applications (e.g., smartphone authentication), biometrics have not yet reached the maturity stage in order to serve as a standard tool for user authentication to computer systems.

Various types of physical characteristics are either used or studied to be used for user authentication. The following characteristics consist the most common:

- **Fingerprints**: Fingerprints are a means of identification that has been employed for centuries, and the process has been automated and employed widely in the law enforcement sector. A fingerprint entails the pattern of ridges and furrows that can be found on the surface of the fingertip. It is considered that fingerprints are unique across the whole human population. Practically, automated fingerprint recognition and matching system export certain features from the fingerprint and store these features as a numerical surrogate representing the full fingerprint pattern.
- **Voice**: While the signature style of an individual shows the unique physical attributes of the writer and the developed writing habit, voice patterns are closely associated to the physical and anatomical traits of the speaker. However, multiple samples over time from the same speaker may present variations between them. Subsequently, this raises the difficulty of the task of biometric recognition.
- **Retinal pattern**: The retinal pattern is the pattern of veins that exist beneath the retinal surface. The retinal pattern is unique and thus appropriate for use in identification. A retinal biometric system captures a digital image of the retinal pattern by transmitting a low-intensity beam of visual or infrared light into the eye.

- **Facial characteristics**: Facial characteristics are used most typically in human-to-human identification and therefore, they are commonly employed in identification by computer. The typical technique requires to describe characteristics according to relative location and shape of key facial features, such as eyes, eyebrows, nose, lips, and chin shape. Another approach relies on the use of an infrared camera to create a face thermogram that corresponds with the underlying vascular system in the human face.
- **Iris**: The detailed structure of the iris constitutes another unique physical characteristic.
- **Hand geometry**: Hand geometry systems detect features of the hand, such as the shape, and widths and lengths of fingers.
- **Signature**: The style of handwriting of each individual is unique. This is especially apparent in the signature, that is a written sequence and performed frequently. Nevertheless, it is normal that multiple signature samples of the same individual may present differences between them. This results in the increase of the difficulty of designing a computer equivalent of the signature that can be matched to future samples.

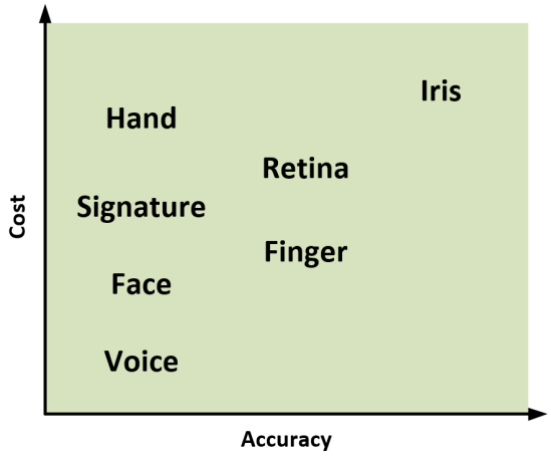


Figure 8-9: Cost Versus Accuracy of Various Biometric Characteristics in User Authentication Schemes
https://csrc.nist.gov/glossary/term/monitoring

Figure 8-9 provides a rough depiction of the relative cost and accuracy of the biometric measures. The notion of accuracy is not applicable to user authentication schemes utilizing smart cards or passwords. For instance, when a user inserts a password, there are two

possible results; either perfect match with the password relating to that user or not. In contrast, a system, that uses biometrics parameters, requires to measure how closely a given biometric characteristic relates to a saved characteristic. Before discussing further about the notion of biometric accuracy, it is important to understand how biometric systems generally work.

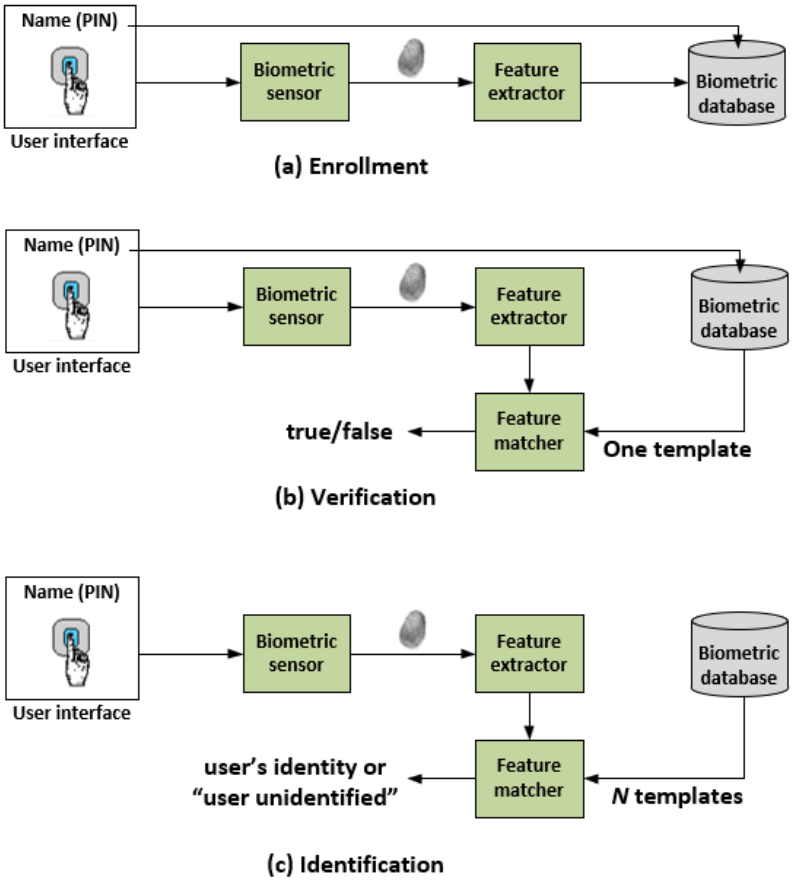


Figure 8-10: Generic Biometric System
https://csrc.nist.gov/glossary/term/monitoring

Figure 8-10 depicts how a biometric system operates. Each individual must first **enroll** himself in the system in order to be considered as a legitimate entity. This is similar to how a user is assigned a password. In the case of the biometric system, the user gives a name and, usually, a PIN or password to the system. At the same time, the biometric system grasps user´s biometric characteristic. Afterwards, the authentication system converts the input to a digital form and exports it as a collection of features that are saved as a number or set of numbers symbolizing this unique biometric characteristic; this collection of numbers constitutes the

user's template. The enrollment of the user in the system is now finished and the system retains a set of values for this user. These values consist of a name (ID), perhaps a PIN or password, and the biometric value.

8.2.2 Authorization – Access Control

Broadly speaking, all of computer security revolves around the term of access control. This chapter elaborates on a narrower and more precise concept of access control: Access control implements a security policy that determines who or what (e.g., in the case of a process) may be permitted to access each specific system resource, and what type of access is allowed in each case.

8.2.2.1 Access Control Policies

An access control policy can be implemented in an authorization database. The access control policy decides which are the permitted types of access, under what circumstances, and who can obtain access. In general, the categories of the access control policies are the following:

- **Discretionary access control** (DAC): Controls access according to the identity of the requestor and to access rules (authorizations) determining which actions requestors are (or are not) permitted to take. This policy is referred as *discretionary* because an entity might have access rights that allows it, by its own volition, to let another entity to gain access to a resource.
- **Mandatory access control** (MAC): In this case, access is regulated by comparing security labels (an indication of how sensitive or critical a system resource is) with security clearances (an indication about whether system entities are allowed to obtain access to specific resources). This policy is named as *mandatory* since an entity, that possesses permission to access a resource, may not, just by its own volition, allow other entities to access that resource.
- **Role-based access control (**RBAC): In this case, access is regulated according to the roles of the users within the system and according to the rules which state which types of accesses are permitted to users in given roles.

- **Attribute-based access control** (ABAC): In this case, access is controlled based on the user attributes, the resource to which access is requested, and current environmental conditions.

The four policies mentioned above are not mutually exclusive. It is possible for an access control mechanism to use two or even three of these policies in order to secure different classes of system resources.

8.2.2.2 Basic Elements – Subject, Object, Access Rights

The basic elements of access control are: subject, object, and access right.

A **subject** constitutes an entity that has the capability to access objects. In general, the concept of the subject is equivalent to that of the process. Any application or user indeed obtains access to an object through a process that symbolizes that application or user. The process assumes the user attributes, including the access rights of the user.

Typically, a subject is accountable for their performed actions, and it is possible to utilize an audit trail to log the relation between a subject and security related actions that the subject carried out on an object.

Three classes of subject are commonly defined in basic access control systems, and each class has different access rights:

- **Owner**
- **Group**
- **World**

An **object** is defined as a resource to which access is regulated. Generally, an object is utilized to hold and/or receive information. The quantity and different categories of objects that require protection by an access control system is reliant on two parameters. The first parameter is the environment where access control performs its operations. The second parameter relates to the desired tradeoff between security on the one side and complexity, processing burden, and usability on the other side.

An **access right** defines how a subject may gain access to an object. Access rights may involve the following:

- **Read**
- **Write**
- **Execute**
- **Delete**
- **Create**
- **Search**

8.3 Identity, Credential, and Access Management (ICAM)

8.3.1 Identity Management

Identity management refers to the process of assigning attributes to a digital identity and associating that digital identity with an individual or NPE. The aim is the establishment of a trustworthy digital identity that does not rely on any specific application or context. The traditional approach, regarding access control for applications and programs is the generation of a digital equivalent of an identity for the particular use of the application or program. Consequently, the protection and the maintenance of the identity itself is considered as unimportant compared to the mission of the application. Moreover, there is noticeable overlap in the effort to create these application-specific identities.

In contrast to the accounts utilized to logon to networks, systems, or applications, enterprise identity records are not associated with job title, job duties, location, or whether access is needed to a specific system. These items may develop into attributes linked to an enterprise identity record and may also become part of what uniquely describes an individual in a particular application. Access control decisions will be according to the context and related attributes of a user—not only the identity of a user. The concept of an enterprise identity refers to the fact that individuals will possible a single digital representation of themselves. This digital representation can be used across departments and agencies for various purposes, as well as access control.

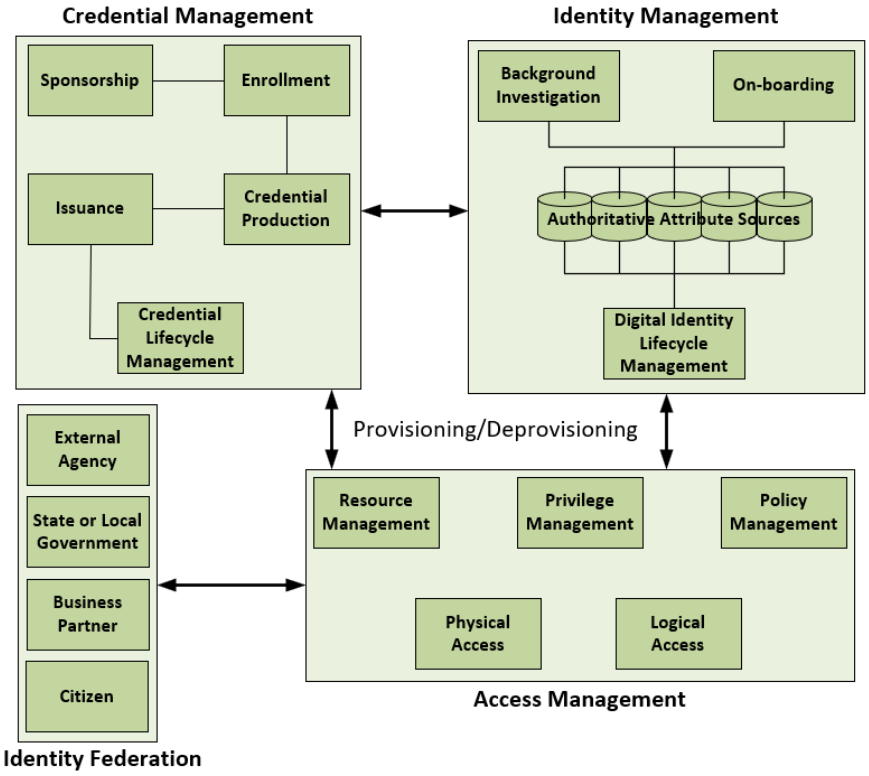


Figure 8-11 Identity, Credential and Access Management
https://csrc.nist.gov/glossary/term/monitoring

Figure 8-11 illustrates the key functions used in identity management. Normally, the establishment of a digital identity start with gathering the identity information as part of an on-boarding process. A digital identity often consists of a collection of attributes. If those attributes are aggregated, they can uniquely identify a user within a system or an enterprise. In some cases, an agency may also want to perform a background investigation in order that the individual, that is symbolized by the digital identity, can be trusted. The attributes of an individual may be saved in several authoritative sources of an agency and combined to create an enterprise view of the digital identity. This digital identity can then be offered to various applications to assist in physical and logical access (part of Access Management) and retracted when access is not needed anymore.

Lifecycle management constitutes the last element of identity management and it involves the following:

- Procedures, mechanisms, and policies to protect personal identity information
- Regulating access to identity data

- Techniques for providing the authoritative identity data to applications that require it
- Revocation of an enterprise identity

8.3.2 Credential Management

As stated, a credential is an object or data structure which authoritatively links an identity (and optionally, other attributes) to a token that a subscriber possesses and controls. Some instances of credentials are the smart cards, the digital certificates, and the private/public cryptographic keys. Credential management relates to how the life cycle of a credential is managed. Credential management involves the five logical elements that are mentioned below:

1. An authorized individual sponsors an entity or individual for a credential to create the necessity for the credential. For instance, a department supervisor sponsors a department employee.
2. The sponsored individual registers for the credential. Typically, the enrollment process comprises of identity proofing and the record of biometric and biographic data. In this step, the authoritative attribute data, that are retained by the identity management component, may also be incorporated.
3. A credential is generated. Based on the type of the credential, its generation may require encryption, the creation of a smartcard, the use of a digital signature, or other different functions.
4. The individual or NPE receives the credential.
5. Lastly, the maintenance of the credential during its life cycle is crucial and it may involve revocation, expiration, reissuance/replacement, personal identification number (PIN) reset, reenrollment, suspension, or reinstatement.

8.3.3 Access Management

The function of the access management component is to regulate and control how entities can access the resources. It encompasses both physical and logical access and, as an element, it can be either internal to a system or external. Access management has the purpose of

providing assurance that an individual 's identity is properly verified when this individual tries to gain access to security sensitive data, computer systems, or buildings. The credentials of those that request to gain access plus the digital identity of the requestor are used by the access control function. Generally, three support elements are required by an enterprise-wide access control facility:

- Privilege management: that deals with the establishment and maintenance of the entitlement attributes included in an access profile of an individual.
- Policy management: determines what is and is not permitted in an access transaction.
- Resource management: determines rules regarding a resource that needs access control.

8.4 Referencing

[1] Stallings, W., Brown, L., Bauer, M. D., & Bhattacharjee, A. K. (2012). *Computer security: principles and practice* (pp. 978-0). Upper Saddle River, NJ, USA: Pearson Education.

[2] Matt, B. (2002). Computer security: art and science.

[3] Easttom, C. (2019). *Computer security fundamentals*. Pearson IT Certification.

[14] Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2018). *Handbook of applied cryptography*. CRC press.

[32] https://csrc.nist.gov/glossary/term/security_concept_of_operations

9 Security Operations: Intrusion detection & Prevention

Author(s): Filippos Pelekoudas Oikonomou
Georgios Mantas
Claudia Barbosa
Jonathan Rodriguez

9.1 Firewalls

Firewalls function as protection for either a system or a network of systems, from threats that are related to network-based security, allowing access to the outside world using the Internet.

Information systems in various organizations, corporations, and government agencies, have endured a constant change. Important developments are the following:

- The development of a centralized data processing system, that supports several connected terminals, with a central mainframe.
- Local area networks (LANs) connecting internally computer devices to terminals and to the mainframe.
- Property networks, that consist of several LANs, connected computer devices, servers, and possibly several mainframes.
- Enterprise-wide network, that consists of different, topologically networks that are connected by a private wide area network (WAN).
- Internet connectivity, in which all individual networks fuse into the Internet and also have the possibility be connected by a private WAN.
- Enterprise cloud computing, that contains virtualized servers placed in one or more data centers which are able to provide internal organizational and also external Internet accessible services.

9.1.1 The Need for Firewalls

Organization has no choice in not having Internet connectivity. The organization has an immediate need of Available services and information. Additionally, users inside the organization have the demand of Internet access, and in the case, it is not provided through their LAN, they have the option to utilize wireless broadband capability originating at their computer device to an Internet service provider (ISP). Nevertheless, Internet access except it grants enhancements to the organization, it is also giving access to the outside world interact with assets inside local network. This can be proved as a threat to the organization. Although there is the option of providing each server and workstation on the premises network with

efficient security traits, for example an intrusion protection, there is a chance this is not adequate and even not cost effective in some cases.

In a network with a vast variety and number of systems, when a security flaw is detected, there is a need for every affected system to be upgraded to patch this weakness. In order to succeed a scalable configuration management is needed and also an aggressive patching to work effectively. although this is not easy, it is necessary and is possible only if it is used host-based security. Firewall is an accepted possible choice or a complementary solution to host-based security services. Firewall is added amidst the property network and the Internet to create a perimeter or a security wall, around the network and a controlled link. This perimeter aims to create a strait point in which security and auditing can be placed and to guard this private network from Internet-based attacks. An individual computer system or more systems can act as a firewall by cooperating to accomplish the firewall function.

A supplementary layer of defense is provided by the firewall, in which the internal systems are closed off from external networks.

9.1.1.1 Firewall Characteristics

The aims of the firewall design are listed as follows:

1. The traffic from inside to outside, and the other way around, has to go by the firewall. This aim cab be achieved by blocking physically every access towards the local network if it is passing first through the firewall. This can be done with different configurations.
2. Firewalls apply a local security policy that defines which gives authorization to specific traffic and allows it to pass. There are different kinds of policies and firewalls for this purpose.
3. The use of a strengthened system with a dependable operating system is implied as the firewall is immune to penetration.

9.1.2 Access Policy

Specifying suitable access policy is a dire component in the design and deployment of firewalls. The access policy enumerates which types of traffic are authorized to go via the firewall. That contains protocols, address ranges, content types and applications and content types. The information security risk assessment of the organization, ought to be responsible to develop each policy. An open specification of the traffic types, each organization has the necessity to support should develop each policy.

9.1.2.1 Filter characteristics

According to NIST a firewall access policy can use a series of characteristics to separate traffic:

- **IP Address and Protocol Values**: This trait controls access regarding addresses or port numbers of the source and destination, the course of the flow either being inward or outgoing, or other network and transport layer traits. Stateful inspection firewalls and packet filters are using this kind of separation - filtering and they use it to restrict the access to particular services.
- **Application Protocol:** It controls access regarding the data of the authorized application protocol. An application-level gateway that delivers as well as controls the information transferring for distinct application protocols use the indicated filtering. Examples are the check of Simple Mail Transfer Protocol (SMTP) e-mail for spam, or HTTP Web that requests to solely authorized websites.
- **User Identity:** It controls access on the basis of users' identity. It functions usually in relation to interior users that use a form of authentication technology, e.g., IPSec, to identify themselves.
- **Network Activity**: The characteristic handles the access regarding the considerations such as rate of requests or time of requests.

9.1.3 Capabilities and Limits

Summarizing, firewalls have the following capabilities:

1. A firewall sets an individual control point which tries to keep the unauthorized users outside the network it protects, restricts possible vulnerable services to infiltrate or flee the network, and protects against different types of routing attacks and IP spoofing. By using an individual control point makes security management more simple due to security capabilities that are combined on a sole system or even group of systems.
2. A position to monitor events that are security-related and incidents is provided by a firewall. Also, on the firewall system there can be implemented audits and alarms.
3. There are not security related Internet functions that are part of a firewall. These are a network management function that monitors or logs Internet utilization and a network address interpreter, which maps local addresses to Internet addresses.
4. IPSec can use a firewall as a platform to be set up. This is achieved with the capability of tunnel mode. There the firewall can be applied to create virtual private networks.

On the other hands Firewalls have disadvantages, which are:

1. Firewalls protect only against attacks that have not bypassed it. Internal systems can own the capability of dial-out or mobile broadband for connection to an ISP. This capability can be provided by a modem pool that is supported by an internal LAN.
2. There is a chance that the firewall will not protect against threats from within, e.g., a member of personnel which unintentionally works with an external attacker.
3. A wireless LAN which is not properly secured could be accessed from externally of the organization. The firewall, which is located internally and separates portions of an enterprise network does not have the ability to secure against wireless communications between local systems.
4. There is a chance that a device that is used outside the organization will get infected and then connected internally.

9.1.4 Type of Firewalls

Firewall can audit network traffic in different levels, beginning from low-level network packets, to the traffic inside a transport connection, up to examining application protocols information. The desired firewall access policy determines which level is appropriate. Firewall has the ability to function as a positive filter. This way, allows packets to pass that meet specific requirements. It functions also as a negative filter, by repudiating packets that meet particular criteria. The access policy for the firewall is implemented by these requirements and criteria. Which part of the packet is examined by the firewall, e.g., the pattern generated by a sequence of packets, one or more protocol headers in each packet or the payload of each packet, is depended on the type of the firewall.

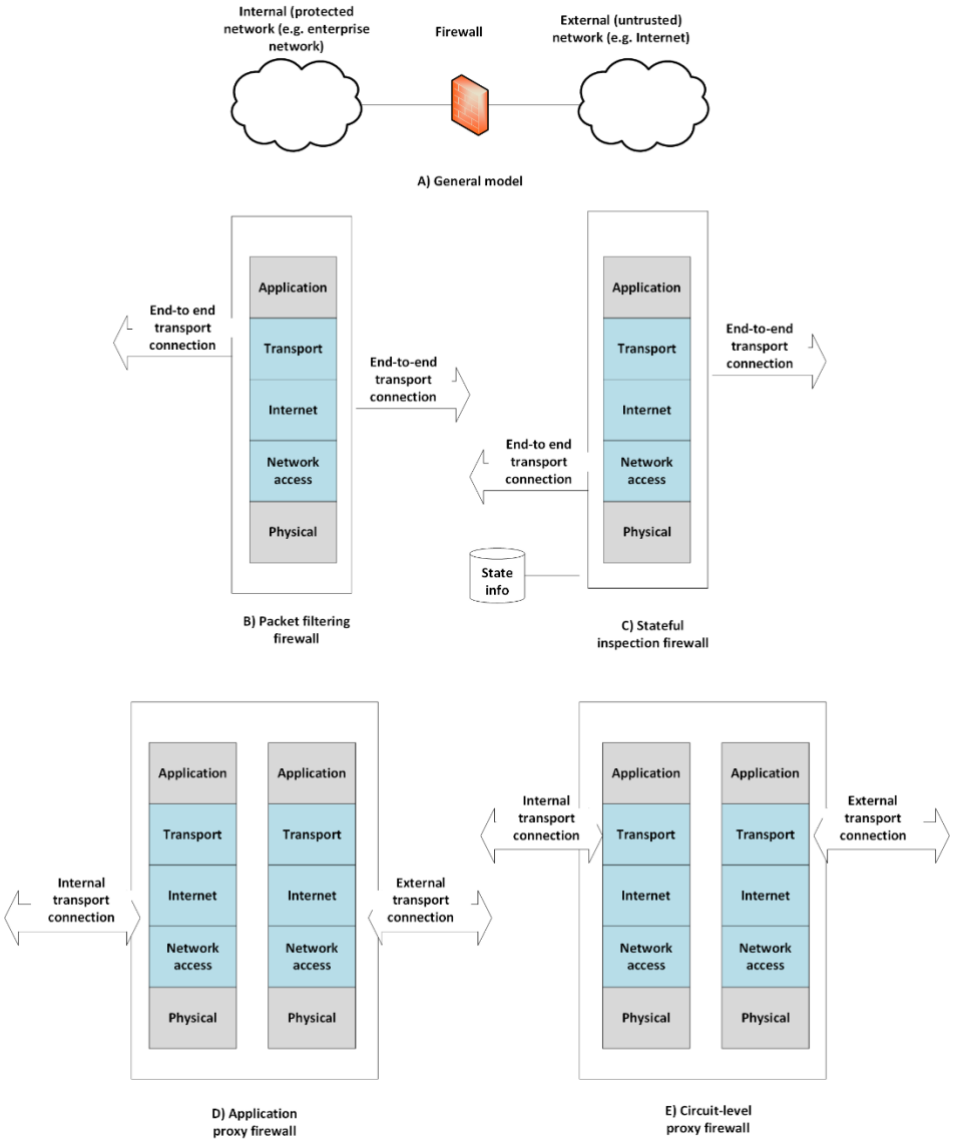


Figure 9-1: Types of Firewall

9.1.4.1 Packet Filtering Firewall

Packet filtering firewall puts into use a collection of rules to each arriving as well as departing IP packet and then advances or disposes the packet (Figure 9-1B) accordingly. Firewalls are commonly set up into filter packets move bi-directionally. The filtering rules are set according to the information that is included in a network packet:

- Source IP address: system´s IP address that initiates the IP packet
- Destination IP address: system´s IP address where the IP packet is arriving to
- IP protocol field: It specifies the transport protocol.
- Source and destination transport-level addresses: They are the transport-level (port number (TCP, UDP). SNMP or HTTP are defined by them.
- Interface: It is the part of a firewall with more than two ports, that acts like an entrance or an exit for the packet.

Packet filter is usually configured as a set of rules on the basis of matches to fields in the IP or TCP header. In packet filter exist two possible default policies:

- Default = forward: That which is not expressly prohibited is permitted.
- Default = discard: That which is not expressly permitted is prohibited.

The default forward policy makes it easier for end users to use, but it reduces security; the security administrator hast to react to new security threats as they are recognised. Organizations that are more flexible, e.g., universities, are more probable to adapt this policy. The default discard policy tends to preserve more caution. At first, everything is blocked, and services have to be appended on a case-by-case basis. Users have more visibility to this policy, and they have the chance to see the firewall as a hindrance, if more rules are written, the user's visibility decreases. This policy is preferred by businesses and government organizations.

9.1.4.2 Stateful Inspection Firewall

With the creation of a list of TCP connections that are outbound, a stateful inspection packet firewall systemizes the rules for TCP traffic. Each currently established connection has an

entry. Entering traffic is allowed by the packet filter, to high-numbered ports just for the packets which match with an entry in the specific directory.

The stateful packet inspection firewall records information about TCP connections and the same packet information is reviewed such in the case of a packet filtering firewall, (Figure 9-1C). A category of stateful firewalls follows TCP sequence numbers so they are able to prevent attacks which depend on them, e.g., session hijacking.

Another category also inspects a finite amount of data from applications for protocols.

9.1.4.3 Application-Level Gateway

The Application proxy or more officially application-level gateway, functions like a relay of application-level traffic (Figure 9-1D). The application proxy is contacted by the user aided by a TCP/ IP application (Telnet, FTP), and it requires the name of the remote host to be accessed from the user. First the user responds by providing a valid user ID and the necessary authentication information, after the gateway (the application proxy) communicates with the application located in the remote host and relays TCP segments that contain the date of the application amidst the two endpoints. Only if the gateway implements the application's specific proxy code is the service supported and forwarded outside the firewall.

Furthermore, the gateway can be configured to only work with specific features of an application that are considered acceptable by the network administrator while rejecting the other features.

This type of gateways has the tendency to be more secure than packet filters. Instead of making attempts to calculate various possible combinations that are to be either permitted or prohibited at the level of TCP and IP, the application-level gateway needs just to examine closely certain allowable applications. Moreover, all incoming traffic at the application level is easier to be logged and audited.

The processing overhead that is added to each connection is the main disadvantage of application-level gateways. Actually, there are two spliced connections between the end users and the gateway, with the gateway acting as the splice point and examining and forwarding all traffic in both directions.

9.1.4.4 Circuit-Level Gateway

The last type of firewall is the circuit-level gateway, that can be a specialized function or a stand-alone system that an application-level gateway performs for specific purposes. Similarly, with the application gateway, in the case of a circuit-level gateway, end-to-end TCP connections are not permitted; instead, two TCP connection are created by the gateway, at first within a TCP user on an inner host and itself and then another connection withing a TCP user on an outside host and itself. When the two connections are set, TCP segments from one connection to the other are usually transmitted by the gateway without checking the content of the segments.

Typically, the use of circuit-level gateways is when the system administrator places confidence to the internal users. The gateway is suitable to support application-level or proxy service on inward connections and circuit-level functions for outgoing connections and can be configured accordingly. In this configuration, the gateway has the ability to obtain the processing overhead of checking arriving application data for forbidden functions although, it does not acquire that overhead on departing data.

9.1.5 Bastion Hosts

Bastion hosts are systems labelled by firewall administrators as a crucial strong point in the security of a network. Typical features of bastion hosts are presented:

- To make it a hardened system, the bastion host hardware platform runs a secure version of its operating system.
- On the bastion host, only the administrator-defined critical services are installed (e.g., proxy applications for FTP, DNS, SMTP, HTTP).
- In some cases, for a user to access the proxy services, bastion hosts require supplementary authentication.
- Only a subset of the standard application's command set is supported by each proxy. Also a proxy is set in a way to enable separate host systems to be accessed.

- Each proxy logs all traffic, each link, and the length of each connection to keep accurate audit details. The audit log is a critical tool for detecting and stopping intruder attacks.
- Each proxy is a small and straightforward software package dedicated to network security.
- On the bastion host, the proxies are self-contained. It can be uninstalled if there is a problem with the operation of some proxy, or if a potential flaw is discovered.
- Proxies generally perform no disk access other than to read its initial configuration file. This way it can be challenging for an adversary to install Trojan horse sniffers or other malicious files on the bastion host.
- Proxies run as a non-privileged user in a secure and private directory on the bastion host.

9.1.6 Host-Based Firewall

Host-based firewalls are a software module that is used to secure individual hosts. This type of modules is provided as a separate add-on package or it can be provided with many operating systems. Similar to conventional stand-alone firewalls, a host-resident firewall filters and limits the flow of the packets. These firewalls are usually located in a server. The use of a server-based or workstation-based firewall has many advantages:

- Filtering rules can be adjusted to the environment of the host. There can be implemented specific corporate security policies for servers.
- Regardless of topology, protection is given.
- Host-based firewalls provide an extra layer of protection when utilised in combination with stand-alone firewalls. If a new type of server is connected to the network, it will have its own firewall, so there will be no need for the network firewall configuration to be changed.

9.1.7 Personal Firewall

Personal firewalls are responsible of controlling the communication between a computer device or a workstation and the Internet or a corporate network. The functionality of a personal firewall can be used in both the home and on corporate intranets. Generally, personal firewall is a software module on a computer device. In a home environment with several computers connecting to the Internet, firewall features can be incorporated in a router that links all of the home computers to Internet interfaces.

The complexity of personal firewalls is less than the one of stand-alone firewalls or server-based firewalls. The personal firewall's primary function is to prevent unauthorized remote access to the computer system. Firewalls are also able to audit departing activity in order to detect and block malicious software.

The capabilities of personal firewalls are provided by different packages depending on the netfilter on Linux systems, OS, pf package on BSD and Mac OS systems and the Windows Firewall. Generally, all inward connections are blocked for the most part, with the exception of those that the user expressly authorizes when a personal firewall is allowed. In most cases, outgoing connections are allowed.

To increase protection, there is a need for configuration of advanced firewall features. Such an advanced feature is stealth mode, which disseminates the device over the Internet by falling random communication packets, to make the system to appear as if it is unpresented. Firewalls have the ability of logging, which is a valuable tool to check on undesirable activity. Other personal firewalls grant the ability to the user to decide that only selected or validly signed applications, can provide services obtained from the network.

9.1.8 Firewall Configuration

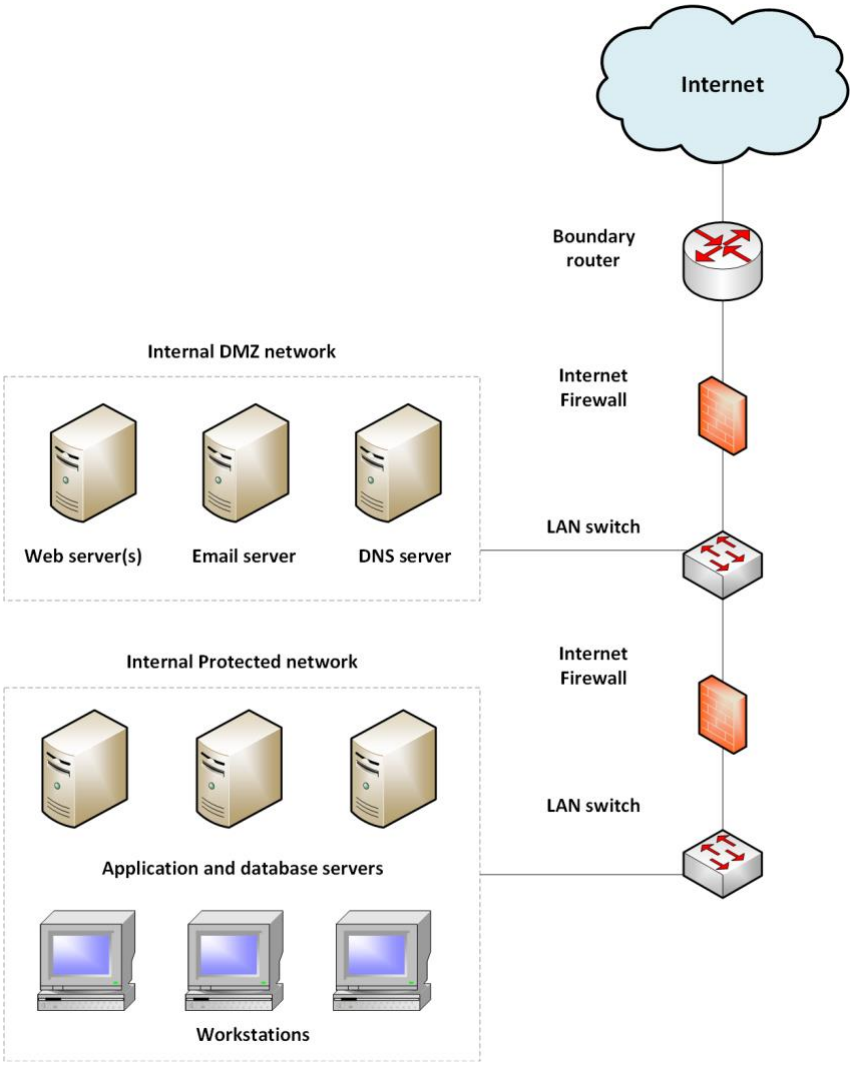


Figure 9-2: Example of Firewall Configuration

Figure 9-2 presents a typical firewall set up which includes a supplementary network segment between an internal and an external firewall. Internal firewalls secure the main part of the enterprise network, while external firewalls are situated at the edge of a local or enterprise network, in the interior of the boundary router that provides access to the Internet or a wide area network (WAN). DMZ (demilitarized zone) network is a region that is located between the internal and external firewalls. On DMZ networks are located systems which are accessible from outside but need some protections are usual. Systems in the DMZ need or facilitate external communication, examples of such systems are a DNS (domain name system) server, an e-mail server, or a corporate Web site.

In accordance with the DMZ systems' need for external communication, external firewalls serve as a means of access control and security.

9.1.9 Virtual Private Network (VPN)

A virtual private network (VPN) is a collection of computer devices that communicate over an insecure network and use encryption and special protocols to ensure protection. Servers, databases, and workstations are all connected through one or various LANs at each corporate location. The Internet and other public networks can be used to link locations, saving money from using a private network and allowing public network operators to handle the wide area network management. The public network also offers a way for mobile workers to access corporate networks from a distance.

A fundamental issued that has to be faced is security. Corporate traffic is exposed to eavesdropping, in a public network, and unauthorized users are provided an entry point. To answer this issue, a VPN is necessary. VPNs operate by implementing a protected link over an unreliable network, most commonly the Internet, using authentication and encryption in the lower protocol layers. Encryption can be achieved using firewall software or routers. At the IP level, IPSec is a common protocol mechanism used for this purpose.

A common scenario of the use of IPSec is depicted in figure 9-3. Each organization keeps LANs at distributed locations. IPSec protocols are used for traffic off site, via various private or public WANs. Each LAN is connected to the outside world through these protocols that function inside networking devices, e.g., router or firewall. The IPSec networking device encrypts and compresses the traffic which is going inside the WAN and decrypts and decompresses the traffic departing from the WAN. This kind of operations are clear to servers and workstations on the LAN. User workstations like these, have to implement the IPSec protocols in order security is granted. User workstations are often required to develop high levels of host protection, as a result of which they are directly connected to the Internet.

In Figure 9-3, it is demonstrated that implementing an IPSec within a firewall is a viable option. VPN traffic that passes through the firewall in either direction is encrypted if IPSec is embedded in a separate box behind the firewall. In this situation, firewall is incapable to

operate as filter or to perform other security functions such as access control, virus scanning, or logging. IPSec may also be embedded in the boundary router, which is outside the firewall.

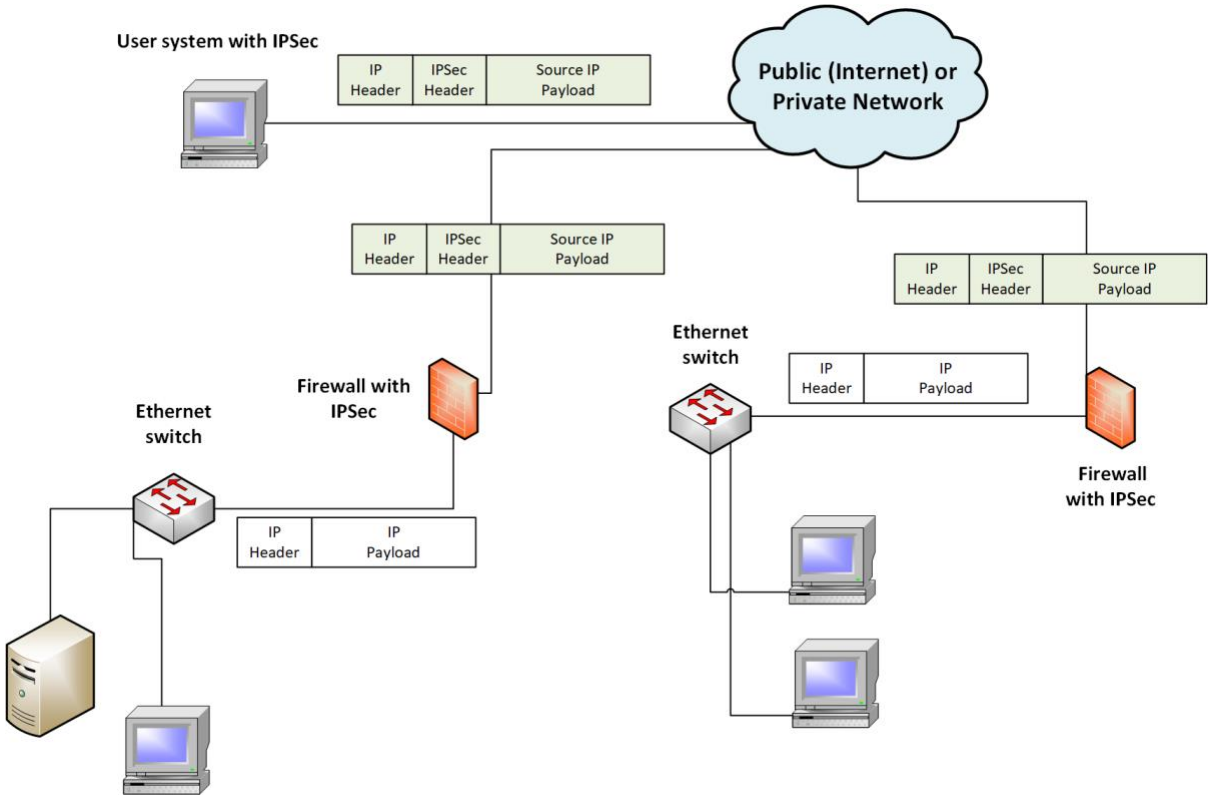


Figure 9-3: A VPN Security Scenario

9.1.10 Distributed Firewall Configuration

Host-based firewalls as well as Stand-alone firewall devices operate under centralized administrative control in a distributed firewall configuration. A distributed firewall configuration is depicted in Figure 9-4. Host resident firewalls are configured by administrators on hundreds of servers and workstation and also personal firewalls can be configured similarly on remote or local user systems. Monitoring security and setting policies across the network can be done with help of tools. These firewalls can achieve protection, opposing attacks from within and providing protection measured to fit to particular machines and applications. As previously mentioned, stand-alone firewalls consist of internal and external firewalls providing global security.

In a distributed firewall configuration, there is the possibility to deploy a DMZ that is both internal and external. For the web servers that require less protection since they do not own enough critical information could be placed outside the external firewall, in an external DMZ.

Security auditing and monitoring are important features of a distributed firewall configuration. This kind of monitoring usually involves log analysis and aggregation, firewall statistics, and fine-grained remote monitoring of individual hosts.

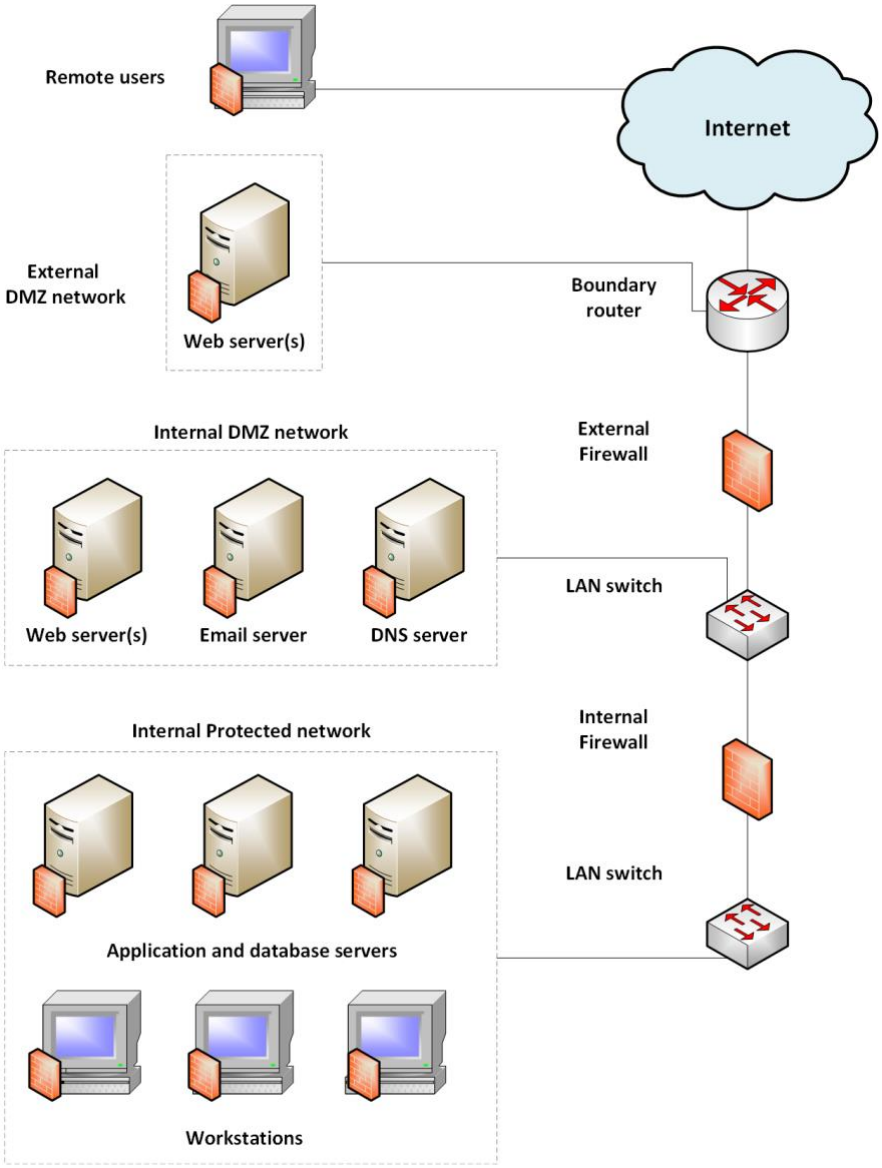


Figure 9-4: Example of Distributed Firewall Configuration

9.1.11 Firewall Topologies

We can now summarize the discussion to define a spectrum of firewall locations and topologies. The following alternatives can be identified:

- **Host-resident firewall**: Personal firewall software and server firewall software are also included in this group. These firewalls may be used as a stand-alone solution or as part of a larger firewall deployment.
- **Screening router:** With stateless or complete packet filtering, a single router connects internal and external networks. This configuration is common in SOHO (small office/home office) applications.
- **Single bastion inline:** Between an internal and external router, a single firewall device (e.g., Figure 9-1a). Stateful filters and/or device proxies can be implemented by the firewall. For small to medium-sized businesses, this is the standard firewall appliance setup.
- **Single bastion T:** Similar to a single bastion inline, but with a third network interface on the bastion that connects to a DMZ where publicly visible servers are housed. This is a typical appliance setup for medium to large businesses.
- **Double bastion inline**: The DMZ is nestled between bastion firewalls in this configuration, as shown in Figure 9-2. Large corporations and government agencies often use this configuration.
- **Double bastion T**: On the bastion firewall, the DMZ has its own network interface. This configuration is also common in large corporations and government agencies, and it may be needed in some cases.
- **Distributed firewall configuration**: This kind of configuration is mainly used by government organizations and large-scale businesses, as shown in Figure 9-4.

9.2 Intrusion Detection Systems

To properly understand the concept of Intrusion Detection Systems (IDS) it would be good to define two terms, security intrusion and intrusion detection. An unauthorized act of bypassing a system's protection mechanisms is known as a **security intrusion. Intrusion detection** is a hardware or software feature that collects and analyzes data from different areas of a device

or network in order to detect potential security breaches. In order to defend against a security intrusion there is the need to implement a proper Intrusion Detection System (IDS).

Three logical components constitute an IDS:

- **Sensors**: Sensors are the entities in charge to collect data. The input in these components can be any part of a system that could include proof of intrusion. Various kinds of sensors' input contain log files, network packets and system call traces. The information collected by the sensors is forwarded to the analyzer.
- **Analyzers**: They obtain the input from the sensors or from other analyzers. Analyzers are in charge of deciding whether an intrusion has taken place or not. Analyzers' outputs are hints that show whether an intrusion has occurred. An output could contain proof backing the conclusion that an intrusion has taken place. Sensors' inputs may be saved for further study and evaluation in the future in a database.
- **User interface**: It is the component of an IDS that gives the capability to a user to view outputs from the system and even manage the system's behaviour.

The number of sensors and analysers can defer from one IDS to another. In a distributed architecture, an IDS may use a single sensor and analyser up to several sensors spread across a range to send data to a central analyser and user interface.

The classification of IDSs is based on the type and the source of data analysed, and can be divided in the following categories:

- **Host-based IDS (HIDS)**: This type of IDS monitors the traits of a host and the events that are taking place inside that host, for any kind of suspicious activity.
- **Network-based IDS (NIDS)**: This type of IDS analyses network, transport, and application protocols for suspicious activity detection and tracks network traffic for segments and devices.
- **Distributed or hybrid IDS**: Distributed or hybrid IDSs combine data from multiple sensors, typically both host and network-based, in a central analyser that can detect and respond to intrusion activity.

9.2.1 IDS requirements

To be of practical use, an IDS should detect a substantial percentage of intrusions while keeping the false alarm rate at an acceptable level. If only a modest percentage of actual intrusions are detected, the system provides a false sense of security. On the other hand, if the system frequently triggers an alert when there is no intrusion (a false alarm), then either system managers will begin to ignore the alarms, or much time will be wasted analysing the false alarms.

Unfortunately, because of the nature of the probabilities involved, it is very difficult to meet the standard of high rate of detections with a low rate of false alarms. In general, if the actual numbers of intrusions are low compared to the number of legitimate uses of a system, then the false alarm rate will be high unless the test is extremely discriminating. This is an example of a phenomenon known as the base-rate fallacy. A study of existing IDSs indicated that current systems have not overcome the problem of the base-rate fallacy. See Appendix I for a brief background on the mathematics of this problem.

An IDS must:

- Be able to run constantly with the slightest supervision
- Be able to restore itself from crashes and restarting
- Be able to detect and resist sabotage
- Not Impose a big overhead on the host system
- Be able to be configured in accordance with the monitored system's security policies
- Be adaptable to alternations in the behaviour of the system and the user
- Be able to adjust itself to monitor many hosts
- Be able to keep functioning even if some elements stop operating
- Have the ability to reconfigure itself without the need to restart

9.2.2 Analysis Approaches

Usually, IDSs examine sensor data to detect abnormalities and/or intrusions, working in the two possible modes:

1. **Anomaly detection**: This mode includes the collection of information related to the behaviour of valid users and according to that it can detect behaviour that is not ordinary in comparison with the data collected and define an intrusion.
2. **Signature or Heuristic detection**: This mode detects intruders by comparing current behaviour to a list of documented malicious data patterns (signatures) or attack rules (heuristics).

Practically, anomaly approaches focus to define common behaviour, so they can identify unauthorized or malicious behaviour.

9.2.3 Anomaly detection

The first step in detecting anomalies is to build a model of valid user behaviour by examining and handling sensor data from the monitored system's daily activity. Once finalizing the model, each observed behaviour and the model are compared to come to the conclusion whether it is an intrusion or not.

A variety of classification methods are employed:

- **Statistical** is the analysis of the observed behaviour with the use of univariate, multivariate, or time-series models of observed metrics.
- **Knowledge based**: This method employs an expert framework that classifies observable behaviour using a collection of rules that define acceptable behaviour.
- **Machine-learning**: This Approach immediately decides an appropriate classification model from the data used as a definition, using data mining techniques.

9.2.4 Signature or Heuristic Detection

Signature or heuristic techniques are able to expose intrusions by paying attention to device events and applying a set of signature patterns to the data or a set of rules that define the data. With this method the system is led to a decision regarding whether the observed data signifies usual or uncommon behaviour.

Signature approaches pair a large array of malicious data patterns with data stored on a system. The size of the signature has to be sufficient in order to reduce the false alarm rate, while the signature is still able to detect an adequately big part of malicious data. This approach has the advantage that its cost in time and resources is relatively low, and it is widely accepted. Disadvantages of this approach can be the powerful effort needed to continuously and review new malware to generate signatures capable of recognizing it.

Rule-based heuristic identification includes the use of guidelines or rules to identify known infiltrations to the system or infiltrations that would take advantage of known vulnerabilities. Even if the behaviour is within the boundaries of existing use patterns, rules can be identified that define irregular behaviour. Usually, each machine or operating system has specific rules that it uses. Security personnel with knowledge and experience can also supplement these rules. In case this happens, according to a certain procedure, interviews with system managers and security experts are conducted in order to compile a list of documented intrusion cases and key incidents that compromise the target system's security.

9.2.5 Host-Based Intrusion Detection (HIDS)

Host-based intrusion detection systems (HIDSs) offered a specialized layer of security software to vulnerable or insecure systems. HIDSs audit activity on the system with different methods to expose suspicious behaviour. There are instances that an IDS can suspend the attack before any harm is done, however, expose intrusions, log unusual events and transmit alerts are their main purposes.

An HIDS can discover intrusions happening form within the system or from outside, this is its fundamental benefit due to the fact that this is not possible in the cases of network-based IDSs or firewalls.

9.2.5.1 Data Sources and Sensors

An essential part of intrusion detection is the data-collecting sensor. Data of constant activity are provided as input from the sensor to analysis. Common data sources are presented:

- **System call traces**: A record of a system's processes making a series of system calls.
- **Audit (log file) records**: Information about user activity is collected with accounting software in current operating systems. This type of information gives the advantage that no additional data collection software is necessary. However, the audit records may not include the appropriate information and it could be exploited by intruders.
- **File integrity checksums**: Intrusion detection can occur by periodically scanning essential files for any alternations from the desired baseline, with a comparison of existing cryptographic checksums for these files against a database of known proper values. However, there is a need to create and protect the checksums with the use of using known valid files is a disadvantage of this source.
- **Registry access**: Monitoring access to the registry is an approach used on Windows systems, because programs on these systems use a big amount of information and access to it. However, this data source has limited used because it is very Windows specific.

Sensors collect data from the chosen source, filter the collected data, and ship the result to the IDS analyser.

9.2.5.2 Distributed IDS

In a typical organization there is a distributed a group of hosts connected to a LAN or intranet that need to be defended and not a single system. With the coordination and cooperation among IDSs across each network it can be achieved a more effective defence rather than a single defence for each host with a stand-alone IDS.

Notable issues in the design of a distributed IDS are the following:

- In a heterogeneous environment, it is difficult to handle different sensor data formats that are gathered and processed inside an IDS.
- Integrity and confidentiality of the data is crucial, due to the fact that all data of the distributed network are gathered and analysed in a single point. There has to be assurance that this point is uncompromisable, so integrity and confidentiality are not compromised.

- A centralized or decentralized architecture may be used for these systems. In a centralized architecture, although it could be easier to gather and analyze data in a single node, it may cause a single point of failure. In a decentralized architecture activities and exchange of information have to be well coordinated due to its distributed nature.

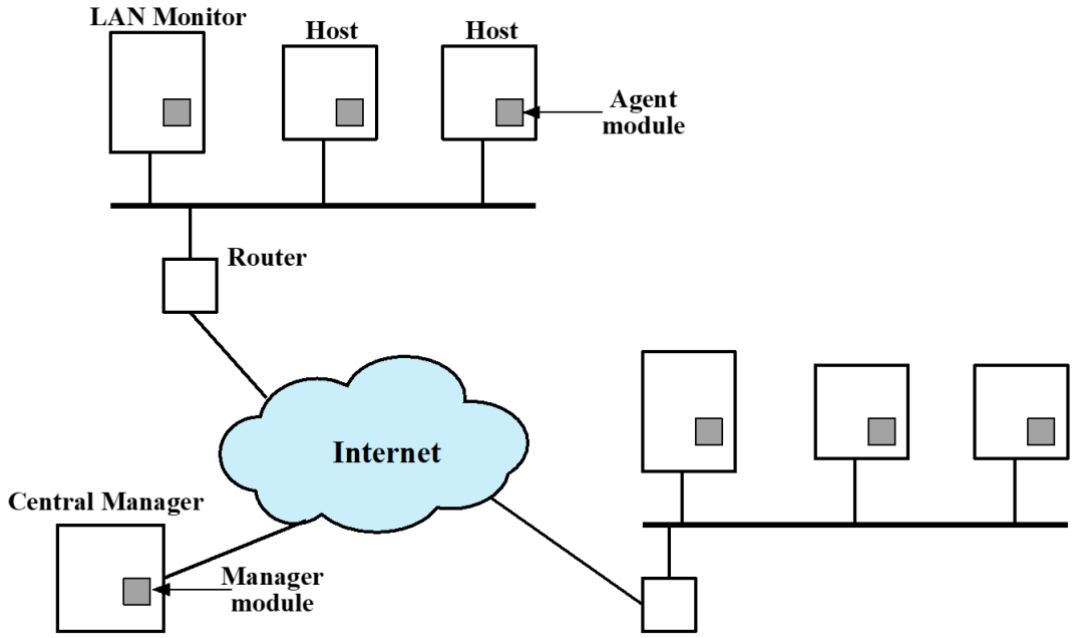


Figure 9-5 Architecture for Distributed Intrusion Detection Systems [1]

Figure 9-5 Depicts the overall architecture of a distributed IDS, that contains three main components:

1. **Host agent module**: It is an audit gathering module that runs in the background on a controlled device. It functions as a data collector for security-related incidents on the host, transmits the data to the central manager.
2. **LAN monitor agent module**: It functions similarly to a host agent module, with the exception that it analyses LAN traffic and reports the results to the central manager.
3. **Central manager module**: It receives reports from LAN monitor and host agents, which it then analyses and compares to detect intrusion.

9.2.6 Network-Based IDS (NIDS)

The network-based IDS (NIDS) examines and monitors the traffic packet by packet in real time at selected points in order to detect intrusion patterns on a network or on an interconnected set of networks. NIDS examines network-, transport-, and/or application-level protocol activity. A NIDS differs from a host-based IDS in that a NIDS examines packet traffic directed at potentially sensitive systems on a network, while a host-based system examines user and software behaviour on a host.

A standard NIDS installation consists of a number of sensors that control packet traffic, one or more servers that perform NIDS management functions, and one or more management consoles that act as the human interface.

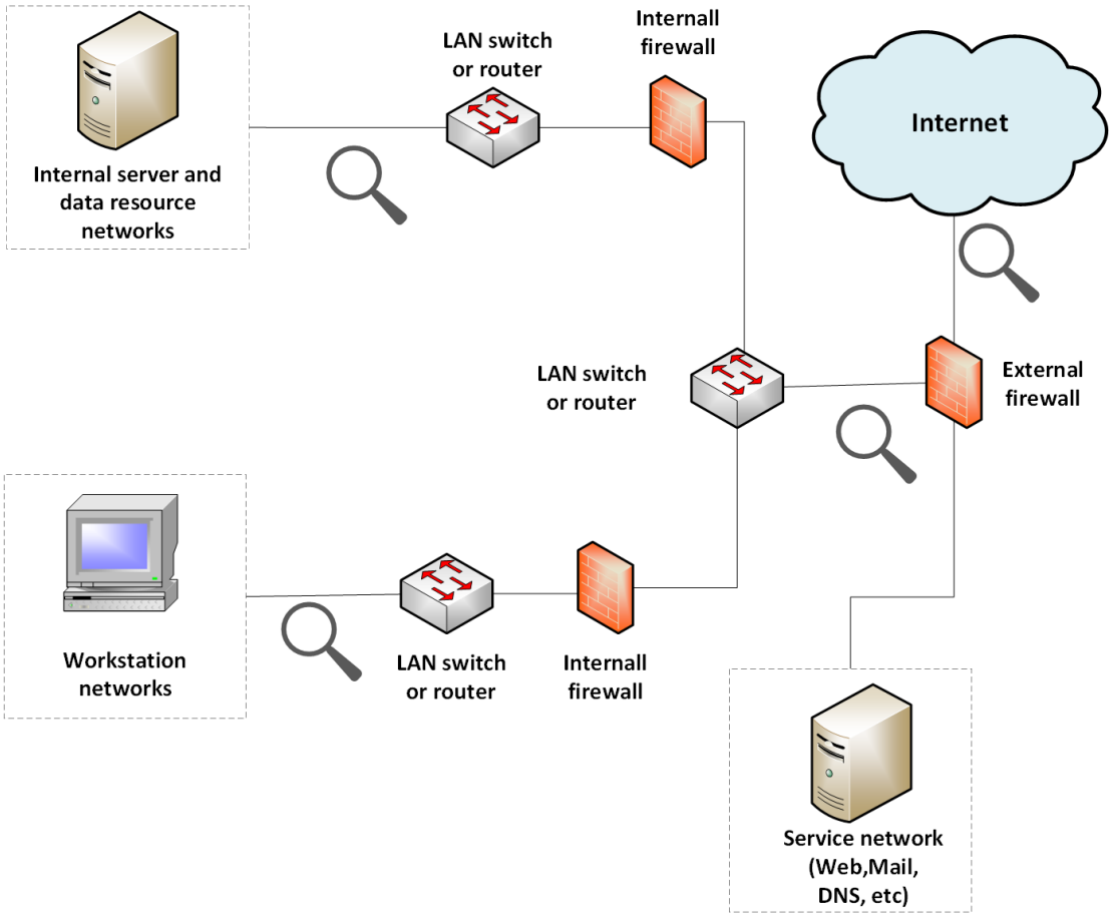


Figure 9-6 Example of NIDS Sensor Deployment

9.2.7 Intrusion Detection Techniques

As Network-based intrusion detection employs signature detection and anomaly detection in the same way as host-based intrusion detection does. Several commercial anomaly NIDS products exist. The Statistical Packet Anomaly Detection Engine (SPADE) is a well-known one, and it is available as a Snort system plug-in.

The following examples are listed by NIST as types of attacks which suit to signature detection:

- **Application layer reconnaissance and attacks**: NIDS technologies analyse several application protocols however Dynamic Host Configuration Protocol (DHCP), DNS, Finger, FTP, HTTP, Internet Message Access Protocol (IMAP), Internet Relay Chat (IRC), Network File System (NFS), Post Office Protocol (POP), rlogin/rsh, Remote Procedure Call (RPC), are the most common among them. Attack trends, such as buffer overflows, malware transmission and password guessing. that have been established as targeting these protocols are being looked for by NIDSs.
- **Transport layer reconnaissance and attacks**: The NIDS analyses TCP and UDP traffic. Attacks in these protocols include unusual packet fragmentation, vulnerable ports scanning, and TCP-specific attacks.
- **Network layer reconnaissance and attacks**: The NIDS analyses IPv4, IPv6, ICMP, and IGMP at network layer. Attacks in this level include spoofed IP addresses and illegal IP header values.
- **Unexpected application services**: The NIDS tries to figure out whether the operation on a transport link is in line with the application protocol that should be used. A host running an unauthorized application service constitutes an example.
- **Policy violations**: These attacks are among others, the use of inappropriate Web sites and the use of forbidden application protocols.

The following types of attacks are listed in NIST SP 800-94 as fitting for anomaly detection:

- **Denial-of-service (DoS) attacks**: In order to overload the target system, such attacks require significantly increased packet traffic or significantly increased communication attempts. Such attacks are well-suited to anomaly detection.
- **Scanning**: Scanning attacks occur in the case of an attacker penetrates a system or network by sending packets of various types. The intruder will learn many of the

system's characteristics and weaknesses from the responses obtained from the goal. The attacker can receive data, such as vulnerabilities of the system from the target by the responses the target provides. A scanning attack could be described as a target identification tool for an attacker.

- **Worms**: Worms can be detected when they transmit quickly and use a big extent of bandwidth, or when they cause hosts to use ports or communicate with other hosts that usually do not.

9.2.8 Logging of Alerts

While a potential violation is detected from a sensor, an alert and logs information regarding the event is being sent. The NIDS analysis module will use this data to fine-tune intrusion detection parameters and algorithms. Prevention techniques can also be designed by security administrators with the use of this information. NIDS sensors log the following typical information:

- Timestamp (usually date and time)
- Connection or session ID (typically a consecutive or unique number assigned to each TCP connection or to like groups of packets for connectionless protocols)
- Event or alert type
- Rating (e.g., priority, severity, impact, confidence)
- Network, transport, and application layer protocols
- Source and destination IP addresses
- Source and destination TCP or UDP ports, or ICMP types and codes
- Number of bytes transmitted over the connection
- Decoded payload data, such as application requests and responses
- State-related information (e.g., authenticated username)

9.2.9 IETF Intrusion Detection Working Group

Standards are necessary for interoperability in order for the development of distributed IDSs to be facilitated so that it can function across an extensive spectrum of platforms and

environments. IETF Intrusion Detection Working Group focuses on these standards. Working group's purpose is to give the description of data formats and exchange processes for distributing information of interest to intrusion detection and response, and management systems. The following RFCs were in 2007 by the working group:

- **Intrusion Detection Message Exchange Requirements (RFC 4766)**: It specifies the Intrusion Detection Message Exchange Format (IDMEF) as well as a communication protocol for exchanging IDMEF messages.
- **The Intrusion Detection Message Exchange Format (RFC 4765)**: It explains a data model for displaying information from intrusion detection systems. .
- **The Intrusion Detection Exchange Protocol (RFC 4767)**: It explains the Intrusion Detection Exchange Protocol (IDXP), an application-level protocol for exchanging data between intrusion detection entities.

9.2.10 Autonomic Enterprise security System

Recently, the idea of communicating IDSs has developed to plans which consist of distributed systems that work together to recognize intrusions and to adjust to uncertain attack profiles. The combination of those is in a central IDS, to govern and organize intrusion detection and response in the IT infrastructure of an organization. These systems have confronted two types of problems. Primary, these tools may be unable to detect new threats or significant changes to existing threats and second, updating schemes fast enough to deal with increasingly spreading attacks is difficult.

These problems have been exploited by adversaries in various ways. Developing fast spreading malicious software like worms, to develop attacks such as DoS attacks, to strike with fast before a defence has the chance to be implemented, remains a common attack approach. Although, this type of attacks is widespread, currently adversaries work in a different manner. they reduce the spread of the attack so that it can be less detectable by conventional algorithms.

Countering this kind of attacks can be done by developing cooperated systems that identify attacks based on slight hints and then adapt fast. According to said method, anomaly

detectors located at local nodes seek for evidence of uncommon activity. There are two probable cases if only this evidence exists, the local system risks a false positive in the event it reacts to the alleged attack, it risks a false negative; if it ignores the attack and waits for more facts, it risks a false negative. On the other hand, in the case of an adaptive, cooperative system, a peer-to-peer "gossip" protocol is used by the local node to inform other devices of its suspicion, in the wat of a contingency that the network is under attack. The computer responds locally to protect itself and send a warning to a central device.

The autonomic enterprise security constitutes an example of this approach and it is a model developed by Intel. This approach is depicted in Figure 9-7.

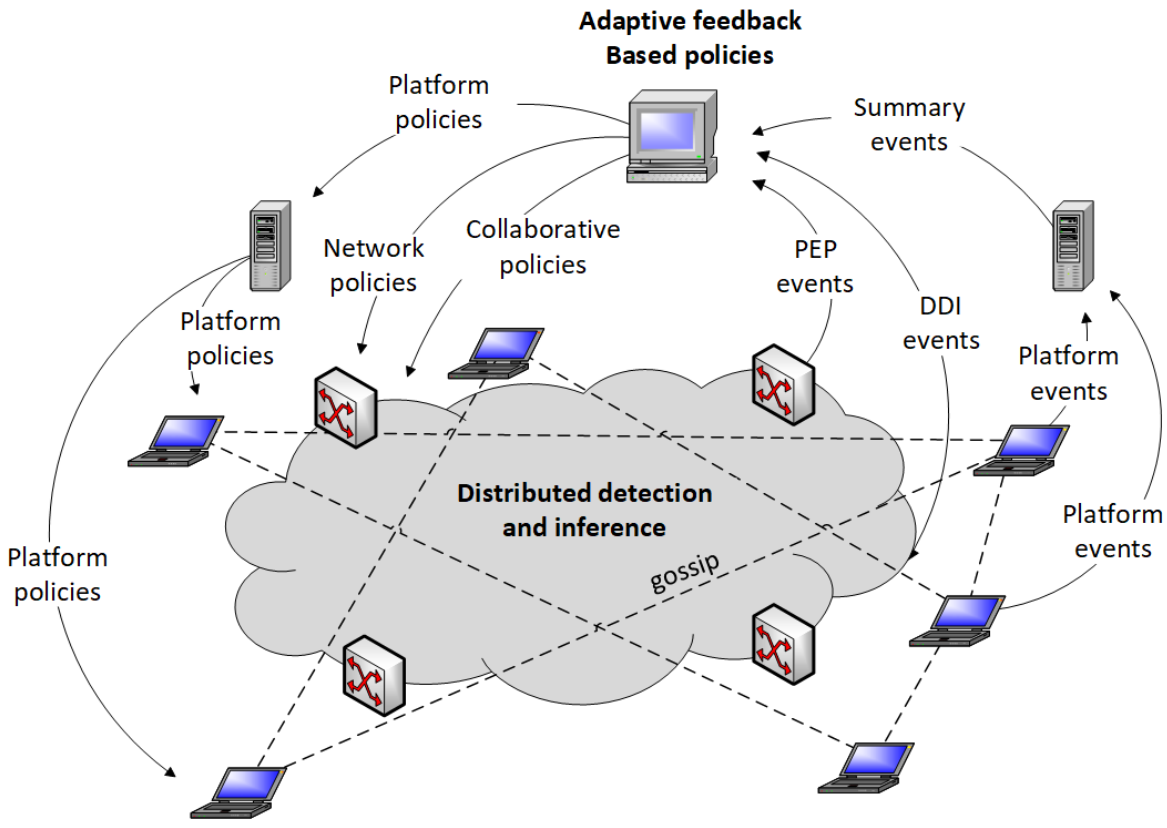


Figure 9-7: Security Architecture of an Autonomous Enterprise

The components of the approach depicter in Figure 9-7 are summarized here. A device is initially configured with a default set of security policies. These policies are modified based on sensor data from distributed sensors, then actions are transmitted to the s platforms inside

the distributed system. The central system also transmits collaborative policies to every platform that adapts the schedule of collaborative gossip messages. Central systems are guided by the following input types:

- **Summary events**: These events origin from a variety of places, are collected by transitional aggregation points such as firewalls, intrusion detection systems, or servers that function for a specific network segment, and then summed up for delivery to the central policy system.
- **Distributed detection and inference (DDI) events**: These events consist of alerts created in the event that the gossip traffic allows a platform to come to the decision an attack is in progress.
- **Policy enforcement points (PEP) events:** Trusted, self-defending platforms or intrusion detection systems (IDSs) can detect such events. These systems compare data from various sources, local decisions, and actions of individual devices to discover intrusions that are not apparent at the host level.

9.2.11 Honeypots

The honeypot is another part of intrusion detection technology. Honeypots are essentially decoy networks that are used to entice a potential enemy away from sensitive systems. Honeypots are made to:

- Turn an attacker in a different direction than accessing critical systems.
- Gather intelligence related to attacker's activity.
- Encourage the intruder to stay on the device for as long as possible so that administrators can react.

The honeypot is a resource a legitimate user has no access, this way any type of access to the honeypot is suspicious. Administrator can detect an adversary which is stuck in the honeypot and prepare to defend the system. There are not any legitimate reasons for individuals outside the network to interact with a honeypot. In this manner, any effort to exchange information with the system is probably a probe, scan, or attack. In the event that a honeypot

launches outbound communication, there is a high chance that system has been compromised.

9.2.11.1 Classifications

There are two main classifications for honeypots, low interaction or high interaction.

- **Low interaction honeypot**: This kind of honeypot consists of a software package which emulates IT systems and services in a sufficient manner as a result of which it can have a practical first interaction, although it cannot execute a full version of those systems or services.
- **High interaction honeypot**: This kind of honeypots is an actual system, equipped with a full operating system, services and applications, that are arranged in positions they can have access by attackers.

A high interaction honeypot seems like truthful targets that can occupy the attacker long term. However, significantly more resources are required, and in the event it is compromised can be used to launch attacks on other systems. This result may be unfortunate for an organization because it can create unwanted legal or reputational issues.

A honeypot with low contact is a less trustworthy target, unable to detect intruders in the early stages of an attack. Typically, this is enough for use as a part of a distributed IDS to alert for a forthcoming attack.

9.2.11.2 Deployment

Honeypot deployment can be done in variant locations. In Figure 9-8 there are depicted a probable one. The location depends on several factors, for example the information types each organization is interested in collecting or the risk level and organizations can allow to acquire as much data as possible.

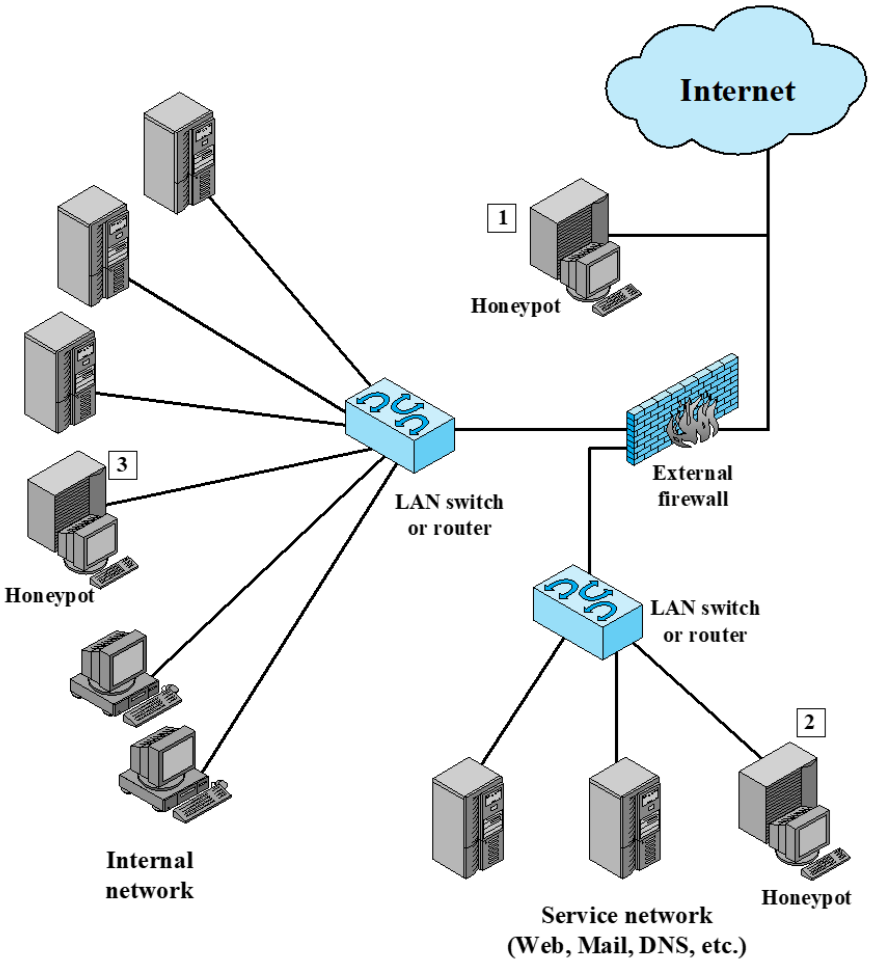


Figure 9-2 Example of Honeypot Deployment [1]

A honeypot (location 1) located outside the external firewall is useful for monitoring attempts to link to unused IP addresses within the network's reach. This deployment is useful in order to track attempts from adversaries to connect to the network utilizing unused IP addresses. Since the honeypot attracts possible attackers, the number of warnings sent out by the firewall or internal IDS sensors is reduced, making management easier. The external honeypot's drawback is that it is unable to identify and capture internal attackers.

A honeypot can also be located in the DMZ (depicted in fig 9-8 in position 2), which is a network of externally accessible services (e.g., Web and mail). However, the security administrator must ensure that any activity created by the honeypot does not compromise the security of the other systems in the DMZ. The downside of this implementation is that a standard DMZ is not completely available, and the firewall usually blocks traffic to the DMZ

that tries to access unnecessary services. This way, the firewall either must accept the risk and release impermissible traffic or minimize the effectiveness of the honeypot.

In location 3 in fig 9-8 is illustrated a fully internal honeypot. The advantages of this deployment are many, with the most important advantage to be the ability to catch internal attacks. Firewalls that are falsely configured and forward impermissible traffic from outside to the internal network, can also be detected by honeypots at this spot. Although this deployment has many disadvantages, the most severe the event that the honeypot could be compromised in a way that it attacks other internal systems.

An emerging related technology is the use of honeyfiles, that emulate legitimate documents with realistic, enticing names and possibly content. These documents should not be accessed by legitimate users of a system, but rather act as bait for intruders exploring a system. Any access of them is assumed to be suspicious. Appropriate generation, placement, and monitoring of honeyfiles is an area of current research.

9.3 Malicious Software Countermeasures

9.3.1 Malware Countermeasure Approaches - Anti-Malware

In order to deal with the threat of malware, it is better to prevent it. It is important to forbid malware to enter the system to begin with or cut off the ability of it to change the system. It is difficult to achieve perfect security with prevention but taking security measures can decrease the probabilities of an attack or an infection in the system. A proper anti-malware policy is essential, as it is the basis for implementing appropriate preventive countermeasures.

Systems must be up to date with all the current patches applied so vulnerabilities that compromise the system are reduced. Afterwards, suitable access controls have to be set on the data and that are stored on the system, in order to decrease files that any user can access, and probably infect or degrade due to the execution of a malware code. These are countermeasures to worms, viruses, and Trojans' main propagation mechanisms.

Using appropriate user awareness and training is another effective and common propagation mechanisms that works against social engineering attacks on users. This mechanism’s goal is

to equip users with awareness about these attacks, and less probable that they will take measures that will lead to their compromise.

In the event of prevention fails, technical mechanisms can be used to assists the subsequent threat mitigation alternatives:

- **Detection:** Once the infection has been initiated, conclude to its occurrence and detect the malware.
- **Identification:** After the attainment of detection, the malware that has infected the system has to be identified.
- **Removal:** After the step of identification, remove all traces of malware virus have to be removed from the infected systems so that the malware is not able to spread further.

9.3.2 Sandboxing

Sandboxing is a technique for detecting and analysing malware that involves running potentially malicious code in a virtual machine or an emulated sandbox. This way the code is run on a controlled environment and it is monitored and does not threat the security of the system. These environments include from sandbox emulators and virtual machines. In the first case memory and CPU of a target system is simulated and in the second case the full functionality of target systems is replicated. By running potentially malicious software in these environments provides the opportunity to detect complex encrypted, polymorphic, or metamorphic malware. The code is unpacked, transformed into required machine instructions, and gets decrypted and then scanned for known malware signature or has its behaviour examined as long as its execution continues. As a result of this process and analysis anti-virus signatures for new, unknown malware can be developed. Although usually malware elements appear soon after a program begins executing in some cases malware increasingly uses evasion approaches avoid detection in the analysis time used by sandbox systems, and uncertainty in the running time of each implementation constitutes a big design issue. Although enough running time can lead to better results resources and time are limited in the case of sandbox analysis.

9.3.3 Operation System Security

Computer systems, clients and servers constitute the IT infrastructure for the majority of organizations the handle and transmit data and run applications, and this makes them an essential part of the organization's functionality. Because there is a high rate of threats embedded inside an operating system while it is distributed, during the installation process, there is a good risk that a device will be compromised before it implements measures for defence or before it can install the latest patches. For this reason, designing and setting a system should be a carefully prepared process formed in a way to counter this type of threat, and to maintain security while operating.

According to NIST SP 800-123 the process must:

- Assess the threats and devise a strategy for deploying the device
- Secure the operating system first, then the critical applications
- Ensure that all sensitive information is protected
- Ascertain that adequate network security protocols are in place
- Ascertain that proper procedures are followed to maintain protection

9.3.3.1 Strategies

The Australian Signals Directorate (ASD) list of the "Top 35 Mitigation Strategies", since 2010 states that at least 85 percent of the targeted cyber intrusions investigated by ASD would have been avoided if only the top four of the strategies of the list had been in place. would have prevented at least 85% of the targeted cyber intrusions investigated by ASD given the fact that according to reports the use of certain basic hardening measures could prevent a large part of the attacks that are taking place recently. Therefore, since 2013, all Australian government agencies have been expected to use the top four tactics on this list. There are the top four strategies:

1. White-list approved applications
2. Patch third-party applications
3. Patch operating system vulnerabilities and use the latest versions
4. Restrict administrative privileges

9.3.3.2 System security planning

To begin with the deployment new systems is essential to form a plan. Planning provides assurance that the system complies with the essential policies and it is as secure as possible. Because every company has different security requirements and issues, this preparation should be guided by a broader security evaluation of the organization. The aim of the device implementation planning process is to increase protection while lowering costs. During the planning, the security requirements for the system, its application, its data and its users need to be determined. Finally, planning identifies the selection of appropriate software for the operating system and applications, as well as the appropriate personnel and appropriate training to install and manage the system.

9.3.3.3 Operating System Hardening

The base operating system is the component on which all other applications and services depend, thus it is essential to secure this first in order to secure a system in general. A well installed, patched, and configured operating system provides an intact security foundation, and usually the default configuration for enough operating systems prefers to focus on functionality and user friendliness rather than security.

There is not any universal configuration for a secure operating system, each one has its own specifications and needs to be configured accordantly, however there are some general rules and basic steps that need to be followed for a well secured system. Sometimes this can be done with the help of automated tools.

To protect an operating system, NIST SP 800-123 recommends the subsequent basic steps:

- Install and patch the operating system
- Harden and configure the operating system to meet the security requirements that have been developed
- Install and configure supplementary security controls (e.g., anti-virus, host-based firewalls, intrusion detection systems (IDS))
- Verify that the steps taken, properly address the security needs of the operating system by testing its security

9.3.3.4 Initial Setup and Patching

The installation of the operating system is the primary step of system security. New systems must be installed on a secure network and not to be expose while in the state of design. Whether network is isolated or with restricted access to the Internet, the complete installation, and the process to harden it should take place prior to the system's deployment to its final destination where it will be more vulnerable.

The basic installation has to contain the least necessary for the particular system. Additional software and application can be installed furtherly after the system is secure.

Defining a necessary password for alterations to the BIOS code for the system's initial boot or options adjustment is needed to accomplish this. Limiting the amount of media from which the device will normally boot.

Supplementary device driver code has to be check carefully due to the fact that is supported by a third party and has access to the operating's system. Validation of this driver code's source is required, because of the privileges it has in the system. Driver with malicious code can bypass security controls and compromise the system by installing malware (e.g., Blue Pill rootkit, Stuxnet worm).

Systems must be kept up to date at all times, because new software and vulnerabilities are continuously discovered. Updates are usually provided automatically in current systems.

There is a perception that automatic updates may be detrimental and that introduce instability. However, according to ASD, delaying in patch testing can leave systems vulnerable to compromise, leading to the conclusion that automatic update is preferable. In cases that availability and uptime are of principal importance for a system all patches on test systems need to be staged and validated before being deployed in production.

9.3.3.4.1 Windows Security

Microsoft Windows own a big part of "general purpose" system installations. Because of this, these systems are a target for malicious actors, and therefore security countermeasures are necessary to address this kind of challenges. Followingly there are some specific issues regarding installation, configuration, and management of these systems.

System hardening guides provided by NSA—Security Configuration Guides and tools such as "Microsoft Security Tools & Checklists" are some of the many available resources for administrators to reinforce their systems against attacks.

Security ID (SID) is defining users and groups in windows systems. The Security Account Manager (SAM) oversees storing and using this type of data. In the case of organizations with multiple systems it is preferable to be managed using domains. A common policy can be enforced by these systems on users on any system in the domain.

Discretionary access controls are implemented by windows systems to system resources (e.g., files, shared memory, named pipes). The access control list contains several entries that can grant or deny access rights to a specific SID, which can be a group of users or a single user. Integrity controls are included in later systems. These controls assign a level of low, medium, high, or system integrity to all objects in the system. The system then ensures that the subject's integrity is at least equal to the object's level whenever data is written to it.

9.3.3.4.2 Linux Security

Keeping security patches up to date on system and application code is a widely accepted and critical control for maintaining security.

On Unix and Linux systems, application and service configuration are mostly done using separate text files for each application and service. Details about system configuration are usually found in the /etc. directory or in the installation tree for a specific application.

Where appropriate, individual user configurations that can override system defaults are stored in hidden "dot" type files in each user's home directory. The name, format, and use of these files are all heavily influenced by the operating system and applications being used. Therefore, suitable training is needed for the system administrators that are responsible for the secure configuration of such a system.

Disabling services, and applications, that are not required is an important change to improve system security. The next step is to ensure that all necessary software and services are properly configured, according to each one's relevant security guidance.

9.3.3.5 Remove Unnecessary Services, Applications and Protocols

There is a possibility that some of the software packages on a server contain security flaws. Because of this if there are less software packages available to run this risk is minimized. There has to be a balance between usability, and security and that can be happened by installing the necessary software for the preservation of both. Depending on each organization needs the range of services, applications, and protocols required will vary widely. The system planning process should determine what is needed for a given system in order to provide an acceptable level of functionality while removing software that is not required to enhance security.

Most distributed systems' default configuration is fixed to improve ease of use and functionality over protection. When performing the initial installation, the supplied defaults should not be used; instead, the installation should be configured only for the packages that need to be installed. In the event more packages are needed, they can be installed at a later date and time. Other security hardening manuals, such as NIST SP 800-123, include lists of utilities, programs, and protocols that are not needed and should not be installed.

9.3.3.6 Configure Users Groups and Authentication

Users must be given access to data and services on the system based on their needs, not uniformly to all. Access controls to data and services are implemented in modern operating systems. Role-based or mandatory access control mechanisms are also available in some systems.

User in the system must be categorized according to their needs, the privileges they have, the authentication they need to provide and the type of data and information they can have access to. Users with elevated privileges to manage the system, regular users who share sufficient access to files and other data when required, and restricted access guest accounts are all types of users in a system. It is preferable that guest users only have access to elevated rights if they need to perform a job that requires them. Security is improved this way by providing a less opportunities to an attacker to take advantage of those privileged users'

actions. Further, special tools are provided by operating systems to aid administrative users in only elevating their rights when absolutely necessary.

9.3.3.7 Configure Resource Controls

Subsequently to the definition of the users and their associated groups, to fit the required policy, acceptable permissions can be set on data and resources. This is useful for limiting which users are allowed to run those programs, and more specifically the users that modify the system state. Security hardening guides may provide recommendations for changes to the default access configuration for security to be improved.

9.3.3.8 Install Additional Security Controls

Additional protection tools, such as IDS or IPS software, anti-virus software, or device whitelisting, may be installed and configured to enhance security. Operating systems may supply these tools as part of the installation, but usually these tools are not configured or enabled by default.

Appropriate anti-virus that is effective against a variety of malware types is an essential security component on many systems, given the predominance of malware. Because of the broad use of Windows systems and the number of attackers focusing on them, anti-virus products have been used on them in the past. However anti-virus has been developed for other operating systems since its usage is also broadening.

9.3.3.9 Test the System Security

Security testing constitutes the last process in the securing the base operating system procedure. This final phase checks to see if the previous security configuration steps were performed correctly, as well as identify any vulnerabilities that need to be fixed or handled.

There are systems specifically developed to analyse system checking for bugs and bad configuration practices, as well as to ensure that it meets the fundamental security

requirements. This operation should be carried out after the device has been hardened, and then performed on a regular basis as part of the security maintenance procedure.

9.3.4 Vulnerability Management

9.3.4.1 Maintenance

Vulnerability management's first aspect is related to the implemented controls ongoing monitoring and maintenance ensuring their constant proper operation and suitability. Someone should be in charge of the organization's maintenance procedure, which is organized by the security officer. Maintenance responsibilities include the following:

- The constant review of controls to ensure that they still operate as expected
- The upgrading of controls when new requirements are identified
- Assurance that controls are not affected adversely by changes to systems
- There have been no new risks or vulnerabilities discovered

This review consists of a regular examination of log files to ensure that different elements of the system are running properly and to provide a baseline of operation against which irregular events can be evaluated during incident handling. Maintenance aims to ensure that the controls continue to work as intended, and therefore that the organization's risk exposure remains as defined.

9.3.4.2 Security Compliance

Security compliance checking is an inspection procedure to examine the security procedures of an organization. It aims to confirm agreement with the security plan. The inspection may be carried out by either internal or external staff. It is typically on the basis of the use of checklists, that confirm the creation of proper policies and plans, the choice of suitable, and that the proper usage and maintenance of the controls.

9.4 Referencing

[1] “Computer Security: Principles and Practice”, 4/e, by William Stallings and Lawrie Brown, 2018.

[2] “Computer Security: Art and Science”, 2/e, by Matt Bishop, 2019.

[3] “Computer Security Fundamentals”, by Chuck Easttom, 2020

10 Security Operations: Recovery & Incident Response

Author(s): Filippos Pelekoudas Oikonomou
Maria Papaioannou
Georgios Mantas
Jonathan Rodriguez
Claudia Barbosa

The advancement in processes to act effectively against computer incidents is considered important for most organizations. Security incidents are going to happen sooner or later to most organizations. In general, the majority of incidents are related to risks with small impacts on the organization, but in some cases, it can occur a more severe incident. There is a need for reflection of the range of possible consequences of incidents on organizations by incident handling and response procedures and a suitable response to be designed. By implementing appropriate procedures ahead of time, an organization is able to avoid the unfortunate situation that occurs in the event of an incident and the personnel being unprepared for response.

NIST SP 800-61 [Computer Security Incident Handling Guide, August 2012) specifies the following advantages of owning an incident response capability:

- Responding to events in a coordinated manner to ensure that the proper measures are taken.
- Assisting employees in recovering rapidly and effectively from security incidents, reducing data loss or theft and service interruption.
- Using information gathered during incident response to better plan for future accidents and provide stronger system and data security.
- Taking care of any legal problems that could occur as a result of the incident.

10.1 Events and Incidents

According to NIST SP 800-61 an event is “any observable occurrence in a system or network”. There are many cases of events like a user sending an email or having access to a file. Adverse events are defined as events that have a negative impact on a system. Example cases of adverse events may include, among others, a system crash, packet floods or execution of a malware that destroy data.

A computer security incident, according to NIST SP 800-61 is a breach of computer security rules, reasonable usage policies, or common security practices, or the possibility of a violation. Example cases included as incidents are, attack on a system causing it to crash or sensitive information exposed by a user and handled on an adversary.

10.1.1 Incidents Happen

Protection is never optimal, and it cannot prevent every possible scenario. There are cases that it inevitably breaks down. These cases, when a security attack is taking place, are labelled as security incidents, compromises or breaches.

10.1.2 Incident severity

Not all incidents are severe in the same degree. The incidents are separated in four categories depending on their severity degree: False alarms, minor incidents, major incidents and disasters.

- False alarms

By this term we consider the situations that seem to be incidents or potential incidents but they are activities that cause no harm to the system. In some cases, the actions that employees, network managers, or system administrators perform daily in their job position, resemble the actions of an attacker. Intrusion detection systems (IDSs) are mistakenly flag an number of legitimate activities as suspicious. In reality, in almost all IDSs, a vast majority of suspicious activities is revealed to be false alarms or false positives.

These false alarms, which are handled by the on-duty staff, are very costly in security time, which is a valuable resource. In some cases, that there are too many false alarms, there is a possibility to lead to dull readiness to investigate each potential incident. This way, real incidents could go unnoticed.

- Minor Incidents

Minor incidents are above in the severity scale. Minor incidents are true breaches that the personnel that is on duty must handle and that do not have broader implications for the organization. A case of a minor incident is, for example, a virus infection affecting a number of computer devices. Minor incident response methods are usually breach-specific and for this reason are challenging to discuss about in general.

- Major incidents

Major incidents, on the contrary, have a large impact for the IT personnel, that stays on-duty, to handle. For major incidents, numerous organizations are creating specific computer security incident response teams (CSIRTs). A computer security incident response team (CSIRT) is in charge of quickly identifying accidents, mitigating the vulnerabilities that were exploited, reducing damage and disruption, and restoring computing resources in large or medium-sized organizations. In addition to having IT and IT security professionals, CSIRTs typically have members from also other departments across the organization.

Major incidents can be hazardous. These incidents must be handled fast, efficiently, and effectively by the organizations in order to contain losses.

- Disasters

Floods, fires, and other kinds of disasters are beyond the abilities of even CSIRTs. Sometimes disasters threaten organization and business continuity, that is the maintenance of the day-to-day revenue-generating operations. Organization and business continuity planning purpose is to keep the particular entity running continuously or getting it back in operation as fast as possible.

10.1.3 Security Incidents

A variety of incidents may be classified as a security incident. In fact, any behavior that jeopardizes one or more of a system's traditional security services of confidentiality, honesty, availability, transparency, authenticity, and reliability is considered an incident. Unauthorized access to a system and unauthorized alteration of information on the system are examples of these. A person's unauthorized access to a device includes:

- Getting access to details that someone else is not authorized to see
- Getting access to information and handing it on to someone who is not authorized to have access to it.
- Attempting to get around a system's access control mechanisms.
- Using someone else's password and user id for some purpose.
- Attempting to restrict access to the system to someone who has not been given permission to do so.

The following are examples of unauthorized changes to data on a device:

- Attempting to tamper with details that might be useful to anyone else.
- Trying to change details and/or services without permission.
- Information is being processed in an illegal manner.

10.2 Organizing the Incident Response Capability

10.2.1 Need for incident Response

The need for an incident response capability emerges due to the fact that the adversaries compromise the data of users or organizations. Rapid and effective response is necessary in order to avoid unfortunate results, breaches and harms. Incident response capability helps to respond immediately to the threats, to take measures appropriate for each case of attack and minimize the loss of data and assets.

Organizations that pose regulations and policies for incident response are OMB, FISMA and Federal Information Processing Standards (FIPS).

10.2.2 Principles of an Incident Response

- Speed

Major security threats, intrusions and breaches to the organization's continuity are creating harsh time pressure. Adversaries are continuously doing damage until they are ceased. Attackers are finding also ways to make their actions more complex to detect and resolve. Even after the organization ceases the attack, the necessity for speed remains. In some cases, there is a chance for important corporate systems to fail, and their failure can have severe consequences to the organization. Fast recovery is crucial for minimizing damage.

- Accuracy

Accuracy is as important as speed. A typical mistake which individuals under pressure make is to respond carelessly, performing an action without priory having a complete understanding of the issue. Hurried response may deter individuals to the real root cause of the problem, that still remains unseen and allows the adversary continuously do damage while the problem solver is occupied with solving the wrong problem.

- Planning

In order to be able to respond both fast and effectively there is the need to prepare in advance. The actions taken prior to an incident sometimes are more essential than the actions taken subsequently of the occurrence of an incident. Organizations should plan specifically how a respond scenario will perform in case of major incidents and disasters. Probably is more suitable defining incident response as reacting to incidents in accordance with a plan. During the incident it is not effective to start deploying a response plan from scratch. Of course, there is not a plan that can respond one hundred per cent effectively to any threat. However, improvising inside a given plan is more efficient than working without any.

- Rehearsal

Another major factor that aids to a fast and effective incident response is rehearsal. Minor incidents are ordinary for accuracy and speed to be normal. In the case of major incidents and disasters, rehearsals are essential, and organizations should perform them in frequent time spams because with the repetition the performance is becoming better. The first and simplest type of rehearsal is the walkthrough (also called table-top exercises). In this management and

key personnel arrange meetings and examine, step by step, which role has every individual during an incident.

A plan scenario exercise is often used in walkthroughs, in which an initial scenario describes the incident or disaster. In some cases, live tests for essential systems have the individuals take the actions like a real situation has occurred. Live tests reveal flaws that walkthroughs are unable to. Live tests are costly, so organizations perform them less often than walkthroughs.

10.2.3 Policies, Plans and Procedure Creation

10.2.3.1 Policy elements

Policies are the rules that govern incident response and differ between each organization. Although, there are some standard elements that define these policies, according to NIST SP 800-61:

- Statement of management commitment
- Purpose and objectives of the policy
- Scope of the policy
- Definition of computer security incidents and related terms
- Prioritization or severity rating
- Performance measures
- Reporting and contact forms
- Organizational structure and definition of roles, responsibilities, and levels of authority

10.2.3.2 Plan elements

Every organization need to have an approached that is formal centered and coordinated to respond to incidents. That includes a plan that states the main framework for the implementation of the incident response capability. Due to the fact that every organization has a different structure, this plan has to be adjusted to each organization and each

organization's needs. The plan has to specify the resources that are needed for the response and the management support. According to NIST SP 800-61 the incident response plan has to contain the elements:

- Mission
- Strategies and goals
- Senior management approval
- Organizational approach to incident response
- Metrics for measuring the incident response capability and effectiveness
- Basic roadmap

The plan of each organization has to be developed and approved by the management. Then it needs to be implemented according to its need and be reviewed under a specific time stamp, so the organization can ensure its effectiveness and apply updates if needed.

10.2.3.3 Procedure elements

The procedures have their basis on the plans and the policies each organization has introduced. Although there are some standard procedures that, according to NIST SP 800-61 "are a delineation of the specific technical processes, techniques, checklists and forms" that are used in order an incident response capability to perform. These standard procedures have to be deployed in detail and to be understandable to ensure that the priorities of the organization are manifested in the operations. Additionally, they have to minimize errors and especially the ones that are caused by mishandled incident situations. The life cycle of procedures is the following according to NIST SP 800-61:

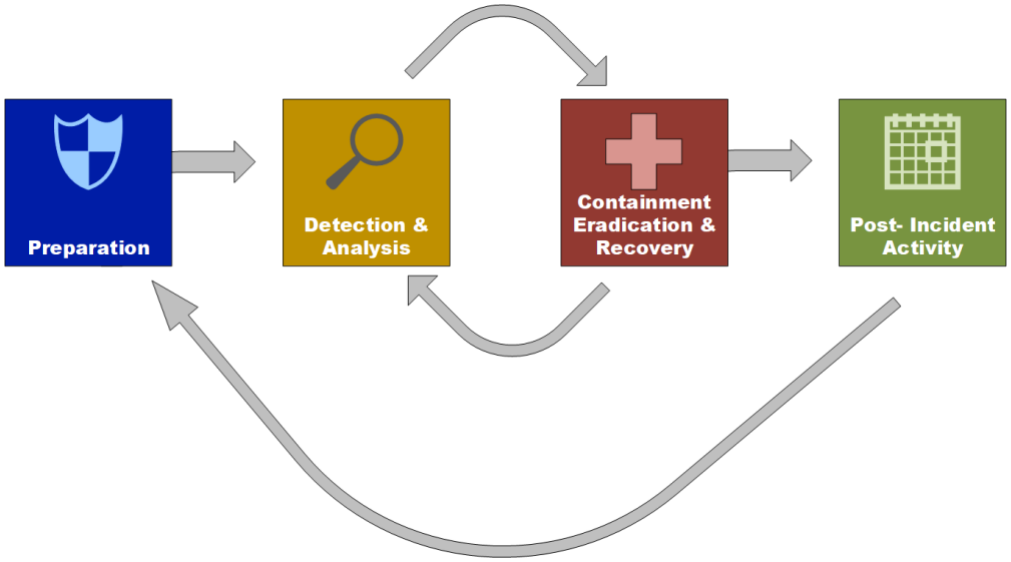


Figure 10-1: Incident Response Life-Cycle [33]

- Preparation
- Detection and Analysis
- Containment Eradication and Recovery
- Post- incident activity

But we can also simplify these procedures to:

- Preparation
- Detection
- Respond
- Document

10.2.4 Sharing information with third parties

Outside parties have to be informed by organizations occasionally for an incident. Such parties may be law enforcement, fielding media inquiries, experts or an internet service provider (ISP). Organization can also share information before an incident occurs to improve their response. Information sharing with outside parties has to be discussed within the organization between the incident response team, management and legal department before procedures and policies concerning information sharing are established. If this happen,

sensitive information will be available to unauthorised parties that can lead to another way of harm or loss of data for the organization. The team is responsible to document all the parties and also documents and communications with them for liability and evidentiary reasons.

Figure 10-2: Communications with Third Parties [33]

Fig. 10-2 illustrates the communications between the incident response team and outside parties that are going to be analysed below.

10.2.4.1 The Media

Communication between the Incident Response team and the media needs to be established that are in accordance with organizations policies on media interaction and information disclosure. Two type of contacts, a single point and a backup, are designated between the organization and the media for the discussion of incidents. To prepare these designated contacts, NIST SP 800-61 proposes the following actions:

- Training sessions to be conducted on interacting with the media regarding incidents.

- Procedures to brief media contacts should be established, on the matters and sensitivities concerning an incident before examining it.
- Maintenance of a statement of the current status of an incident for keeping the communications consistent and up-to-date.
- Advice the personnel about the general procedures regarding the media inquiries handling.
- Hold press conferences during incident handling exercises.

10.2.4.2 Law Enforcement

Incidents can have an extent on legal issues for an organization. There are several levels of law enforcement for the investigation of incidents depending on the country (FBI, district attorney offices, state law enforcement in US). The incident response team has to be acquainted with its various law enforcement representatives, prior to the occurrence of an incident, in order to discuss about conditions under which incidents have to be mentioned, how the report has to be done, which evidence are needed for collection and in which way they have to be collected.

Like in the case of media, law enforcement has to be contacted through designated individual and according to the procedures of the law and the organization. There are cases that organizations appoint a team member responsible as a point of contact with law enforcement. The responsible individual has the capability to understand the procedures and the ways under these communications will occur and which data are allowed to be discussed or are necessary to be revealed.

10.2.4.3 Other Outside Parties

There are cases that the organization needs to discuss incidents with other third parties.

- **Incident Reporting Organizations**: US-CERT in USA
- **Organization's ISP**: assistance from ISP may be needed on some incidents

- **Software Vendors**: suspicious activity can arise from a certain flaw of a software, so organization may need to communicate with a software vendor about a certain incident regarding their software.
- **Owners of Attacking Addresses**: sometimes the origin of the attacks is an external organization's IP address, thus the organization has to initiate contact with this particular
- **Other Incident Response teams**: some issues may be identical to others so an organization can keep connections with other organizations in order to discuss particular experiences with incidents.
- **Affected External Parties**: incidents in one organization can affect other external organizations or individuals (e.g., clients) so the organization has to define a policy of communication in this type o situations.

10.2.5 Incident Response Team Structure

The incident Response Team is responsible to be contacted when an incident takes place. It needs to be reachable by the members of the organization and available to act, according to the plan, in any given time. Its purpose is to handle the incident, analyse the data, find out the impact of the incident and take action to lessen the damage. Its success level is proportional to the cooperation and the participation of its members.

10.2.5.1 Team Models

There are different structures an Incident Response Team can have:

- **Central Incident Response Team**: one particular team handles all the incidents within an organization. Effective for small organizations.
- **Distributed Incident Response Team**: many response teams for different purpose and responsibilities for each section of the organization. Effective for large organizations.
- **Coordination Team**: an incident response team which provides advice to different other teams within the organization. Its role is simply advisory and has no authority over them.

Incident response team can have the following models regarding the personnel:

- **Employees**: the incident response is performed by technical support from contractors.
- **Partially Outsourced**: the incident response work is partially outsourced by the organization.
- **Fully Outsourced**: the incident response work is fully outsourced by the organization.

10.2.5.2 Team Model Selection

There are factors to be considered in order to decide which structure or model an organization should follow:

The Need of 24/7 Availability: in most cases constant surveillance and availability of the Incident response team is needed. That means that the incident response team has to be reachable or on-site in order to encounter the incident as soon as possible. Usually, this need appears in the cases that an organization is working with other organizations.

Full Time vs Part-Time Team Members: some organization cannot afford to have an incident response team fully available, so in some cases there is a team that handles the incident at first (e.g., IT service desk) until an external or a part time incident response team is informed and act. The service desk can also be trained in basic incident response handling.

Employee Morale: it is difficult for a skilled team to be 24/7 available for the incident handling. Many employees in these positions are overly stressed. The best practice here is to define roles and responsibilities to limit the amount of unnecessary work for the employees and boost their morale.

Cost: as referred also in the full time vs part time factor, cost has a major role in the deployment of the incident response team. Factors that increase the cost is the number of people in the team, their level of expertise and their working hours.

Staff expertise: people who are focused on incident handling have a deeper knowledge on how to respond on a certain incident. On the other hand, technical personnel inside the organization have the knowledge of the organization environment. So, in many cases a good cooperation in both sides is necessary to resolve an incident.

In case an organization prefers the option of Outsourcing then the following factors have to be taken into consideration:

- **Current and Future Quality of Work**: a way to assess of the quality of work of the contractor needs to be established inside the organization in order the outsourcer to be aligned with the present and upcoming needs of the organization.
- **Division of Responsibilities**: due to the sensitive data an organization possesses, it is crucial to define the action and the permissions that are to be given to outsourcers for this data.
- **Sensitive Information Revealed to the Contractor**: it is important to divide the responsibilities of the incident response to avoid unauthorised access to information. Also, special agreements for non-disclosure can be applied.
- **Lack of Organization- Specific Knowledge**: the organization has to provide the contractors with all the updates and the details of the infrastructure and the events happening in any given time, in order to make their response more efficient.
- **Lack of Correlation**: lack of correlation between the data existing and the data given to the outsourcer is a major issue during an incident response. The provision of critical data though, maybe can make the incident response more efficient but also creates vulnerabilities in the system by handling data to third parties.
- **Handling Incidents at Multiple Locations**: physical presence is necessary in many cases of incident handling. Thus, is why an organization should have people on-site communicating with the outsourcers that are located in their position outside the organization.
- **Maintaining Incident Response Skills In-House**: complete outsourcing may come with some issues in incident response. This is why it is preferable for an organization to have basic incident response personnel in-house.

10.2.5.3 Dependencies within Organizations

Inside an organization there are a number of groups that need to participate in the incident handling process. Groups that may help the incident response team are:

Management: with the cooperation of the management, the policies for incident response are established as well as the budget and the responsible staff.

Information assurance: members of this group are needed in order to define which information is to be handled to other parties.

IT Support: IT technical personnel can be an aid to the incident response team given the fact that they have a good understanding of the technology and the infrastructure of a current organization as well as skills to handle some incidents.

Legal Department: the legal department has to be fully aware of the policies and the procedures of incident handling and be sure that they comply according to the law.

Public Affairs and Media Relations: as previously mentioned, some incidents are required to be reported to the media, so there has to be some individuals responsible for this communication.

Human Resources: can define the process of employing skilled individuals and also assisting with disciplinary proceeding in the staff that are suspected to cause incidents.

Business Continuity Planning: incident response policies and plan has to be in adjustment with the business continuity plan.

Physical Security and Facilities Management: due to the fact that some incidents are happening through physical means- hardware, the incident response team has to be in communication with the team responsible for better understanding of the incident and faster and more efficient response.

10.2.5.4 Incident Response Team Services

The incident response team has other responsibilities except being able to respond to incidents and events:

- **Intrusion Detection**: before responding to an incident, it is better to detect an intrusion, so an intrusion respond team has to be able to detect the incidents before they occur.

- **Advisory Distribution**: the team can also perform advisory updates to the rest of the organization by providing information about the possible vulnerabilities and actions that have to be executed in the event of an intrusion.
- **Education and Awareness**: in correlation with the advisory distribution the team needs to educate the members of the organization with basic skills of incident handling. That can be useful for the response of an incident when it happens, before it reaches the awareness of the incident response team.
- **Information Sharing**: the incident response team can be a part of information sharing groups and manage the information shared within and outside the organization.

10.3 Incident Handling

The incident response process has various phases. According to NIST SP 800-61, the starting phase is about to establish and prepare the incident response team, and to obtain the necessary resources. In the process of preparation, the organization tries to reduce the number of incidents that will occur. Inevitably some incidents will occur. Detection is the step necessary to give the hint in the organization that an incident has occurred. An informed organization is able to mitigate the attack or the incident and thus recover from it. After the organization handles the incident, it has to document the problem and how it was handled to be prepared for the next similar incident. In figure 10-1 we can see the connection between the post-incident activity (documenting) and the preparation.

10.3.1 Procedures

Managing security incidents involves processes and controls that focus on:

- Preparation
- Documenting breaches in security for future reference
- Detecting and reacting to security breaches
- Detecting potential security incidents

10.3.1.1 Preparation

Preparation is the key for an efficient incident response capability. The organization has to establish the incident response capability and also ensures that its systems, networks and other assets are secure in order to prevent breaches. The incident response team is not utterly responsible for the incident prevention, rather than handling the occurred incident, although, preparation is essential for the good functionality of an incident response program.

Preparation has also its own different phases. To begin with there are certain tools and resources that can be used for incident handling and can be separated in the subsequent categories, according to NIST SP 800-61:

- Incident Handler Communication and Facilities
    - **Contact information** of team members and other facilitators
    - **On-call information** inside the organization
    - **Incident reporting mechanisms** (email, addresses, instant messaging systems, phone numbers, online forms)
    - **Issue tracking** for incident status and information
    - **Smartphones**
    - **Encryption software** for communication between the team members
    - **War room** for central communication and coordination
    - **Secure storage facility** for sensitive information and materials
- Incident Analysis Hardware and Software
    - **Digital forensic workstations and backup devices** to preserve files and data
    - **Laptops** for activities related to incident handling
    - **Spare workstation, servers and networking equipment** for many purposes or emergency use
    - **Blank removable media**
    - **Portable printer** print, copy log files or other documents from non-networked systems
    - **Packet sniffers and protocol analyzers** capture and analyze network traffic
    - **Digital forensic software**
    - **Removable media** to gather evidence and log files

  - **Evidence gathering accessories** (recorders, cameras, evidence bags and tags etc.)
- Incident Analysis Resources
  - **Port lists** (commonly used ports)
  - **Documentation** for OSs, applications, protocols, intrusion detection products
  - **Network diagrams and lists of critical assets** (database servers)
  - **Current baselines** of expected network, system and application activity
  - **Cryptographic hashes** of critical files, speeding up incident analysis verification and eradication.

10.3.1.2 Detection

There are many ways that an incident can occur and in each case the response may be different. The following list is presented with the routes through an incident can occur, according to NIST SP 800-61:

- **External/ Removable Media**: an attack that is performed through a removable media
- **Attrition**: an attack that deploys brute force as a method of acting and weakens a system or a service or sometimes it makes it inaccessible (e.g., DoS attack)
- **Web**: attacks may occur from a web server or a website (e.g., malicious scripts running on a webpage)
- **Email**: attacks may be executed through an attachment or an email.
- **Impersonation**: something benign can be replaced with something malicious and through this an attack can be performed
- **Improper Usage**: when an organization's usage policy is violated, incidents may occur
- **Loss or Theft of Equipment**: equipment loss (laptops, tokens, media, data assets) can lead to incidents
- **Other**

Users or administration staff may detect security related incidents and can report a system malfunction or anomalous behaviour. These reports are important, and It should be

encouraged for employees to apply them. Any alleged device flaws should also be noted and broadcast by staff.

Automated tools can also be used for security incidents detection. These tools analyse information which is gathered from the systems and connecting networks. Evidence can be reported by these tools of either a precursor to a probable upcoming incident or indication of a current incident appearing. The following are some of the tools that can be used to identify accidents:

- System integrity verification tools: They scan essential system files, directories, and services to make secure that there has been no unauthorized change.
- Log analysis tools: They use pattern recognition techniques to evaluate the information gathered in analysis logs in order to identify possible security incidents.
- Network and host intrusion detection systems (IDS): These systems track host and network behaviour and equate it to different attack signatures on a regular basis to identify potential security incidents.
- Intrusion prevention systems: These systems improve intrusion detection systems with the ability to block in real time detected attacks. There is a necessity for careful use of these systems, because there is a chance to cause problems in the event of responding to a mistakenly diagnosed attack and cause a reduction in system functionality when not justified.

The accuracy, the patterns and the signatures that are used in these automated tools' configuration is a major factor that determined their effectiveness. There is a necessity for continuous update of the tools need so they can reflect new attacks or vulnerabilities. The automated systems need to be regularly updated so they can track changes in current attacks and vulnerabilities. Security administrators have difficulties to keep pace with the continuous evolution of security risks to their systems and to respond in a timely manner. The risks to the company from this delayed response can be minimized with the aid of automated software.

Risk assessment process is the one which defines whether to deploy automated tools or not based on the organization's security goals and objectives. Significant resources, monetary and time, are necessary in order to Deploy automated tools.

10.3.1.2.1 Incident analysis

Incident analysis is part of the detection process. Some incidents can be detected with ease although, there are others that they cannot be specified, probably because they are not associated with clear symptoms. Sometimes a small unwanted change can be a signed of an incident. The detection of an incident is a difficult task, so the team must be equipped with what is necessary to evaluate symptoms and analyse indicators in order to understand which incident is occurring and the time it is occurring. When there is an indication that states that an incident has occurred the team must respond quickly to analyse it and validate it to determine incident's scope and origin.

NIST SP 800-61 proposes the following for making the incident analysis simpler and efficient:

- **Profile Networks and Systems**: in order to be aware of any change of a given activity it is optimal to measure the characteristics of each activity, this is called profiling.
- **Understand Normal Behaviours**: the members of the incident response team should be aware of how networks, systems and applications behave in order to be able to detect anomalies in their functionality.
- **Create a Log Retention Policy**: except measuring and understanding the behaviour of a system it is useful to document this information into a log file and store it on a secure and accessible place. This way the members of the team can refer to the log anytime to determine if a certain behaviour is according to usual.
- **Perform Event Correlation**: it is useful to correlate indicators in different logs and sources to determine an incident.
- **Keep All Host Clocks Synchronized**: event correlation is difficult if the systems are not performing in the same time standard. Clocks of all systems has to be synchronized to make the identification of an incident and the correlation easier.
- **Maintain and Use a Knowledge Base of Information**: a simple structured knowledge can prove an effective tool in incident handling. Like a log repository a knowledge base can be helpful for the handlers when they want to refer to an old event or a detail for a certain incident of the past.
- **Use Internet Search Engines for Research**: internet engines can provide data about a particular issue that may have be encountered in the past in the community.

- **Run Packet Sniffers to Collect Additional Data**: due to the fact that the indicators do not record all the necessary data to realize an incident, packet sniffers can provide the team with additional data in order to make the situation clearer.
- **Filter The Data**: sometimes the amount of incoming data is very large and for this a categorization of data and indicators can happen in order to prioritize data that are more significant than other.
- **Seek Assistance from Others**: in case the team has not enough understanding or data about a particular incident it is important to seek advice in internal resources or external resources.

10.3.1.2.2 Incident prioritization

Incident prioritization is a part of the detection phase. It is important to have an understanding which incident are more important to be detected and handled first. Some can harm critically a system or an organization so a first-in, first-served approach is not optimal. Teams must define a way of prioritizing the incoming incidents. NIST SP 800-61 propose the prioritization of the incidents according to the following factors:

- **Functional Impact of the Incident**: the handlers should consider how much a certain incident affects the organizations functionality and its systems, to determine its severity.
- **Information Impact of the Incident**: due to the fact that some incidents affect the integrity of the information of a system, the handlers should decide the severity of the incident according to its damage to the organization's information.
- **Recoverability from the Incident**: depending on the harm of an incident does to the resources and the amount of time the organization can recover from it, is a major factor for the team to determine the severity of the incident and prioritize it accordingly.

Combining all these factors and the overall impact of an incident in the organization, as well as the possible recoverability of that incident the incident response team can decide which incident is to be detected and handled in which order.

10.3.1.2.3 Triage Function

This role aims to ensure that information en route to the incident handling service is directed through a single focal point for appropriate redistribution and management within the service, regardless of how it arrives at its destination (such as IDS, helpdesk, hotline, or email). By propagating the triage function as the single contact point for the entire incident handling service, this goal is achieved. The triage feature responds to new data in one or more of the subsequent ways:

1. The triage feature can need to request supplementary details in order to identify the incident.
2. If the occurrence is linked to a known vulnerability, the triage feature notifies the appropriate sections of the company and distributes information about how to minimize the vulnerability.
3. Each incident is identified by the triage function either as a new one or as a part of an ongoing incident and shares this information with the incident handling response function.

10.3.1.3 Respond

It is important to contain the incident before it performs severe damage into a system or an organization. Containment plans differ depending on the incident, although NIST SP 800-61 proposes the following criteria for the incident response team to determine the most suitable response strategy:

- Potential damage to and theft of resources
- Need for evidence preservation
- Service availability
- Time and resources needed to implement the strategy
- Effectiveness of the strategy
- Duration of the solution

In some cases, the attacker is redirected to a sandbox, in order its actions to be monitored. The incident response team has to examine this strategy with legal department for its

feasibility. It is not appropriate to use other monitoring methods of an attacker's activity due to the fact that the attacker can cause damage in the physical system. Also, some attacks may cause additional damage when they are contained so the handlers have to keep that in mind while choosing a containment strategy.

When a probable incident is exposed, documented procedures should exist in order to respond to it. Next there are listed possible response activities:

- Act to protect networks and systems damaged or threatened by adversary activity
- Granting solutions and strategies for mitigation from appropriate advisories or alerts
- Monitoring for intruder activity on other sections of the network
- Filtering network traffic
- Reconstructing systems
- Patching and fixing systems
- Establishing different workaround or response strategies

How to identify the source of the security incident must be detailed by the response procedures. The action taken must be described to bounce back from the incident in a way which minimizes the risk or the harm to the organization. Without any doubt it is impossible to specify each probable kind of incident. Despite this, the processes ought to identify common categories of these incidents and the appropriate procedure used to respond to them. Optimally, these ought to contain descriptions of probable incidents and common response actions. The management personnel that is responsible for the decisions that affect the systems of the organization has to be identified and be reachable at any given time in the event of an incident. This is especially relevant in situations like the mass e-mail worm infection we discussed, where the solution is a trade-off between major loss of functionality and further system compromise. These decisions would have a significant impact on the organization's activities and must be taken quickly. NIST SP 800-61 lists the types of security incidents that must be handled in incident response policies:

- Malicious code that infects a host
- Denial-of-Service (DoS) attacks that restrain or hinder normal use of systems

- Inappropriate usage of a system in violation of acceptable use policies
- Unauthorized access to a system
- Multiple-component incidents, that involve a number of the aforementioned categories in a single incident

There are several issues need to be considered in order to determine the appropriate responses to an incident. These involve the criticality of the system to an organization's function, and the present and probable technical effect of the incident regarding how much the system has been compromised.

The response protocols should identify the situations in which security violations must be reported to third parties, such as the police or a related CERT (computer emergency response team) agency. A high chance of variance exists among organizational attitudes in this kind of reports. These reports provide help to third parties to monitor the overall activity and tendencies in computer crimes. Nevertheless, especially in the event of a legal action could be established. It may be the organization's duty to collect and present relevant evidence. While the legislation may require reporting in some instances, there are several other forms of security incidents where the response is subjective. Therefore, it has to be decided beforehand whether such reports would be considered as suitable for an organization. When an event is reported externally, an incident may also be reported in the public media if it is reported externally. It is necessary to determine the organization's regular actions.

Evidence gathering about an incident is part of the response process. Initially, this information was intended to aid in the recovery from the incident. This evidence may be required for legal proceedings if an incident needs to be reported to the police. In this matter, it is essential to take meticulous steps to document the evidence collection, storage, and transfer. This has to be done according to the appropriate legal procedures, if not, there is a possibility the evidence will not be admissible in court. NIST SP 800-61 provides guidance on this matter.

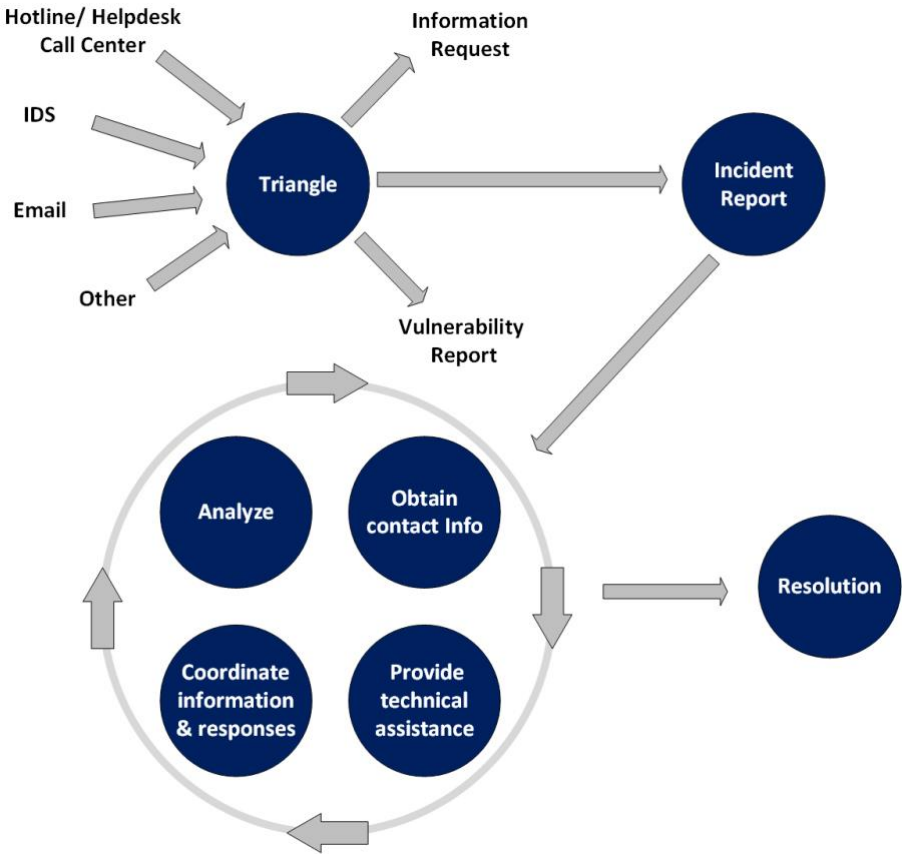


Figure 10-3: Incident-Handling Life Cycle [1]

In Figure 10-3 it is depicted a common incident-handling life cycle. The moment an incident is initiated, it progresses through several states, along with the incident related information, up to the point that all actions from the team's perspective are completed and the incident is terminated. On the lower left side of figure 10-3 they are indicated those states which probably will be visited more than once in the duration of the activity's life cycle.

10.3.1.4 Document and Post-Incident Activity

After the immediate response to an incident, in order to understand in which way to prevent the incident in the future, it is needed to identify what vulnerability of the system led to the manifestation of the incident. Recorded details of the incident and the way it was resolved are kept for future use. It is important to also reconsider the impact on the organization's systems and their risk profile as an outcome of the incident.

This usually entails feeding the information gathered as a result of the incident back to an earlier stage of the IT security management process. In some cases, the incident could be an isolated rare occurrence and appear out of bad chance. Normally, though, a security incident indicates an alternation in the organization's risk profile that needs to be confronted. This involves reviewing and possibly changing or extending the risk assessment analysis of the relevant systems. It entails reviewing controls for a variety of risks, reinforcing existing controls, and implementing new ones. The cyclic process of IT security management is reflected in this.

NIST SP 800-61 proposes the following questions to be asked in a post-incident meeting of the team for further development of the incident response plan:

- What was the exact incident, and when?
- How well did the personnel and management act in dealing with the incident? Were they act in accordance with the documented procedures? Were their actions satisfactory?
- What information was needed sooner?
- Were any steps or actions taken that might have inhibited the recovery?
- What would the staff and management do differently the next time a similar incident occurs?
- How could information sharing with other organizations have been improved?
- What corrective actions can prevent similar incidents in the future?
- What precursors or indicators should be watched for in the future to detect similar incidents?
- What additional tools or resources are needed to detect, analyze, and mitigate future incidents?

10.3.1.4.1 Collected data

The data to be documented in this phase of the incident response capability must be actionably and not only gathered because it is available. The data have to be understood and processed rather than be absolute numbers with no use or left to be processed. NIST SP 800-61 proposes possible metrics for incident related data:

- **Number of Incidents Handled**: to measure the number of handled incidents it is better to compare this number to the amount of work a team had to provide in order to respond to these incidents. Handling more incidents is not always better, because a more secure system can have fewer incidents.
- **Time per Incident**: time can be counted differently in each incident**:**
  - Time of work spend on the incident
  - Time between the initiation of the incident and the diagnosis of the incident
  - Time of response of the team to the first report of the incident
  - Time taken to report the incident to the management or other external parties
- **Objective Assessment of Each Incident:** it is useful to observe the response of an incident in order to define its effectiveness; this can happen with the following ways:
  - Reviewing the documents made of the incident to determine future response policies and plans.
  - Identify the effectiveness of the logging that took place during the incident.
  - Determine the damage caused by the incident prior to its detection.
  - Regulate the cause if the incident, identify its route and the vulnerabilities exploited by it.
  - Detect if the incident was a by-product or similar to a previous incident.
  - Calculate the damage caused by it to the organization.
  - Determine the measures that incident could have been prevented.
- **Subjective Assessment of Each Incident:** assessment of the team's performance by the team's members.

10.4 Coordination and Information Sharing

It is for the benefit of every organization to share information about the way they handle incidents in order to help others and itself to be up to date to any threat. Information to be shared is threats, attacks and incidents that occurred, vulnerabilities that the organization has spotted as well as ways the organization used to mitigate and handle these incidents and threats in the past. As mentioned earlier it is important to establish trust with third parties for the share of this information and establish a policy for which information is to be shared

and which is sensitive and needs to be kept inside the organization. Information sharing and coordination is a practise that benefits mutually all the organization that join this process.

10.4.1 Coordination overview

Coordination is taking place between organizations in order to be in a common ground on mitigating threats and handling incidents. These organizations may be also incident response teams, Internet service providers (ISPs), law enforcement agencies and costumers and constituents. Coordination can be efficient only if all the parties related understand their role and position and establish an efficient way of communication.

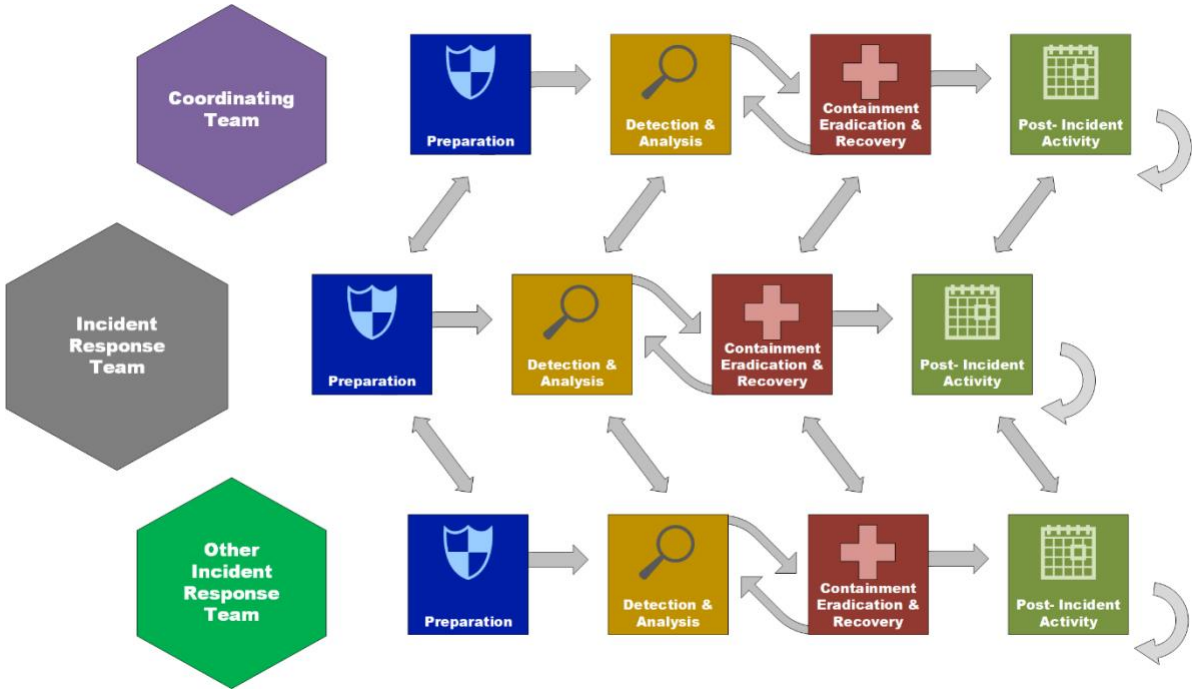


Figure 10-4: Incident Response Coordination

Coordination relationships according to NIST SP 800-61 can be:

- **Team-to-Team**: this relationship occurs when technical incident responders from separate organizations come into a collaboration with peers during a state of the incident handling lifecycle. The participants are often peers without any authority over each other. They share information and knowledge to solve common problems. The

information that is share more frequently is tactical and technical, but it also can be plans, procedures or lessons learned if necessary.

- **Team-to-coordinating team**: this relationship takes place between an organizational incident response team and a different organization which is acting in the role of main point for coordinated incident response and management (e.g., US-CERT, ISAC). In this kind of relationship reporting is necessary from the member organizations by the coordinating body. The type of information shared in this relationship is tactical, technical and information regarding vulnerabilities, threats and risks to the community provided by the coordinating team.
- **Coordinating team-to-coordinating team**: these relationships emerge between multiple coordinating teams and exist to share information regarding incidents that affect multiple communities. The parties-coordinating teams act on behalf of the community member organizations and distribute information on the sphere of cross-cutting incidents to help in inter-community response. The kind of information that is being distributed in this relationship usually consists of regular analysis during "steady state" operations, response plans and impact or risk assessment information during coordinated incident response activities.

Building this kind of relationships can be a difficult task for the organizations, some of them are mandatory to be built and others voluntary, in any case effort is needed from each party in order to establish a legit coordination relationship for mutual benefit for all the participants.

10.4.2 Information sharing techniques

Information sharing is very important factor in order coordination across organizations to be facilitated. Methods of information sharing are:

- **Ad Hoc**: ad hoc methods are a usual way to share information. These methods include instant messaging clients, email and phone. Ad hoc information sharing mechanisms usually rely on a single employee's connection with others in incident response teams of partner organizations. This type of connections is used by employees to manually share information. The size of the organizations has a role in defining these ad hoc

techniques in terms of cost-effective way to share information. Although, this method is informal, and it is not possible to guarantee that the process will be always functional. Ad hoc methods are understadarized regarding what information is shared and in which way that communication occurs. That lack of standardization tends to require manual innervation from the participants and makes these methods resource-intensive to process than the alternative partially automated methods.

- **Partially Automated**: organizations try to automate as much as possible the information sharing procedures to make cross-organizational coordination more efficient and cost-effective. Full automation is difficult to be performed in reality and also not optimal due to the fact that may be more insecure and less effective. This creates security considerations that the incident response team should consider when planning their information sharing. Engineers need to plan which information has to be shared through this automated information sharing. Organizations could also decide to create a formal data dictionary listing all entities as well as their relationships, that they want to be shared. Also, machine-processable models to capture this information have to be constructed. Data exchange standards have to be used wherever possible. Except choosing which data exchange models are suitable for sharing incident information, an organisation has to work in collaboration with its partner organization and agree on a type of transport mechanisms in order information exchange to occur in an automated fashion.

10.4.3 Granular Information Sharing

Information sharing can come with drawbacks of sharing sensitive information, so the organization has to come to the sharing only the important and necessary information. Incident information can be separated into two types: business impact information and technical information.

- **Business impact information**: it describes the ways an incident affects the organization regarding financial impact, mission impact, etc. This information is needed in order the coordinating teams make decisions about the degree of assistance to provide to the reporting organization or make decisions comparative to

how an incident affects additional organizations in the community. This kind of information is useful to be reported only to organizations that have an interest to ensure the goal of the organization encountering the incident and it should be avoided to be shared to other parties unnecessarily.

- **Technical Information**: technical indicators of an incident can be of many types and they derive from a diversity of technical information associated with incidents. Organizations collect their own indicators, although, by mutual sharing of indicators with other partner organizations, they can enrich their knowledge of incidents with more information and data. Another organization's experience can be used as help for the organization in case the same incident occurs in the future. But organizations have to be aware for which data are to be shared and which are confidential. Technical indicator data is useful when they lead to the identification of an incident. Although, some indicator data received from other sources will create false assumptions and may lead resources to be spend on nonexistent issues. The personnel responsible of the analysis and sharing this kind of data has to be skilled in order to identify the significance of the indicators.

10.5 Referencing

[1] "Computer Security: Principles and Practice", 4/e, by William Stallings and Lawrie Brown, 2018

[33] Scarfone, K. A., Grance, T., & Masone, K. (2008). Sp 800-61 rev. 1. computer security incident handling guide. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

11 Security Operations: Security Assessment and Testing

Author(s): Enrique Costa-Montenegro
René Lastra Cid

11.1 Introduction

11.1.1 Security Assessment

Security Assessment is defined as the process to evaluate, identify, quantify, and prioritize the security issues (policies, procedures and technical protections) in a system.

To provide direction and guidance to security assessors, organizations must develop information security assessment policies. Assessment requirements can be identified and assessments can be ensured to meet established requirements by implementing this security policy. This policy should be distributed to all employees as well as third-party assessment providers. This policy should be reviewed at least once a year [34].

11.1.1.1 Why do we need an assessment?

The purpose of a security assessment is to ensure that the required security controls are integrated into the design and implementation of a project by providing documentation that describes any security gaps between a project's design and approved corporate security policies.

The security assessment should identify potential security issues. It is advisable to prioritize safety issues and send a report to management.

11.1.1.2 Security Assessment Pros and Cons

Security assessments have the following Pros and Cons.

Pros:

- Oriented to technical and non-technical issues
- Has a defined scope: which systems and what potential problems are evaluated
- Identifies most common technical issues
- Repeatable and quantitative

Cons:

- Difficult to well

- It can identify a lot of issues
- Just recommend best practices

11.1.2 Security Assessment: test, assessments and audits

11.1.2.1 Assessment and Testing Program

An information security team's key function is security testing and assessment. This assessment and testing program includes audits and assessments that are used to ensure the security of an organization's information properties on a regular basis. It has three major components:

- Security tests are used to ensure that a control is functioning correctly.
- Security assessments are in-depth examinations of the system's security.
- Security audits are similar to security assessments, but they are conducted by a third-party auditor.

11.1.2.1.1 Security tests

Security testing is a type of testing technique used to see how an information system protects data and functions as anticipated. While security tests do not guarantee the system's complete security, it is necessary to include them as part of the testing process. The following six measures are used in security tests to create a safe environment.[35].

- Confidentiality prevents material from being disclosed to unauthorized users.
- Integrity requires true and relevant information to be transferred from senders to planned recipients.
- Authentication verifies and checks the user's identification.
- Authorization defines the users' and services' access privileges.
- Availability guarantees that information is available as needed.
- Non-repudiation means that neither the sender nor the recipient denies sending or receiving the letter.

Security testing verifies that a system is functioning properly including automated scans, penetration tests, and manual attempts to undermine security. A comprehensive evaluation and testing strategy must be designed and validated. It should be done on a regular schedule frequent automated testing with infrequent manual testing. Results are for internal use.

11.1.2.1.2 Security assessments

Security assessments consist of review of the security of a system, identify vulnerabilities in the tested system and make recommendations for remediation. The use of security testing tools is also included. A careful review of the threat environment, current and future risks needs to be included in the security assessment. The result is an assessment report addressed to the management in a non-technical language with recommendations to improve the security of the system

11.1.2.1.3 Security audits

Security audits are evaluations to demonstrate the effectiveness of controls. These audits should be performed by independent auditors to eliminate bias. There are two types:

- Internal audits: by internal personnel for internal audiences
- External audits: by an external company with a high degree of validity because there is no conflict of interest

A security audit is a comprehensive examination of an organization's IT security procedures, both physical and nonphysical (software). Vulnerability scans are used to find gaps in operating systems, and penetration tests are used to obtain unauthorized access. Furthermore, after completing all of the necessary procedures, the reports are submitted to the organization for further review. Physical access to physical hardware for authentication and other logistical problems is also part of a computer security audit [35].

11.2 Assessment and Elements

11.2.1 Vulnerability Scans & Penetration Testing

11.2.1.1 Vulnerability Scans

The frequency and comprehensiveness of vulnerability scans are determined by the information system's protection classification. Vulnerability detection for custom software and applications may necessitate the use of additional, advanced methods and techniques (e.g., web-based application scanners, source code reviews, source code analysers). Vulnerability scanning looks for particular features, ports, protocols, and utilities that may not be open to users or applications, as well as poorly designed or running knowledge flow mechanisms. [36].

The goal is to probe systems, software, and networks for vulnerabilities that an attacker might exploit, using scanning tools that include quick tests, planned scanning, and detailed reporting to administrators. There are three main categories:

- Network Discovery Scans
- Network Vulnerability Scans
- Web Application Vulnerability Scans

11.2.1.1.1 Network Discovery scans

- SYN Scanning (half-open scanning) on TCP: sends packet with SYN flag set (request to open a new connection). If you get an answer with the SYN and ACK flags set, it means the port is open.
- TCP Connect Scanning establishes a complete link with the remote device on the designated port.
- TCP ACK Scanning: sends packets with the ACK flag set to indicate that they are part of an open link.
- Xmas Scanning: sends a packet with the PSH, FIN, and URG flags package.

11.2.1.1.2 Network Vulnerability scan

Vulnerability scanners may be used locally or remotely to search a host for vulnerabilities. Some network-based scanners provide administrator-level privileges on specific hosts and can use those credentials to retrieve vulnerability data from those hosts. Other network-

based scanners don't have those keys, so they have to focus on network scanning to find hosts, then scanning certain hosts for vulnerabilities. In such cases, network-based scanning is mainly used to do network exploration and locate open ports and associated vulnerabilities. Internal and external network-based scanning without host credentials may be done, and although internal scanning normally uncovers more vulnerabilities than external scanning, checking from all perspectives is essential. External scanning must deal with traffic-blocking perimeter control systems, restricting assessors to scanning only the ports allowed to move traffic [37].

Not only do scans look for open ports, but they also look for suspected vulnerabilities. Unauthenticated scans put the device to the test without the use of codes or any special data. It is suggested that authenticated scans be used to enhance screening.

11.2.1.1.3 Web Vulnerability scan

Special tools are used to check web applications for known vulnerabilities. It is important to conduct a web vulnerability scan because web pages are the connection point of organizations with customers. Always scan new website before moving into production, it is also recommended to scan the modified website before code changes moved into production. It is important to use scheduled scans to scan all applications

11.2.1.1.4 Penetration Testing

Penetration testing can be used in conjunction with vulnerability testing. There are various methods with the same aim of finding and assessing security vulnerabilities. Penetration testing is a more involved procedure that involves manual testing and exploitation by a security expert to mimic the actions of a specific attacker. As a result, the risk faced by various vulnerabilities will be more accurately assessed.

Penetration testing is more than just a security scan; it tries to exploit devices. Penetration testing target a device and attempt to gain entry, while vulnerability scans search for the existence of vulnerabilities but do not take aggressive action.

11.2.2 Code Review and Testing

Software is an important part of system security. Privilege access to the operating system, hardware, and other services is common in software applications. Furthermore, software applications manage sensitive data such as credit card numbers, social security numbers, and confidential company records, as well as files containing sensitive data.

Code review and testing can help identify security, functionality, and compatibility vulnerabilities in software before they go live.

Most systematic code review procedures have a several step review and testing process: Planning, Overview, Preparation, Inspection, Rework, Follow-up.

11.2.2.1 Static testing

Static testing is a form of software testing in which the software is checked without having to run the code. The use of automatic tools to identify basic program bugs such as buffer overflows is involved. It is divided into two parts, as follows:

- Review: A technique for locating and correcting mistakes or ambiguities in documentation such as specifications, designs, and test cases.
- Static analysis: Developers' code is examined (usually using tools) for design flaws that might lead to defects.

11.2.2.2 Dynamic testing

Dynamic Testing is a form of software testing methodology that examines the code's dynamic behaviour. For dynamic testing, the program should be compiled and run, and parameters like memory utilization, CPU usage, response time, and overall software output should be examined [35].

Dynamic testing entails putting the program to the test for input values and analyzing results. Dynamic testing assesses the software's protection in a running environment. Any difference between the real and planned outcomes may indicate a bug in the code, which should be explored further.

11.2.2.3 Synthetic transactions

Synthetics don't monitor individual user sessions; instead, they track programmed transactions that mimic a user's behaviour. RUM (Real User Monitoring) is a form of passive site surveillance that relies on utilities that continuously monitor the device in the background, tracking availability, accessibility, and responsiveness.

- Capturing and analyzing each user's transaction on a website or application (passive monitoring).
- Collecting server-side data from the bottom up
- Capturing user interface from the top down

11.2.2.4 Structural testing

Structural testing (also known as glass-box or white-box testing) is a methodology in which measurements are extracted from experience of the software's internal configuration or execution. Clear-box testing, open-box testing, logic-driven testing, and path-driven testing are all terms used to describe structural testing.

Structural testing identifies test cases from examining the source code. Can identify “Dead Code”, that is, code that is never executed during program execution. It uses metrics to show the percentage of software structure that has been evaluated, usually referred to as “coverage”. 100% coverage means each program piece has been executed at least once. Some of the coverage types are statement, decision, condition, multi-condition, loop, path, data flow, branch coverage, etc.

11.2.2.5 Fuzzy testing

Fuzz testing is a software testing procedure in which the device is given random data as inputs. If the program fails, the system must resolve the problems/defects. Inputs that are unexpected or random will produce unexpected outcomes [35].

It is a specialized dynamic testing technique that uses random malformed data as input to software program to stress its limits and determine if it will crash or break finding previously undetected flaws.

It is limited to detecting simple vulnerabilities and uses invalid input to the software, either randomly generated or specially crafted to trigger known software vulnerabilities.

The key goal is to keep an eye on the application's output, looking for things like program crashes, buffer overflows, and other unwanted and/or unexpected outcomes.

There are two main categories: Mutation (Dumb) Fuzzing and Generational (Intelligent) Fuzzing.

11.2.2.6 Interface testing

Interface testing is used to see whether programs or components are communicating data and commands properly. It's to make sure that all of these modules' connections are running well and that bugs are treated properly. It entails several development teams operating on various aspects of a program that may communicate with one another. As a result, it's important to have well-defined interfaces.

When all implementation activities are complete, interface testing compares the performance of modules to the interface requirements to ensure that they can fit together properly.

There are three categories of interfaces that should be evaluated:

- Application Programming Interfaces (API): standardized way for code modules to interact and can be exposed to the outside world through web services.
- User interfaces (UI): graphical user (GUI), command line (CLI) and web interfaces, which provide end users with the ability to interact with the software.
- Some programs that handle machinery, logic controllers, or other objects in the real world use physical interfaces.

11.2.2.7 Misuse case testing

Misuse case testing, also known as abuse case testing, assesses the software's susceptibility to known threats. Ensures that an application can accommodate unexpected user behaviour or invalid input. Exceptions are expected such as:

- Required form fields: Populating Required Fields (leave empty)
- Data mismatch: Correspondence between Data and Field Types (enter incorrect data types)
- Field limits: Allowed Numbers or Characters (too many/too few)
- Data bounds: Allowed Data Bounds and Limits (exceed range)
- Reasonable Data (invalid data)
- Unauthenticated pages: Web Session Testing (open web pages without logging in)

11.2.2.8 Test Coverage Analysis

Code coverage is a software testing metric or also termed as a Code Coverage Testing which helps in determining how much code of the source is tested which helps in accessing quality of test suite and analysing how comprehensively a software is verified. Currently, in simple code coverage refers to the degree of which the source code of the software code has been tested. This Code Coverage is considered as one of a form of white box testing. It is impossible to completely test any piece of software, too many ways that software might malfunction or undergo attack. It is necessary to do an analysis of the test coverage to estimate the degree of the tests performed with the new software, with the formula:

Test coverage = number of use cases tested / total number of use cases

Enumerate all the possible use cases is an exceptionally difficult task, so result may not be right.

11.2.2.9 Software maintenance tasks

Among the software maintenance tasks we find the review of the software validation plan, anomaly evaluation, problem identification and resolution tracking, proposed change assessment, task iteration and documentation updating.

11.2.3 Log, Account Managements & Backup Reviews

11.2.3.1 Log reviews

Log review decides whether the security controls are capturing the correct data and whether the company is adhering to its log management policies. Audit logs may be used to verify that the system is running in compliance with the policies in place. For example, if the logging policy mandates that all authentication attempts on sensitive servers be logged, the log analysis will decide if this data is being collected and displayed at the required level of detail. If the logging policy mandates that any authentication attempts to sensitive servers be logged, the log analysis will decide when this information is being obtained and at what level of detail.

Unauthorized accesses, attempted intrusions, and misconfigured services and security measures can all be revealed by reviewing logs. If an intrusion detection system (IDS) sensor is installed behind a firewall, for example, the logs from the sensor can be used to investigate communications that the firewall allows into the network. If the sensor detects events that should be blocked, it means the firewall isn't properly installed [37].

It is advisable to periodically conduct log reviews, particularly sensitive functions, to ensure that all processes are working properly. User records should be reviewed to see if they agree to their privileges. It is an important procedure in the event of a security incident.

- Phases:
    - Configuration of the log sources
    - Performing log analysis
    - Initiating responses to identified events
    - Management of long-term storage of logs

11.2.3.2 Account Management reviews

It is recommended to check that users retain authorized permissions and that unauthorized modifications do not occur. In addition, a full review of all accounts is recommended. Some accounts must be highly privileged to do so.

A good practice is to compare the list of users with privileged access and privileged access rights with the list of system in order to know if any user has accessed any element without the privileges needed.

Compromising privileged users of the system is the attackers' preferred technique. Three ways to accomplish:

- Compromise an existing privileged account
- Create a new privileged account
- Elevate the privileges of a regular user account

11.2.3.3 Backup reviews

Checking copies to ensure that the procedure works well and meets the organization's data security requirements is a common practice. Reviewing logs, checking hash values, or demanding a device or file restore are both examples of backup feedback.

11.3 Penetration testing

Penetration testing is a security procedure to determine if a system can withstand an intrusion attempt from an attacker.

Penetration testing is a form of security testing that examines an application's vulnerability and seeks out any security risks that might exist in the system. Any intruder will interrupt or gain unauthorized access to a device if it is not safe. Configuration errors, programming errors, and software glitches, for example, are common security risks that arise during the development and execution of software.

Penetration testing assesses a system's ability to defend networks, applications, endpoints, and users from external and internal attacks. It also makes an effort to protect security controls and ensure that only approved users have access.

Penetration test assesses how an attacker can target the device using a white hat attack in a simulated environment, assisting in the discovery of vulnerable points that an intruder can attack. It also preserves the initial data and aims to deter black hat attacks. Estimates the size of the assault on a future business. Provides data to support that it is important to expand investments in technology protection.

How is done:

- Basic reconnaissance of the system
- Network exploration scans to find available ports
- Network security tests to find bugs that haven't been fixed
- Web application vulnerability scans to identify web application flaws
- Use of exploit tools to automatically attempt to defeat the system security
- Manual probing and attack attempts

11.3.1 Types

The method of penetration testing used is usually determined by the complexity of the project as well as the needs and specifications of the organisation. Pen testing is another name for it [35].

White Box Penetration: White Box Penetration: This is a thorough evaluation in which the tester is given a wide variety of information about the systems and/or network, such as Schema, Source code, OS specifics, IP address. It's usually thought of as a mock attack by an internal source. Structural testing is also known as glass box, transparent box, and open box testing. White box penetration testing looks at the code coverage and performs data flow, path, and loop testing, among other things.

- Provides attackers with detailed information about system
- Bypasses many of the reconnaissance steps

Grey Box Penetration: In this method of testing, a tester normally only offers a small amount of knowledge about a system's internal details. It can be thought of as an external hacker gaining unauthorized access to an organization's network infrastructure records.

- Partial knowledge test: balance advantages and disadvantages of white and black box penetration tests
- Results of black box results but costs or time of white box

Black Box Penetration: In black box penetration testing, the tester has no prior knowledge of the applications he will be testing. The tester wants to learn as much as possible about the goal network or device. In this testing, for example, a tester only knows what the desired result should be, but not how it can be achieved. He doesn't look at any of the programming codes.

- Provides little transparency to attackers
- Acts as though an unknown intruder is attempting to obtain entry.

Blind Tests

By blind tests, the tester only has publicly available data to work with. The network security team has prior knowledge of this test to defend.

Double Blind Types

It is a blind test to both the tester as well as the security team. It is used to evaluate the security levels and responses of the security team. It is a realistic demonstration of the likely success or failure of an attack.

Targeted

Involves external and internal parties carrying out a focused test on specific areas of interest.

11.3.2 Phases

Penetration testing is a collection of procedures that considers a variety of device problems and performs evaluations, evaluations, and recommendations. It is based on a step-by-step protocol for performing penetration testing.

- **Discovery or reconnaissance:** Footprinting and gathering information about target.
- The preliminary information is analyzed as part of **reconnaissance**. Mostly, a tester doesn't have any knowledge other than the basics, such as an IP address or a block of IP addresses. The tester begins by analyzing the available data and, if necessary, requesting additional information from the customer, such as device descriptions, network schedules, and so on. This move is a kind of passive penetration test. The only goal is to get a full and accurate picture of the structures.
- **Discovery**: A penetration tester would most likely use automated tools to search target properties for vulnerabilities in this phase. These tools usually have their own files that provide information about the most recent vulnerabilities.
- **Network Discovery**: Additional networks, servers, and other computers can be discovered by network discovery.
- **Host Discovery**: Host Discovery: It determines which systems have available ports.
- **Service Interrogation**: It interrogates ports to find out what utilities are working on them.

The most common tool used for network discovery scan is nmap.

Enumeration: scanning and probing. Scanning or probing is the process of collecting information about computers by either listening to network traffic or sending traffic and observing what traffic returns as a result. The exploration and probing is usually done through ping sweeps, port scanning, grabbing banners, vulnerability scanning, etc.

Once a target has been identified, enumeration is the process of identifying what resources are publicly available for exploit.

Both methods must be used in conjunction with each other. You first scan the network to determine what assets or targets are on the network, and then you enumerate each target by determining which of its resources are available. Without knowing which computers and resources are vulnerable, it is impossible to protect these resources from attack.

For the enumeration a tool such as Nessus or OpenVAS is used to find vulnerabilities. Requires manual verification and assessment and must match actual threats to find true risk.

Main features:

- Uses a tool like Nessus or OpenVAS
- Finds vulnerabilities
- Requires manual verification and assessment
- Must be matched to real threats to find true risk

Exploitation

The primary purpose of the exploitation process of a penetration test is to gain access to a device or resource by circumventing security restrictions. The key goal is to find the main access point into the enterprise as well as high-value target properties.

Therefore, in the exploitation phase, an attempt is made to obtain unauthorized access by exploiting the vulnerabilities. Uses a tool like Metasploit.

Post-exploitation

Post-exploitation is an underappreciated but crucial part of the advanced penetration testing process. Advanced penetration testers, also known as ethical hackers, are typically tasked with a high-profile goal that requires advanced techniques to obtain the required degree of access.

The main goal of post–exploitation is to determine the base value and capacities of the compromised system/device of the victims, as well as to obtain access to all parts of the targeted networks without being identified.

Once the system is compromised, try to access as much information as possible and use knowledge of the operating system.

Report

Normally two reports are made, an executive level report, with conclusions and without technical details, and on the other hand, a technical report to drive remediation efforts.

Report writing is a broad role in penetration testing that encompasses methods, processes, a thorough description of research content and design, a concise example of a qualitative study, and the tester's personal experience.

The report should be the application of a standard methodology to the specific system of study. The key elements of a good technical report are:

- Threats
- Vulnerabilities
- Probability of exploitation
- Impact
- Recommended actions

Social engineering

The aim of social engineering is to persuade someone to disclose sensitive information that can be used to target systems or networks. It may expose flaws in user behavior, such as failing to obey standard procedures, and is used to measure the human factor and user knowledge of security. Social engineering can be done in a variety of ways, including analogue (e.g., in-person or over-the-phone conversations) and digital (e.g., via social media) (e.g., e-mail, instant messaging) [37].

Types:

- **Phishing**: the most popular form of social engineering attack conducted through digital communication
- **Spear Phishing**: a type of phishing attack that is targeted to a specific group or individual
- **Whaling**: specific phishing attack targeting senior executives or individuals
- **Drive by download**: invisibly redirect the user to malicious distribution server; it is an automatic attack that is triggered simply by visiting a malicious website
- **Pretexting**: social engineering attack over phone.

11.4 Referencing

[34] Bahga, A., & Madisetti, V. (2014). *Internet of Things: A hands-on approach*. Vpt.

[35] NIST Special Publication 800-15, Technical Guide to Information Security Testing And Assessment.

[36] tutorialspoint, «Learn Software Testing terms,» [En línea]. Available: https://www.tutorialspoint.com/software_testing_dictionary/dynamic_testing.htm.

[37] L. Johnson, Security Controls Evaluation, Testing, and Assessment Handbool, Elsevier, 2016.

12 Software Development Security

Author(s): Felipe Gil Castiñeira
Cristina López Bravo
René Lastra Cid

12.1 Basic Principles of Secure Development

Since several information security incidents today include software vulnerabilities of some form or another, the security professional must ensure that the enterprise security infrastructure includes applications.

The majority of significant accidents are caused by software flaws. Both the applications and code used, as well as the protocols and interfaces involved, are becoming much more standardised. Although this has advantages in terms of preparation and efficiency, it also means that a problematic function may have a large impact on computing and business environments. Furthermore, legacy code and design decisions made decades ago are frequently still present in today's applications, interacting with emerging technology and operations in ways that can expose additional vulnerabilities that the security professional is unaware of [38].

The security professional must understand key security principles that relate to software creation, service, and maintenance.

Early in the risk assessment process, software should be considered an asset that should be evaluated. It should also be considered as a tool with vulnerabilities that can necessitate mitigation or system-specific controls and safeguards.

The security professional must be able to grasp the impact imperfect software can have on the enterprise if it is exploited by a threat or misused by an untrained employee. He or she must be able to translate such impacts into action plans directed at prevention and remediation of these.

The security professional must be able to see themselves as a team member working alongside developers, QA personnel and operations staff to improve risk management, rapid response, better integration, and smoother operations in order to create a more productive and a more secure infrastructure.

In the security of software environments, resources must be available when and where they are needed by authorised users, the integrity of the processing of the data and the data itself must be guaranteed and, where necessary, the confidentiality of the data must be protected.

The current software environment is much more distributed than it has been at any time in the past. This is due in part to a substantial increase in open protocols, interfaces, and the supplies of source code. This results in increased sharing of resources that will also require increased protection, during the sharing relationships and due to the widespread nature of the teams involved, and thus it proves to be a much more complex and potentially much more difficult environment to manage.

The idea behind security in the architecture is that designing it in is much more cost-effective over the long term, than adding it on piecemeal later. It has been shown by many studies over the years that a dollar, for example, spent in the design phase that eventually got built into the program can equate to as much as a 150 dollar in add-on break-fix by the time the system is operationally ready.

Overall, this will prove a drastic improvement in quality and productivity. Re-use, of course, is very good because it saves time and it saves money, and the architectural process in this is intended to be focused on the ability to re-use.

The relationship of security to the system lifecycle basically begins at the beginning. Our infrastructure engineering starts off the project and that runs through all the different phases. It's an iterative cycle, as we all know, with functional and participative overlap, so that an integrated and re-iterative process is involved at every step, all the way through to production operations, and information assurance needs to be involved from the beginning, whether it's informing the design process, informing the implementation process, or the operation, it needs to be involved every step of the way, because the object is to integrate the information assurance and risk assessment processes with the security engineering functions and do so into every phase of the system lifecycle and any IT project intended for production operations, regardless of the development model being employed [38].

12.1.1 Identifying Vulnerabilities and Security Problems

Vulnerabilities and security problems may affect a software product during the whole development process if [39]:

- Security requirements are not described during the analysis stage.

- Designs created have security failures.
- Vulnerabilities are created during the implementation stage.
- The software is deployed inappropriately.
- Occurred security incidents have not a proper response.

Such problems directly affect the developed software and information stored, but it may also affect:

- Other applications that are executed in the shared environment.
- The user's system.
- Other systems that interact with the software to develop.

12.1.2 Secure Principles of Secure Development

The Secure Software Development Life Cycle (S-SDLC) [39] is the software development process that implements security as a transversal element during the whole development cycle. The implementation of security as a transversal element of software development is called “Security by Design”.

Vulnerabilities often provide a way into networks, often on a very deep level. The majority of major accidents and outages can be found to be caused by software bugs when investigated. Every new version of software adds to the size and complexity of the previous one. Furthermore, software is becoming much more structured, both in terms of the programs and code that are used, as well as the protocols and interfaces that are used. Most vulnerabilities may be solved in the application development stage, since many of such vulnerabilities are created when development processes and their associated controls are not implemented properly.

The **S-SDLC**:

- It takes into consideration all the **security aspects** that may be involved **in the software development** from the beginning of the process.
- It allows developers to **detect vulnerabilities** during early stages of development.
- It **saves costs** in vulnerabilities detected in systems that are in production.

- It allows developers to **take into consideration requirements according to different regulations and standards** from the beginning of the development process.
- It **saves costs in the implementation** of new requirements or functionalities related to compliance.

There are processes similar to the S-SDLC:

- OWASP CLASP (Comprehensive, Lightweight Application Security Process): https://owasp.org/
- Microsoft Secure Development Life Cycle: https://www.microsoft.com/en-us/sdl/
- Digital's Security Touchpoints (Developed by Gary McGraw): http://www.swsec.com/resources/touchpoints/
- NIST 800-64 (Set of security considerations suggested by the NIST that should be taken into consideration during the SDLC): http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-64r2.pdf

Generally, all the models include a series of security activities for the development life cycle. Probably the oldest development model is the **Waterfall.** It has a lot of variants that have been developed since. What characterizes the **Waterfall and its variants** is that each phase contains a list of activities that must be performed and documented before the phase begins. The importance of this particular point is there's a lot of rigor in how the process is formed and it places great importance on the documentation and that everything follows the documentation. Each method follows a form of rigorous sequencing of tasks in a finish-to-start project management sort of form.

These methods are adequate for long term, long duration projects, but tend to respond very poorly to changes and the variability that might take place over the long period that the program is developing. It is a **Structured Programming Development**. There you see the graphic that illustrates Figure 12-1 that, where one thing leads to another in a very rigorous formed pathway.

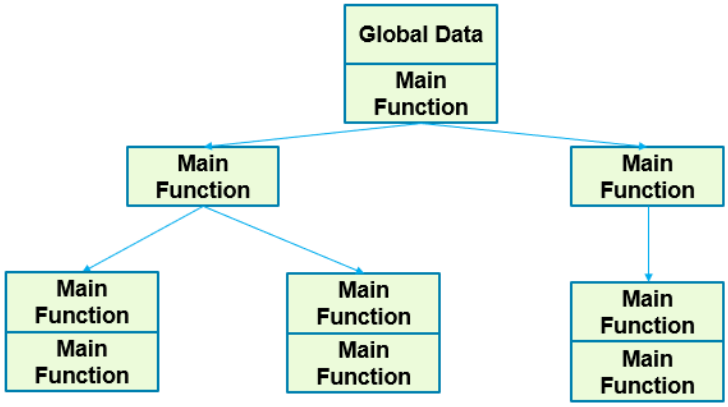


Table 12-1: Waterfall SDLC [40]

Table 12-1: Waterfall SDLC Example [40]

Requirements definition	Application Security Requirements
Architecture and Design	Application Security Architecture and/or Threat Model
Development	Secure coding Practices
	Security Testing
	Security Code Review
Test	Penetration Testing
Deployment	Secure Configuration Management
	Secure Deployment
Maintenance	

Another approach is the **V Model**, which is an **extension of the Waterfall** and is based on the association of a testing phase for each corresponding phase in the development cycle. In Figure 12-2, we have down the left-hand side of the V requirements analysis, system design, architecture design, module design, leading at the bottom of the V to coding. On the right-hand side as we go up, we have unit testing, integration, system testing, and acceptance

testing. In the valley of the V there is, from the bottom to the top, unit testing, integration testing design, system test design and acceptance test design, with arrows pointing to each leg of the V. It means that **for each single phase in the development cycle there is a directly associated testing phase.**

This is a **highly disciplined model** and the next phase starts only after completion of the previous phase. So again, it emphasizes the finish-to-start project management style.

Now, the corresponding **testing phase** of the development phase is planned in parallel. So, marking a line from, at the top, requirements analysis through acceptance test design to acceptance testing, there is a line that goes directly across the top.

So, there is a **verification phase** on one side of the V and **validation phases** on the other side. The coding phase joins the two sides of the V Model. So, as rigorous as this is, it doesn't respond very well to change.

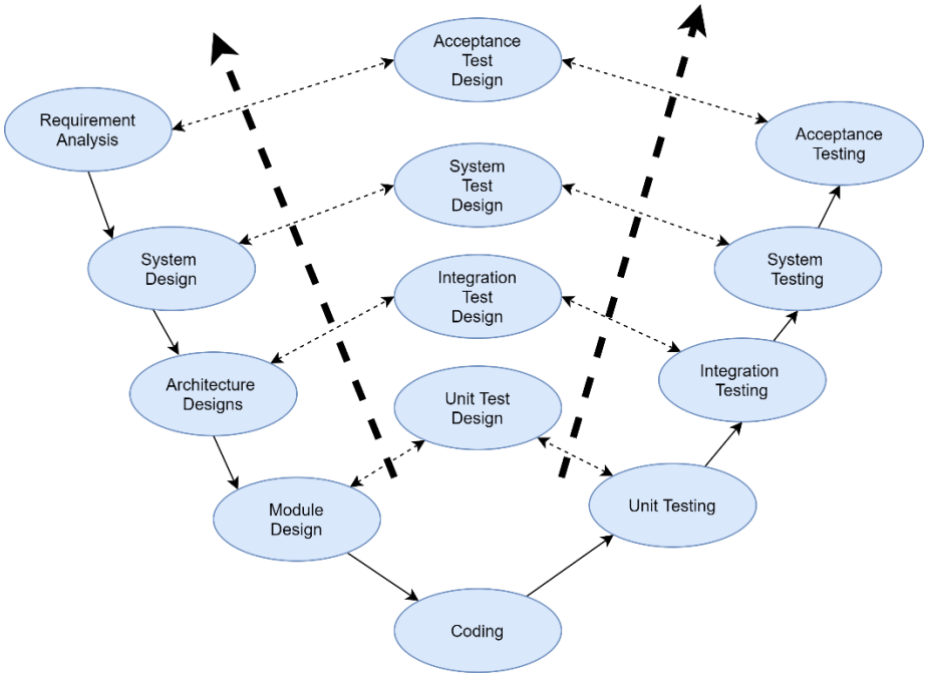


Figure 12-2: The V Model [40]

The **V Model** has **pros and cons**. The **advantage** is that it's highly disciplined and phases are completed sequentially. It works well for smaller projects where requirements are very well

understood. It is simple and easy to understand and use. It's easy to manage due to the rigidity of the model with each phase having specific deliverables and a review process. Verification and validation activities are included in each step on each leg of the V.

However, these very strengths lead to certain **disadvantages**. It tends to make the model very inflexible with a very high cost associated with change. There's high risk and uncertainty brought about by any change that is introduced. It is not a good model for complex or object-oriented projects. And it's a poor model for long or on-going. Furthermore, it is not suitable for projects where requirements are at a moderate to high level of risk of changing and once the application is in the testing phase, it is very difficult to go back and change a functionality, meaning a very high cost in break fix or modification. And no working software is produced until very late in the life cycle.

Now, to overcome those kinds of disadvantages, iterative development techniques were developed beginning with **prototyping** or **modified prototyping models**, **rapid application development**, **joint analysis development** and then, **exploratory models**.

And most of these iterative development types, they're based around the notion of putting together something very quickly as a proof of concept and then once the proof of concept demonstrates that the idea is sound and that the basics are good, it continues to evolve and produce additional prototypes until the final prototype is production ready.

12.1.3 S-SDLC in Agile Environments

Agile methodologies are an alternative process to traditional methodologies that are based on the development through smaller iterations used to include functionality (http://agilemethodology.org). Tasks and activities normally established by the S-SDLC assume the traditional life cycle (**waterfall**) during the software development.

The execution of S-SDLC, as we have studied, requires adaptations to be applied on agile methodologies. In such cases, the S-SDLC activities are executed with three different frequencies:

- By sprint: activities that should be executed for each completed release.
- By bucket: activities that should be executed for each set of sprints.

- By project: activities that are executed only once during the whole project.

➢ **Activities by sprint:**

- Threat modelling of the functionality included on the sprint.
- All the activities related to the S-SDLC implementation stage.
- Final security review by sprint.
- Certification and storage.

➢ **Activities by bucket:**

- Definition of the security metrics that will be used to assess the bucket.
- Dynamic analysis, fuzzing and review of the attack surface tasks.

➢ **Activities by project:**

- Define the security requirements.
- Risk analysis.
- Define the attack surface.
- Create an incident response plan.

Table 12-2: Agile Security Methodology Example [40]

Planning	Identify Security State Holder Stories
	Identify Security Controls
	Identify Security Tests Cases
Sprints	Secure coding
	Security Tests Cases
	Peer Review with Security
Deployment	Security Verification (with Penetration Testing and Security Code Review)

12.1.4 Stages of the S-SDLC

The SDLC identifies the various phases of the development process. Users can see and understand what processes are involved within a given phase by using the life cycle method. It's often used to let them know that steps can be replicated or a previous step reworked at any time if the system needs to be modified or improved.

- **Training**

It is not an S-SDLC stage strictly, but **it is essential to perform it**. **The technical staff** involved in the development of the project have to be able to **perform all the additional tasks that the S-SDLC implies**. They should be aware of the following **concepts:**

- Secure architectures
- Threat Modelling
- Secure encryption
- Penetration testing
- Security and privacy practices

This stage is essential for the different roles involved in a development process to **know** their **responsibilities from the point of view of security**.

- **Requirements**

During this stage, apart from traditional functional requirements of the application, further security, **privacy and regulatory requirements should also be taken into consideration**. A set of minimum security requirements should be defined. As in the case of the rest of requirements, it is important to implement the necessary measures to follow its development during the SDLC.

Another possible mechanism would imply defining a **set of security metrics** that should be maintained during all the development stages:

- Establish security levels for vulnerabilities.
- Explain the acceptable maximum levels for each development stage.

  E.g.: a product cannot pass the launch stage if it includes any critical vulnerability.

In order to facilitate the **identification of requirements**, the following processes are carried out:

- Identification of roles, capabilities and resources of the application.
- Risk analysis.
- Definition of abuse cases.

- **Design**

During this stage, the security solutions that will cover the security requirements explained in the previous stage should be described. Furthermore, in this phase, functional details that have not been specified during the requirements stage should be described. Example: cryptographic algorithms to use.

The S-SDLC adds a set of principles to this stage that should be followed for the design of the system. Such principles should be taken into consideration transversally during the design of the system.

12.1.4.1 Design: Principles of Secure Design

- **Defence in depth**: It implies creating different security layers so that, in case one of them fails, the system will not be compromised. It requires the design of different defence strategies for the same threat.
- **Fail securely**: It means that all the failures will take the system to a status that is considered as secure (without losing confidentiality, integrity or availability).
- **Least privilege**: Each user or process should only have the least amount of privilege required to perform the necessary tasks. Privileges should also be granted for the shortest possible time.
- **Separation of duties**: The performance of any critical activity on the system should require the participation of two or more entities. It is aimed at eliminating single points of failure.

- **Keep security simple**: When two solutions provide the same security level, generally, the less complex one should be used. In general, the simpler, the smaller attack surface.
- **Supervision**: During the execution of any task (access, writing, modification), it is important to verify that the user or process that is executing it has the proper authorisation. In order to avoid synchronisation issues, it is recommended not to use authorisation caches.
- **Open design**: The details of the system design should be open in order to avoid security by obscurity. This principle is useful to create secure systems from the design. It ensures that the design publication or revision will not imply directly a serious security incident.
- **Least common mechanism**: This principle advises not to use the same security mechanism, even if it is common to various processes or users if they have different levels of privilege.
- **Acceptability**: The security mechanisms of the system should be designed taking into consideration the acceptability of users. If users face difficulties when using security characteristics, they will look for different mechanisms to avoid them, rendering them useless.
- **Weakest points**: The security of an entire system will depend on its weakest point.
- **Reuse**: It is preferable to use already existing and verified components than the creation of new ones that may increment the risk of vulnerabilities and the attack surface.

12.1.4.1.1 Surface of Attack

It implies specifying the **entry points** to the system in a structured way. This task should be performed by the designer. The entry points of the application can be divided into:

- Network.
- File System.
- User.

For each entry point, the following **elements** should be **identified:**

- Resources accessible through it.
- Roles that have access to such access points.

It allows users to identify resources leakages to roles that should not have the required privileges.

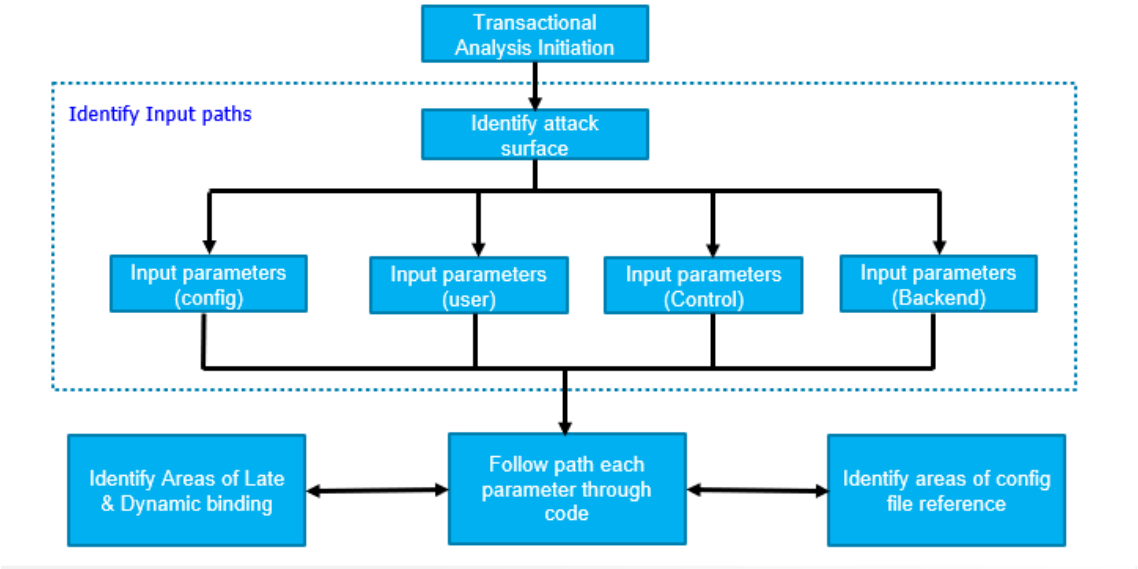


Figure 12-3: Understanding the attack surface [40]

12.1.4.1.2 Threat Modelling

Risks and threats that may affect a given system should be catalogued and assessed. It will be the first task carried out by an attacker. If it is not performed, the protection of systems is reduced. **(Threat = vulnerability)**. Threats are persistent.

The **threat modelling** is an iterative process that implies identifying **assets** or capabilities existing on the system by using a diagram. It is important to check whether they coincide with the ones identified in the documentation.

For each asset or capability, identify potential threats. This task requires the analyst to have some creative skills.

For each threat, assess the risk existing:

- Use threat trees that describe the different steps that should be followed by the attacker in order to materialize threats.
- Measure factors such as: impact, reproducibility, exploitability and affected users.

For each threat, identify controls that can be implemented to mitigate it. At the end of the process, as many threats as possible should be covered.

STRIDE is a threat model that groups them into six categories:

- **Spoofing**: a system or user is masqueraded.
  Example: a person or program tries to act as the system administrator.
- **Tampering**: modification of data or code.
  Example: modification of the source code of the application used to deactivate protections.
- **Repudiation**: denial of a specific action to have been carried out.
  Example: "I did not send that message".
- **Information Disclosure**: access to a piece of information by an entity that has no credentials to do it.
  Example: personal information leaked to the public.
- **Denial of Service**: blocking or degrading a service.
  Example: block of servers due to a high number of requests.
- **Elevation of Privilege**: increase of capabilities without the proper authorisation.
  Example: a user becoming administrator.

In order to **mitigate** the possible impact of a security breach, the developed controls below are established for such threats.

Table 12-3: The STRIDE Threats [41]

Threats	**Security control / service**	**Threat definition**
Spoofing	Authentication	Pretending to be something or someone other than yourself

Tampering	Integrity Controls	Modifying something on disk, on a network, or in memory
Repudiation	Non-repudiation methods	Claiming that you didn't do something, or were not responsible. Repudiation can be honest or false, and the key question for system designers is, what evidence do you have?
Information disclosure	Confidentiality mechanism	Providing information to someone not authorized to see it
Denial of service	Availability	Absorbing resources needed to provide service
Elevation of privilege	Authorisation	Allowing someone to do something they're not authorized to do

12.1.4.2 Implementation, Secure Development Environment

This stage implies the encryption of the different functions of the software product to develop, aimed at helping developers to implement the required functionalities as securely as possible. The main contributions of the S-SDLC to this stage are guides and **good practices on secure encryption**. Furthermore, during the implementation stage, an S-SDLC should consider the following activities:

- Secure configuration of the development environment.
- Revision of the application's source code.
- Revision of third parties' elements.

An **official configuration** should be defined for the **development environment** that will be used during the implementation of the software product. The configuration should specify:

- Valid operative system or systems (including versions).
- Tools supported in the development (including versions):
    - IDE
    - Versions control system
- Remote access restrictions.

The **required mechanisms** to apply and restrict the configuration of workstations to the accepted level should be included:

- User accounts restrictions.
- Pre-installed environments.

12.1.4.2.1 Code Review

The revision of the application's source code allows users to **identify vulnerabilities that are introduced during the implementation stage**. Automatic tools for static analysis make this task easier, but they are not able to find some vulnerabilities that require a manual revision. The review of code is an additional task to the execution of unit and integration tests that are defined within the traditional SDLC. It is not equivalent and should never substitute it. The **review of the source** code may be carried out:

- Lightly during the implementation process.
- Formally, once a part of the implementation process has ended.

Regarding **lightweight reviews of the source code**, it is possible to perform them in the following ways:

- Pair-programming techniques: two people develop code together in the same machine, supervising the written code mutually.
- External review: the author explains the code to other developer that verifies it.
- Assisted review: developers use semi-automatic tools that enable the identification of code problems during the programming.

- Commit review: when there is a commit on the version control system, a tool is launched in order to:
    - Send the element automatically via email to the reviewers
    - Conduct an analysis of the commit with an analysis tool integrated with the version control

      Example: https://www.pullreview.com/ or https://codeclimate.com/

12.1.4.2.2 Third Parties' Elements

During the implementation stage, it may be necessary to **use third parties' tools and libraries**. Controls made in our own code should also be implemented in this type of library. Specifically, the following activities will be carried out:

- If the source code is available, perform an analysis process as the one described before.
- Review vulnerabilities or possible security problems related to the library version used:
    - Secure storage
    - Encrypted communications
    - Validation of input data
    - Problems of misconfiguration or data exposure by default
- Check the use of functions or elements that have been declared to be deprecated by developers.

▪ **Verification**

On the traditional SDLC, this stage includes all the activities aimed at **verifying that the software product works as it is described in the requirements**. S-SDLC tasks during the verification stage allow users to perform security verifications directly on software elements that have been implemented in the previous stage:

- **Dynamic Analysis**: It enables the verification of the system's security properties and their behaviour by executing it.

- **Fuzzing**: it is a part of the dynamic analysis. It is checked whether controls implemented in entry points of the system control the possible entries properly.
- **Revision of the attack surface**: once the code is finished, it is possible to verify that the attack surface identified in previous stages of the S-SDLC corresponds to the real one.

- **Deployment**

In this stage, the software product is prepared for its **deployment**. Regarding the S-SDLC, this stage includes activities to cover security aspects of the product beyond its launch date.

Incident response plan:

It allows users to mitigate the scope of security incidents, reduce risks, and costs of an incident. It should clearly identify events to consider when declaring the existence of a security incident. For each incident, actions to carry out should be described in detail. The roles of each response team member and their contact information should be included as well. This task is essential in order to respond quickly against any security incident. The response plan is not a static document. It progresses according to modifications of the system or the appearance of new threats that were not taken into consideration.

Final security revision:

Before the launch, it should be verified that all the security tasks planned to perform the S-SDLC have been completed. Furthermore, it is advisable to carry out a revision of each task in order to ensure that no failures have been committed during the performance.

Certification:

It allows users to ensure that the product comply with certain security regulations/standards.

Storage:

It implies saving a copy of all the elements involved in the software version that is going to be launched. It will be one of the elements to be considered in case of security incidents. Tasks such as dynamic analysis, fuzzing and other security revisions are still executed during this stage.

- **Response**

This stage is only activated in **response** to events that have been declared as generators of an incident in the incident response plan. Once activated, the guidelines established by the plan should be followed, including:

- Staff to notify and order of notification.
- Data capture for the later analysis of the incident.
- Execution of tasks for the mitigation of the threat.
- Execution of tasks for the re-establishment of the service (if needed).

12.2 Good Practices in Secure Development Software

Guides and practices for the secure encryption of applications are independent from the programming language and the target platform. Its objective is to provide the developer a set of practices in order to implement software in a secure way without needing to know concepts related to vulnerabilities security and exploitation in depth. Following secure encryption good practices is not enough to ensure that an S- SDLC is being performed; it is only a part of the whole process [39].

Set of practices:

- Input validation.
- Output elements encryption.
- Password authentication and management.
- Session management.
- Access control.
- Error management.
- Data protection.
- Security in communications.
- System configuration.
- Security in databases.
- Memory management.

➢ Input Validation:

All inputs to the system are considered as malicious: all input is evil: (Text fields, URL, Cookies and other HTTP fields).

Input data validation should always be performed in a reliable system, generally, in the back-end. All data validation should focus on a specific part of the application. Before carrying out the validation of data, it is advisable to unify the encryption of data for all the verifications to be performed with the same encryption. It is recommended:

- To validate data ranges and length.
- To use white lists to verify that all the elements of an input are valid.
- Check that all the received data correspond to what is expected:
    - HTTP headers should include only ASCII characters
    - If an image is expected to be received in a specific format, verify that it is correct
    - URL text and parameters fields should include the type of data expected in the application
- In case that there are characters or strings that could be considered as dangerous < > ../ ..\ % \ ( ) “ ‘ \’ \” specific additional controls should be added for calls that may include them.
    - In the case of interpretable inputs (such as HTML code), avoid redirects that may render controls ineffective

➢ Output elements encryption:

The output data encryption should be performed in a reliable system such as the application’s back-end.

Multiple libraries and methods for output encryption that have been widely tested and accepted by the community are available:

- The following NSString class method can be used on iOS:
    - stringByAddingPercentEncodingWithAllowedCharacters
- On Android, URLEncoder or DatabaseUtils are available

Output data should be encrypted according to the way they are going to be used in the application:

- In case the output is going to be interpreted by a web browser, avoid the creation of interpretable elements in HTML CSS, Javascript, etc.
- If the output is going to be interpreted by another system, avoid the possible creation or modification of commands to them (SQL, XML, LDAP, etc.).

➢ Password Authentication and Management:

All the pages, except for those strictly defined as public, should require the authentication of users. The following recommendations should be followed for the implementation of authentication controls:

- Authentication controls should always be performed on a reliable system (back-end).
- All the authentication controls should be centralised on a unique module, including libraries able to perform calls to external authentication services.
- The authentication logic should not be connected to the accessed resource's logic.
- Authentication requests should always be performed via properly encrypted (SSL) HTTP POST connections.

The following practices are recommended for the authentication process:

- The validation of authentication data should only be performed if all the necessary information has been introduced (user and password).
- In case there is an authentication failure, no details (either visual nor in the source code used) regarding the specific authentication failure (incorrect password, incorrect user, etc.) should be provided.
- The authentication process should fail in a secure way.
- Regardless of the access method, the password field should not display the elements typed.
- Under no circumstances should passwords be stored in the non-encrypted application.
- All passwords should be stored summarised by using a secure cryptographic function, using a salt in order to complicate brute-force attacks via rainbow tables.

- The application should obligate users to use passwords with a minimum level of complexity:
    - Minimum length of 8 characters, but longer passwords are recommended
    - Alphanumeric characters, punctuation marks and numbers
- Force the user to change the passwords created by default during the first access
- If various failed access attempts occur, deactivate access for a period of time, long enough to avoid brute-force attacks, but not as long as to cause a denial of service.

Regarding the reset of passwords:

When possible, the use of security questions should be avoided. In case they are necessary, questions with a predictable or common answer should be avoided:

- Incorrect example: What is the name of your first pet?
- Correct example: Name of the street your mother lived in.

The email to which the reset request is sent should be verified to be registered in the system. In case any critical operation is going to be performed in the system, such as the change of password itself, the user should be asked to authenticate again. If possible, implement a double authentication factor via:

- Password + application that creates passwords that can only be used once (Google authenticator).
- Password + biometric element.

➢ Session Management

If the session control included in the framework server on which the application is developed provides sufficient guarantees, it is advisable to use it.

- iOS
- Android
- Java EE
- .Net
- Django
- Ruby on Rails

Identifiers should be created by a reliable system (generally, the back-end), with libraries that ensure that they are random enough. Every re-authentication should create a new session identifier and remove the old file. The user should be able to logout easily. Sessions should expire after a minimum period of inactivity. The exposition of information regarding sessions or cookies to third parties (logs record, use of GET parameters, etc.) should be avoided. According to budget constraints, a system that enables the control and logout of active sessions should be provided to the user.

➢ Asses Control

Access control decisions should be taken according to information coming from reliable systems. As in the case of input, output, and authentication validation, it is advisable that the access control system is centralised and separated from the rest of logic, in a unique element of the system. The access control should be performed for all the requests, including those carried out via technologies such as AJAX. Unauthorised users should not be allowed to access elements such as:

- Application and services data.
- URLs only accessible by authorised users, including images.

➢ Error Management

When an error occurs, it should be avoided to disclose sensitive information such as system details, sessions identifiers or accounts information. The application should manage all the errors and never depend on system errors by default.

When an error occurs, the policy used by default regarding the task performed should be the denial. Logs should register relevant events of the system:

- Failures on input validation.
- Failed authentication attempts.
- Connection attempts with expired sessions.
- Changes in the configuration of critical elements.
- Exceptions in the system and other errors occurred during the execution.

➢ Data protection

Passwords, authentication tokens and other sensitive information should be stored encrypted.

The storage of credentials within configuration files or the application's source code itself should be strictly avoided. If open repositories such as GitHub are used, access credentials to services could be published.

Configure the applications server for files of the back-end application's source code not to be downloaded.

Remove documentation and configuration files that are installed by default.

➢ Security in Communications

Since most connections include authentication tokens, sessions or sensitive information in order to access services of the application, connections between clients and the server should be encrypted.

Applications should verify the validity of the certificate provided and even use certificate pinning techniques to mitigate attacks to the PKI system.

Resources accessible via secure connections should not be available through insecure connections (downgrade to non-encrypted connections).

➢ System Configuration

Ensure that the versions of the third parties' elements required for the execution of the application are the ones approved during the design. It is also necessary to check that no vulnerabilities that may affect the approved versions have been registered. In such cases, they should be reviewed and chosen again.

The system in process of production should not include the source code and resources files that have been used to perform tests and verifications during the development process.

In case part of the application is provided by a web server, use the "robots.txt" corresponding file to avoid indexing. It is important to keep this aspect in mind since it may cause the providing of extra information to the attacker.

It is advisable to use a system for version control during the development of the software.

➢ Security in Databases

The access to the database should be made via parametrised queries, regardless the database type. Parameters used and results obtained in queries should go through their corresponding encryption and validation processes (escaping and filtering).

Applications and systems should use as least privileges as possible to access databases' tables. Roles with different access levels should access by using different users to ensure the separation of privileges.

The connection to the database should be maintained only as long as it is strictly necessary to complete the requests needed.

As in the case of the rest of subsystems (applications server, etc.), all the unnecessary files and installation configuration created by default should be removed.

➢ Memory Management

Some programming languages are responsible for the management of memory through their execution environments (Java, Javascript, Python, Swift, etc.) in an automatic way. However, others such as C, have a manual memory management system. In such cases, it is essential to carry out good memory management. Most critical vulnerabilities are caused due to memory management issues (buffer overflow).

The most critical aspects regarding memory management are related to:

- Buffer copies between memory addresses.
- Space reserved in memory for indefinite-length variables.
- Freed up memory that had previously been freed up.

It is advisable to follow a series of guidelines that limit the arising of memory management issues as much as possible:

- When using functions that accept the number of bytes to copy (such as strncpy), users should take into consideration the fact that the target buffer may not end in zero (not all the source bytes are copied).
- When creating copies between buffers, users should verify that sizes are correct and there is no possibility of writing when the space reserved for each buffer is exceeded (end-of-copy condition well defined).
- Maximum sizes should be defined for all the buffers used.
- When a variable that we reserved memory space for is not needed anymore, the resource should be released or closed. Users should not rely on the garbage collector function.
- If possible, users should avoid the use of dangerous functions such as strcat, strcpy, etc.

➢ General considerations

Regardless of the element to program, if there already is a tested and verified code that performs such an operation, using it would always be the best option.

When a task related to the operative system has to be performed, it should be executed through the API provided by it. Under no circumstances should commands be sent to the operative system through the console directly.

Whenever a code that has not been included in the initial deployment of the application (dynamic execution) is going to be executed, its integrity should be verified.

Synchronisation mechanisms existing should be used in the operative system in order to avoid the appearance of race conditions.

All the variables and sources of data should be initialized before first use.

The numerical representation of the programming language should be taken into consideration in order to avoid errors when carrying out calculations. Specifically, the following aspects should be taken into account: accuracy of operations, types of

signed/unsigned data, conversions, castings, and the way that the programming language handles numbers over and under the limits of representation.

If the application will implement automatic updating mechanisms, it should be verified that the code received during such updating comes from a reliable source.

Code signing mechanisms can be used to this aim, as the ones used in mobile applications shops/markets Once the code has been downloaded and before the update, its signature should be verified.

12.3 Software Protection Mechanism

At this point we will talk about some protection mechanisms in the development of secure software:

- Declarative and programmatic security (e.g., cryptographic agility, bootstrapping, and handling configuration parameters)
- Common software vulnerabilities and countermeasures
- Defensive coding practices (e.g., type safe practices, locality, memory management, error handling)
- Exception management
- Configuration management (e.g., source code and versioning)
- Build environment (e.g., build tools)
- Code/Peer review
- Code Analysis (static and dynamic)
- Anti-tampering techniques (e.g., code signing)
- Interface coding (e.g., proper authentication and third-party API)
- Develop a list of banned functions
- Advise developers about how to prevent the most important flaws.
- Develop with least privilege
- Never build your own cryptographic or authentication systems.
- Testing for Security Quality Assurance
    - Functional Testing (e.g., reliability, logic, performance and scalability)

  - Security Testing (e.g., white box and black box)
  - Environment (e.g., interoperability)
  - Bug tracking (e.g., defects, errors and vulnerabilities)
  - Attack surface validation
- Test types
  - Penetration Testing
  - Fuzzing, Scanning, Simulation Testing (e.g., environment and data)
  - Testing for Failure
  - Cryptographic validation (e.g., environment and data)
- Impact Assessment and Corrective Action
- Standards for software quality assurance (e.g., ISO 9126, SSE-CMM and OSSTMM)
- Regression testing
- Use security testing tools to discover common vulnerabilities
- Implement static analysis testing for all Internet facing code
- Add security bug categories to the bug tracking system

We suggest a series of seven guidelines for tackling the complexities of acquiring, designing, deploying, and maintaining software assurance systems in order to obtain the required degree of trust [42].

- Risk drives assurance decisions. Without adequate software assurance, organizations are only aware of threats when successful attacks on software and processes occur, and their reaction is reactive rather than proactive. Assurance options are implemented based on an organization's understanding of the possibility of an attack.
- Both stakeholders and interconnected infrastructure components must be coordinated on risk issues. Risk must be aligned across all parties and all embedded infrastructure components in highly connected systems; otherwise, important threats would be overlooked or neglected at various points in the interactions.

  Interactions take place at a variety of technological stages (e.g., network, security appliances, architecture, applications, and data storage). Protections may be added at any of these stages, but if they are not well coordinated, they can overlap. Because of encounters, successful assurance necessitates that all layers and positions identify and react to danger in a clear manner.

- Dependencies shall not be trusted until they have been proved to be reliable. The integrated software inherits all of the assurance limitations of each interacting component. Organizations must determine how much confidence they put in dependencies based on a practical evaluation of the risks, impacts, and opportunities that different interactions represent. Organizations must assess confidence relationships on a regular basis to recognize changes that need reconsideration.
- It is to be anticipated that there will be attacks. Attackers with increasing technological capabilities can compromise an organization's technology assets' confidentiality, credibility, and availability. Attackers create a solution using technologies, procedures, norms, and practices (known as a socio-technical response). Some attacks take advantage of how we usually use technology, while others establish unusual circumstances in order to get around defences.
- Effective coordination among all technology participants is needed for assurance. Since attackers take advantage of all potential entry points, organizations must apply security through their personnel, processes, and technology. To ensure that members of the company efficiently engage in software assurance, organizations must clearly define authority and responsibility for assurance at an acceptable level. People must be educated on software assurance in organizations.
- Assurance must be well planned and dynamic. Assurance must strike a balance between software and device governance, design, and service, and it is extremely sensitive to changes in both of these fields. Since change is so frequent, this is not a one-and-done activity; it must proceed through organizational sustainment after the initial operational implementation. This cannot be applied later; systems and applications must be designed to provide the degree of assurance that organizations need. No one has the ability to constantly update systems as threats evolve.
- There should be a way to assess and audit overall assurance. To assess operational assurance, all elements of the socio-technical system, including policies, processes, and procedures, must be linked. Organizations with more effective assurance mechanisms are able to respond and recover more quickly. Organizations must take targeted and systemic steps to ensure that the modules are designed with sound protection in mind, and that the relationship between them creates successful assurance.

12.4 Assess Software Acquisition Security

Vulnerabilities in software, malicious code, and software that does not perform as expected pose a significant threat to an organization's software-intensive critical infrastructure, which provides vital information and services. Software assurance's job is to reduce these dangers (SwA). “Software assurance is the degree of trust that software is free of bugs, either deliberately built into the software or inadvertently introduced at any point during its lifecycle,” according to the United States Committee on National Security Systems National Information Assurance (IA) Glossary [38].

Software assurance refers to the degree of trust that a piece of software is free of flaws and performs as expected. The aim of software assurance isn't to achieve perfection. It is attempting to ensure that all of the software's various characteristics are known to users and customers, and that nothing can come as a surprise.

SwA is important because software that does not work as expected and is exploitable causes drastic increases in company. Intellectual property, customer trust, company processes and facilities, and a wide range of essential infrastructure, from process control systems to commercial software products, are all at risk from software vulnerabilities. To ensure the safety of critical infrastructure operations and key properties, confirm that the software is reliable and stable. SwA can be organized around the major phases of a generic acquisition process:

- **Planning Phase**

- Needs to be determined whether to purchase software services or products, evaluate possible alternative software approaches, and assess risks associated with those alternatives;
- Developing software requirements to be included in work statements;
- Developing an acquisition strategy and/or plan that involves defining the risks associated with different software acquisition strategies;
- Developing evaluation criteria and an evaluation plan.

In our planning phase, what we need to do in-house is we need to sit down and decide, literally, what do we need? What do we need it to do? What should it look like? How should it perform? And so, we go through our requirements collection, sifting, sorting, and finalizing

process. We have to create an acquisition strategy. How are we going to buy this? What do we want it built as? What do we want our terms and conditions of support to be? And then we have to develop the criteria by which we're going to measure the software's quality. And develop the plan through which that criteria will be employed so that we can evaluate a competitor's product against another product so that we know we're getting exactly what we have paid for.

- **Contracting Phase**

So, in the contracting phase, and many of you may have been part of this, we're going to create or issue a solicitation or a request for a proposal. Based on how many we get, we'll evaluate the supplier proposals, see which one aligns with what our needs are as we know them and then, after having vetted the various offers of these products, we'll make a selection and then we will finalize our contract negotiation process and sign someone up to do the task.

- **Monitoring and Acceptance Phase**

- Establishing and consenting to the contract work schedule;
- Implementing change (or configuration) control procedures;
- Reviewing and accepting software deliverables.

In the monitoring and acceptance phase, the organization that has been selected to do it will establish their process, they will design and build, and at various points, they will confer with us and we will review and we will agree to proceed. The contract work will follow a schedule of deliveries. There will be a change control process. And, in each case, there will be a process to review and accept software deliverables, either as a final, finished product or in a modular form. More often today, it's in more of a modular form.

- **Follow-on**

- Sustainment (includes risk management, assurance case management, and change management);
- Disposal or decommissioning.

A secondary matter, of equal importance however, is the sustaining engineering. How will the software be maintained? What is its expected life span? How will this process be done? And,

of course, many of the same subjects. What will be the change control process? Will it be supported in-house or will it be supported by the builder? Or will it be done by another party? And then at the end, there will come the question of how will it be disposed or decommissioned at the end of its life? And that, of course, begs the question, what will replace it? And how will our decommissioning process go if questions of compliance and sanitizing have to be answered?

Through the process of software acquisition, we have to be sure that our quality assurance process, both proactively for what we will build into the contract arrangements, and for the analysis process that we will do as each module or as the final product is being delivered. There must be a process for assessing and coping with the risks that occur during the software build and delivery process.

So, to begin this process, we first must develop a software acquisition policy. It needs to be well-documented and it needs to be sure to cover the various topics that we've addressed: security, development, compliance, and a host of others. And this must be the policy that guides how we will do this to make sure that we get what we paid for and that it works as well as expected.

Risk Associated with Software Vulnerabilities

Our contracting language needs to deal with the unintentional errors, potential insertion of malicious code, theft of vital information or even intellectual property, theft of personal information, controlled or unauthorized change of the product, any form of inserted agents or any form of corrupted information.

Now the contract language and the processes that we build and implement and perform driven by that language need to be sure that our review and change processes account for testing and evaluation of each of these and a host of other related items to ensure that these unwanted attributes are not embedded in the program so that we can trust that it will perform as securely with all the integrity that we've bought and paid for.

The acquisition process itself needs to look at the system and software assurance so that we focus on the management of risk, the assurance of safety and security, dependability and a

host of other related attributes including compliance within the context of the system and the software life cycles.

These are some examples of **useful questions** that should be asked of potential suppliers through the vetting process and through the design, build and delivery:

- How does the supplier ensure that a safety and security system is built and maintained?
  - Ensures that the staff is competent in terms of safety and security?
  - Have you created a professional work environment (including the use of professional tools)?
  - Ensures the protection and safety of information?
  - Monitors activities and reports events (in the context of the software's deployment environment)?
  - Is there a way to ensure business continuity?
- What steps would the supplier take to identify and handle safety and security risks?
  - Identifies threats to safety and security?
  - Analyses and ranks threats in terms of safety and security?
  - Who is responsible for determining, implementing, and monitoring the risk reduction plan?
- How does the supplier ensure that the supplier's safety and security standards are met?
  - Who decides on administrative conditions, rules, and guidelines?
  - Develops and deploys goods and services that are safe and secure?
  - Is it possible to test goods objectively?
  - Establishes reasons for protection and security assurance?
- How does the supplier ensure that operations and goods are handled to meet the standards and goals for safety and security?
  - Establishes independent reporting on safety and security?
  - Is there a protection and security plan in place?
  - Uses safety and security standards to select and handle vendors, goods, and services?

- Does it keep track of and monitor activities and goods in terms of safety and security?

12.5 Referencing

[4] C. I. S. S. Professional, OFFICIAL (ISC) GUIDE TO THE CISSP CBK, Fourth edition.

[34] A. B. Vijay Madisetti, Internet of Things A Hands-On- Approach, VPT, 2014.

[38] O. G. t. t. C. CBK, Certified Information Systems Security Professional, Adam Gordom, Fourth Edition.

[39] Incibe, Secure Mobile Application, Unit 5 [Online] Available: https://www.incibe.es/formacion/curso-avanzado-ciberseguridad-dispositivos-moviles.

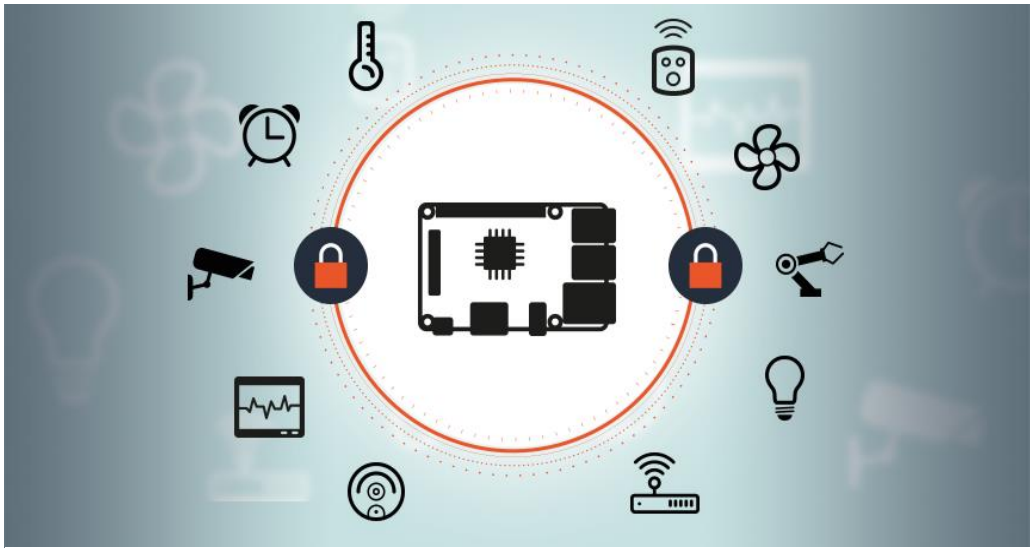
[40] OWASP Foundation, "OWASP CODE REVIEW GUIDE V1.1," 2008.

[41] A. Shostack, Threat modeling designing for security, WILEY, 2014.

[42] N. Mead, «Seven Principles for Software Assurance» Software Engineering Institute, 24 october 2016. [Online]. Available: https://insights.sei.cmu.edu/sei_blog/2016/10/seven-principles-for-software-assurance.html. [Last access: 17 December 2020].

13 Impact of new technologies on cybersecurity

Author(s): Beatriz Lorenzo Veiga
René Lastra Cid



13.1 Advanced Persistent Threats (APTs)

13.1.1 Introduction: Threats and Motives

A threat is described as "a person or thing capable of exploiting a vulnerability." It is a mixture of an attacker's motivations and skills, as well as knowledge of what the attacker has done in the past. While we can't predict a threat's behaviour based solely on their previous actions, it can give us some insight into what they might do in the future [43].

Threat Capabilities. These threats are listed in order of least to greatest capability.

- Unsophisticated Threat (UT)
- Unsophisticated Persistent Threat (UPT)
- Smart Threat (ST)
- Smart Persistent Threat (SPT)
- Advanced Threats (AT)
- Advanced Persistent Threat (APT)

As shown in Figure 13-1, the APT has the most advanced ability set of risks. We may presume that there are much more UTs than APTs because it takes far longer to acquire the skills required to be considered an APT than it does to be considered a UT. Furthermore, empirical evidence suggests that there are far less advanced threats than other threats.

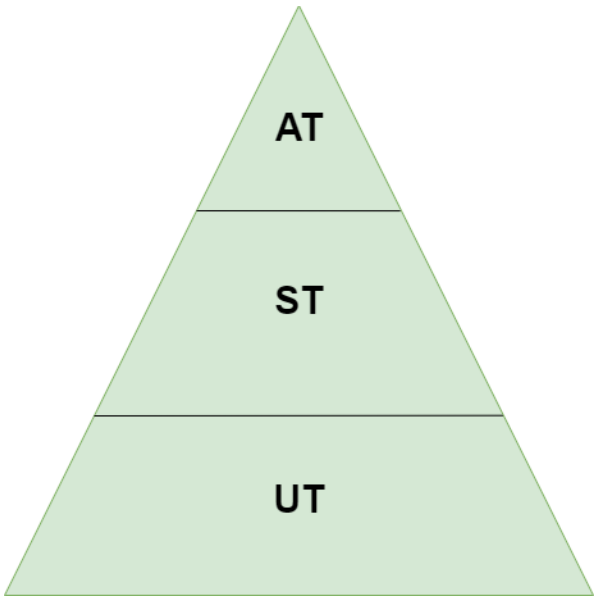


Figure 13-1: Threat capabilities pyramid [43]

13.1.2 Advanced Persistent Threats

An APT is a sophisticated threat with the aim of compromising a particular objective. Simple economics previously constrained an attacker aiming to compromise a particular goal. That is, if it costs an attacker more to compromise a target than the assets gained are worth, the attacker will either avoid attempting to compromise the target or will use all of their limited resources attempting to compromise the target. While economic laws still apply, the cost of compromising any goal has decreased to the point that a single person may penetrate any organization with very limited resources, including time and money.

The people or organizations that represent APTs include nation-states, organized crime, mercenaries and criminals [43].

An APT's objectives are almost infinite. Anyone may become a victim. APTs can target a particular organization for a variety of reasons, including:

- Intellectual property theft (corporate espionage)
- Stealing private data (insider trading, blackmail, espionage)
- Stealing money (electronically transferring funds, stealing ATM credentials, etc.)
- Leaking classified information from the government (spying, espionage, etc.)
- Reasons that are political or activist

Knowing the maximum amount anyone is willing to pay you, or the minimum amount someone else is willing to accept, finding out a public company's financial details before the rest of the world does, knowing what information the prosecution's attorneys have on you, or actually knowing the hidden formula your rivals use are all examples of motivation.

13.1.3 Threat Class and History

A threat class is a combination of an attacker’s motives and their capabilities, as shown in Table 13-1. Based on empirical evidence, there are commonly agreed threat classes into which each of these threats fits; however, all of these threats may map to any of the identified threat classes.

Table 13-1: Threat classes

Motives + Capabilities	=	Threat Class
Hacker + UT	=	Unsophisticated Hacker
Nation-State + APT	=	Advanced Persistent Nation-State
Nation-State + UT	=	Unsophisticated Nation-State
Techno-criminals + ST	=	Smart Techno-criminals

The history of specific threats that have harmed your company will reveal useful information about the capabilities and approaches used to compromise goals. However, you can't predict potential attacks based solely on a threat's past. An organization will be able to eliminate certain attack paths and vulnerabilities, but it will never be able to eliminate all attack vectors available to an APT hacker.

13.1.4 APT Hacker

An APT hacker is a single person with advanced skills and methodologies who can target and compromise any organization while obtaining access to any desired assets.

APT Hacker Methodology (AHM) is a method for systematically targeting and attacking a organization. It entails the following procedures:

- Reconnaissance: One of the most important moves for an APT hacker is reconnaissance. One of the main distinctions between a smart threat and an advanced threat is the ability to conduct proper (and extended) reconnaissance. This step should not be hurried or underestimated because it is critical to fully comprehend your goal, its industry, its people, and the technologies in place.
- Enumeration is the last step in the reconnaissance process, and it focuses on identifying precise information regarding a specific piece or system within an organization. Identifying particular software versions, user name structures, or responsible parties for specific systems are only a few examples.
- Exploitation is the process in which you take advantage of the vulnerabilities you discovered during the reconnaissance and enumeration processes. This is usually

enough to get you a foothold in a target company. The ability to succeed during the exploitation process depends on having properly trained.

- Maintaining access: An APT hacker's access must be maintained at all times. If the weakness you initially exploited is mitigated or otherwise unavailable, this move includes leaving a method for you to easily regain access to the compromised device. This is critical, and depending on the target device and network, it can be done in a variety of ways.
- Clean up: During an attack, cleaning up can take several different forms. Cleaning up evidence of effective exploitation, removing evidence of the process used to retain access to a device, or fully removing all signs of enumeration and reconnaissance are examples.
- Progression: There are several different types of progression. It may be gaining additional rights to a system that was compromised during the exploitation process, or gaining access to more networks on the target network in certain cases. Sections of this process are referred to as lily-padding, leapfrogging, or pivoting, in which we use the compromised system to attack other internal network structures.
- Exfiltration: As an APT hacker, you must think of the most efficient way to get the data you need from your target, whether it's just a username and password from another device or a multi-terabyte file.

In most cases, these steps are completed in this order. They can be iterative, and they can be performed in any order, as well as multiple times within a single attack. You may, for example, conduct reconnaissance and enumeration on a target organization before exploiting a weakness and gaining access to an internal system. You may need to conduct surveillance and enumerate the internal network before moving on to manipulating another device after developing a mechanism to retain access to the compromised system and cleaning up the evidence of your attack.

13.1.4.1 APT Hacker Attack Phases

When attacking a specific organization, there are five stages that must be completed. We begin by launching attacks that will ensure our anonymity. Then we'll move on to attack

stages, where we'll gradually give up a percentage of our anonymity in return for attacks with a high chance of succeeding. Finally, if none of our digital attacks succeed, we will penetrate physically. The attack is divided into five phases:

1. Reconnaissance entails gathering and analyzing all available information about the goal. Non-technical data and technical data are the two broad types of reconnaissance data.
    a. Technical data: Internet-routable subnets used by the organization, antivirus software, and DNS records for the organization.
    b. Nontechnical data: the organization's geographical positions, major departments, and key personnel.
2. Social engineering: Automated techniques are used to exploit specific individuals who may be targeted for manipulation and who are likely to have some level of access to the target asset to expose confidential information, passwords, or obtain remote access to the user's device. Email, instant messaging systems, and USB drives are only a few examples of digital methods.
3. Remote and wireless: Wireless networks and vulnerabilities offer as much anonymity as possible while keeping the target organization's systems in close proximity. Specially built and scalable spoofed wireless access points are often used to hack end-user wireless clients.
4. Hardware spoofing: Trojan hardware devices are used to attack end-users and key physical locations, compromising a linked computer system or remotely accessible eavesdropping systems.
5. Finally, we'll infiltrate unique physical locations, such as the target organization's offices, target users' houses, remote third-party facilities, and even remote staff in hotel rooms. Our physical penetration will be combined with attacks aimed at compromising key technological structures, bugging key physical locations, or gaining access to intermediate or target physical properties.

13.1.4.2 APT Hacker Foundational Tools

In all phases of the attack, some tools and techniques will be needed. The main goal of these tools is to keep your identity as anonymous as possible. We will still leave small traces of our existence, but they will be minor and will sometimes lead investigators on a wild goose chase to a location unrelated to us.

13.1.4.2.1 Anonymous Purchasing

There are alternatives to cash for keeping our transactions private: credit card gift cards and digital currencies.

Some credit card companies provide prepaid gift cards that can be used as a credit card anywhere and don't need any personal information to activate. We can also make use of digital money (cryptocurrency, Bitcoin or Litecoin). Many retailers now embrace these digital currencies, which are designed to keep all of your transactions anonymous.

13.1.4.2.2 Anonymous Internet Activity

When using the Internet for any purpose, we must be vigilant to keep our actions anonymous and untraceable. We'll do this by tunneling all of our communications into an intermediary device, which would then appear to be the network's contact source. As a result, if the correspondence was traced back, it would be assumed that the intermediate host was the true source. There are three main technologies that we can use to keep our online activities private:

- Open, free, or vulnerable wireless networks
- Virtual private server pivots
- Web and socks proxy

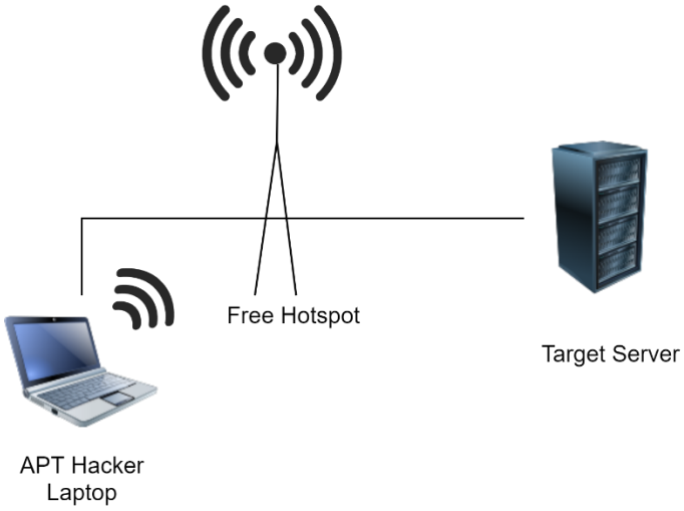


Figure 13-2: Pivoting through open wireless network [43]

13.1.4.2.3 Anonymous Phone Calls

As an example, as shown in Figure 13-2, we can probe and attack our target organization using an open wireless network. The public IP address of the Free Wifi Hotspot will be visible in the logs on the target server.

13.2 Advanced BYOD and Technology Customization

Many people telework, and they use a variety of devices to read and send messages, access websites, review and edit documents, and perform a variety of other activities, like desktop and laptop computers, smartphones, and tablets. The company, a third party (such as the organization's contractors, business associates, and vendors), or the teleworker control each telework system; the latter is known as bring your own device (BYOD). This section contains guidelines for protecting BYOD devices that are used for telework and remote access, as well as those that are directly connected to the company's network [44].

Many organisations implement restrictions on the types of BYOD devices that can be used and the services that they can access, such as allowing only BYOD laptops access to a restricted set of resources and allowing all other BYOD devices to access only webmail. This allows businesses to reduce the risk generated by BYOD products.

13.2.1 Securing BYOD PC used for telework

Teleworkers who use a bring-your-own-device (BYOD) desktop or laptop (PC) for telework should keep the operating system and primary applications secure.

The following steps are involved in securing a BYOD PC:

- To stop most attacks, especially malware, use a combination of security software such as antivirus software, personal firewalls, spam and web content filtering, and popup blocking;
- Having a separate standard user account for each person, assigning a password to each user account, using the standard user accounts for everyday usage, and protecting user sessions from unauthorized physical access are all ways to limit who can use the PC;
- Ensuring that the operating system and key programs, such as web browsers, email clients, instant messaging clients, and security software, are updated on a regular basis;
- Disabling unnecessary networking features on the PC and setting up reliable wireless networking;
- Configuring primary applications to filter content and avoid potentially malicious activity;
- Only installing and using software that is well-known and trusted;
- Configuring remote access software in accordance with the needs and guidelines of the organization;
- Keeping the security of the PC up to date, such as changing passwords on a regular basis and checking the status of security software on a regular basis.

13.2.2 Securing BYOD devices used for telework

Teleworkers who use a BYOD mobile device for telework can protect it according to the device's manufacturer's security recommendations.

- Disable networking features like Bluetooth and Near Field Communication (NFC) unless they're absolutely necessary;

- Ensure that, if available, security updates are downloaded and activated at least regularly, preferably daily.
- Configure applications to support security (for example, blocking potentially malicious activity);
- Apps can only be downloaded and run from approved app stores.
- Do not jailbreak or root the device;
- Do not link the computer to an unknown charging station; and to access the organization's data and services, use an isolated, secure, and encrypted environment that is sponsored and maintained by the organization.

Teleworkers should avoid using any client system that is not regulated by the company.

13.2.3 Securing Information

Sensitive data, such as personally identifiable information (PII), must be secured on or submitted to or from telework devices so that unauthorized parties cannot access or modify it. An unauthorized release of sensitive information may harm the public's confidence in an agency, jeopardize its purpose, or harm individuals whose personal information was published.

Until teleworking, users should be aware of their company's policies and specifications, as well as the best practices for protecting the company's data. The following are some examples of methods that organizations can expect or need teleworkers to use:

- For telework devices and removable media, use physical security controls. When traveling to hotels, conferences, or other locations where third parties may easily access the machines, companies can request that laptops not be left unattended. Papers and other non-computer media containing classified details that are carried off the organization's premises may also be subject to physical security standards.
- Through encrypting files on telework devices and portable media such as CDs and flash drives, attackers are prevented from easily accessing the information contained in the files. Encryption of individual files or directories, and hard drives are only a few of the choices for protecting files. When using encryption to secure files, you must also use

an authentication mechanism (such as a password) to decrypt them when appropriate.

- It's important to make sure the data stored on teleworking devices is backed up. Teleworkers in some organizations will back up their local files to a centralized system (e.g., via VPN remote access). Other companies advise their teleworkers to back up locally (e.g., by burning CDs, copying files to removable media). Teleworkers can back up their data and double-check that the backups are accurate and complete. Backups to removable media must be secured at least as well as the computer that is being backed up.
- When information is no longer required, make sure it is destroyed. A computer that is about to be removed should have all of its organizational files removed. Some remote access methods perform a simple wipe of information, such as clearing web browser caches that may contain confidential information, but a more comprehensive wipe also requires the use of a special tool, such as a disk cleaning software designed to delete all traces of information from a computer. Many companies assist their teleworkers in deleting details from BYOD computers.
- Delete information from devices that have been lost. The content of a missing or stolen smartphone or tablet may be remotely wiped by the organization or its service provider. By erasing the content, an intruder is unable to extract any information from the computer. This service's availability is determined by the product's functionality and the company that provides network services for the product.

Teleworkers must also know how to deal with social engineering challenges. Attackers who use social engineering to trick people into disclosing confidential information or performing specific acts, such as uploading and executing files that seem innocent but are actually malicious, are known as social engineers. Teleworkers should be vigilant of any request that might result in a security breach or telework system theft. If their organizations do not already have training on how to recognize and respond to social engineering attempts, they may request it.

13.2.3.1 Securing Home Networks and Using Other Networks

The application of security measures to the domestic networks to which the teleworker's device is normally connected is a vital part of the teleworker's and remote access security. Another critical aspect of domestic network security is the protection of other computers and mobile devices, because if one of these devices are infected with malware or are compromised in some way, it can be used to attack the teleworker's device or spy on their communications. Teleworkers should be aware of the risks of using public networks, as well as the protocols for connecting theirs telework devices, like BYOD devices, to the company's internal networks.

13.2.3.2 Wired Home Networks

Teleworkers can safeguard their wired home networks such that their teleworking devices are protected. The most critical aspect of the securing of the most wired home networks is the separation of the home network from the ISP network. If a telework interface is directly connected to the ISP of the teleworker, for example, when the device is directly connected to a cable modem, it is accessed directly from the Internet and risks being targeted quite highly. A monitoring device between the ISP and teleworking device should be in the home network in order to avoid this. This is more often done via a broadband router (e.g. cable modem router) or firewall device.

This protection system should be set up to prevent computers on the outside of the home network from communicating with all of the home network's devices, including the telework device. Even if each system has its own personal firewall, an additional layer of security should be provided by a firewall appliance, broadband router, or other similar protection. If a personal firewall on a computer fails, the appliance or router can still protect the computer from uninvited network communications from other computers.

13.2.3.3 Wireless Home Networks

Information is transferred between a telecommuting system and a wireless access point through wireless networks. A wireless home network that is improperly configured can

transmit confidential data without sufficient security, exposing it to other nearby wireless devices. Teleworkers should secure their wireless home networks to protect their remote access communications, and they should follow the security guidelines in the wireless access point documentation for their home network. Assuming that the network uses IEEE 802.11 protocols, the following are some examples of standard security recommendations:

- To secure communications, use strong encryption. Wi-Fi Protected Access (WPA) is a set of product protection certifications established by the Wi-Fi Alliance, which includes the WPA, WPA2, and WPA3 certifications. Wireless networking devices must meet a set of security standards defined by these certifications. Devices with wireless network cards that support WPA, WPA2, or WPA3 can use security features like the Advanced Encryption Security (AES) algorithm to encrypt network communications. The following are the suggested options:
    - WPA3 with AES
    - WPA2 with AES
    - WPA with AES
- Use a key that is WPA3, WPA2, or WPA. This key is a sequence of characters (either a password made up of letters, digits, and punctuation, or a hexadecimal number) used to restrict access to a wireless network. A wireless AP can be set up to request that each device have the same key that the AP has stored. The wireless network is inaccessible to devices that do not know the secret. The main should be long and complicated, making guessing difficult. This should help prevent people in close proximity to the AP from obtaining unauthorized network access.
- Only allow access to specific wireless network cards. Some APs may be set up to only allow those devices to connect to the wireless network. This is achieved by defining each device's wireless network card's media access control (MAC) address and entering it into a list on the AP. Since a MAC address should be exclusive to a network interface, defining the MAC address in the AP will help prevent unauthorized users from gaining access to the wireless network.
- Change the Service Set Identifier that is used by default (SSID). An SSID is a name given to a wireless access point that enables people and devices to differentiate between

different wireless networks. The default SSID on most APs is the manufacturer's or product's name. If the default SSID isn't modified, and another nearby wireless network has the same default SSID, the teleworker's computer can try to enter the wrong wireless network by mistake. Changing the SSID to anything unusual reduces the likelihood of a system selecting the incorrect network.

- Disable the wireless AP's SSID broadcasts. Many wireless APs broadcast the SSID, which effectively advertises the AP's presence to any nearby computers. Configuring an AP so that it does not transmit its SSID reduces the likelihood of people joining the wireless network by accident, but it does not prevent an intruder from doing so.
- Over wireless communications, disable AP management. Wireless AP control utilities are often found to have bugs. If an AP has a bug like this, attackers in the area could use it to disable security features or gain access to the teleworker's home network or the Internet. Teleworkers can configure access points such that they can only be handled locally to prevent such accidents.

13.2.4 External Networks

It's unlikely that non-domestic networks can provide any security to teleworker devices and communications, such as a laptop that connects to an insecure point of access in a cafe. It's possible that external networks don't count the network's communications, making them vulnerable to attack. Teleworkers should assume that third-party networks offer no security. Teleworkers should make sure their computers are up to date before using a third-party network. Updates should be downloaded from a secure network, such as the user's home network. When connecting to a third-party network, they should use a VPN or other safe remote access solution offered by the organization, and they should enable the secure remote access solution (for example, by creating a VPN session) immediately after connecting to the third-party network.

13.3 The cloud and the economics of collaboration: risks and benefits

13.3.1 Cloud Computing Characteristics

Cloud computing has the following five essential characteristics. In order for an implementation to be considered a cloud, each of these five characteristics must be present and operational [45].

- On-demand self-service: Cloud services can be requested, provisioned, and put into use by the customer through automated means without the need to interact with a person. This is typically offered by the cloud provider through a web portal but can also be provided in some cases through web API calls or other programmatic means. As services are expanded or contracted, billing is adjusted through automatic means. Self-service is an integral component of the "pay-as-you-go" nature of cloud computing and the convergence of computing resources as a utility service.
- Broad network access: All cloud services and components are accessible over the network and, in most cases, through many different vectors. This ability for heterogeneous access through a variety of clients is a hallmark of cloud computing, where services are provided while staying agnostic to the access methods of the consumers. In the case of cloud computing, services may usually be reached from either a server or thick or small computer or from a corporate web from a personal user on an open network. It does not include the use of a handheld device, laptop or desktop.

The cloud revolution in computing has occurred concurrently with the mobile computing revolution, making the importance being agnostic concerning the means of access a top priority. Because many companies have begun allowing bring-your-own-device (BYOD) access to their corporate IT systems, it is imperative that any environments they operate within be able to support a wide variety of platforms and software clients. Cloud storage also alleviates the need for users to store their data on their devices; instead, they can access it via the network, thus increasing security by removing data storage physically from the device.

- Resource pooling: One of the most important concepts in cloud computer is resource pooling, or multitenancy. Regardless of the type of cloud offering, there will always be a combination of applications and systems coexisting within the same collection of

physical and virtual resources in a cloud environment. The new resources are dynamically distributed within the cloud as cloud users add to and extend their use, and the customer has no control about where the actual services are implemented.

This aspect of cloud can apply to any type of service deployed within the environment, including processing, memory, network utilization and devices, as well as storage. At the time of provisioning, services can and will be automatically deployed throughout the cloud infrastructure, and mechanisms are in place for locality and other requirements based on the particular needs of the customer and any regulatory or legal requirements that they be physically housed in a particular country or data center. However, these will have been configured within the provisioning system via contract requirements before they are actually requested by the customer, and then they are provisioned within those rules by the system without the customer needing to specify them at that time.

Many corporations have computing needs that are cyclical in nature. With resource pooling and a large sample of different systems that are utilized within the same cloud infrastructure, companies can have the resources they need on their own cycles without having to build out systems to handle the maximum projected load, which means these resources will not be unused and idle at other non-peak times. Significant cost savings can be realized for all customers of the cloud through resource pooling and the economies of scale that it affords.

- Rapid elasticity: Since cloud computing is decoupled from hardware and allows for programmatic provisioning, services can be quickly extended at any time when more resources are required. This functionality may be offered via the web portal or implemented on behalf of the customer, either in response to or during an anticipated or predicted increase in service demand; the decision to adjust scale is measured against the customer's funding and capabilities. If the applications and systems are built in a way where they can be supported, elasticity can be automatically implemented such that the cloud provider through programmatic means and based on predetermined metrics can automatically scale the system by adding additional resources and can bill the customer accordingly.

In a classic data center model, a customer needs to have ready and configured enough computing resources at all times to handle any potential and projected load on their systems. Along with what was previously mentioned under "Resource Pooling," many companies that have cyclical and defined periods of heavy load can run leaner systems during off-peak times and then scale up, either manually or automatically as the need arises. A prime example of this would be applications that handle healthcare enrollment or university class registrations. In both cases, the systems have very heavy peak use periods and largely sit idle the remainder of the year.

- Measured service: Depending on the type of service and cloud implementation, resources are metered and logged for billing and utilization reporting. This metering can be done in a variety of ways and using different aspects of the system, or even multiple methods. This can include storage, network, memory, processing, the number of nodes or virtual machines, and the number of users. Within the terms of the contract and agreements, these metrics can be used for a variety of uses, such as monitoring and reporting, placing limitations on resource utilization, and setting thresholds for automatic elasticity. These metrics also will be used to some degree in determining the provider's adherence to the requirements set forth in the service level agreement (SLA).

Many large companies as a typical practice use internal billing of individual systems based on the usage of their data centers and resources. This is especially true with companies that contract IT services to other companies or government agencies. In a classic data center model with physical hardware, this is much more difficult to achieve in a meaningful way. With the metering and reporting metrics that cloud providers are able to offer, this becomes much more simplistic for companies and offers a significantly greater degree of flexibility, with granularity of systems and expansion.

13.3.1.1 Cloud Deployment Models

The possible options for cloud deployment are shown below:

• Available to the general public • Located on the premises of the cloud provider • May be owned by a private company, organization, academic institution, or a combination of owners • Easy setup and inexpensive to the customer • Pay only for services consumed	Public	Private	• Owned and controlled by a single entity • Primarily used by that entity for their own purposes, but may be opened to collaborating organizations • May be operated by the organizations • May be operated by the organization or a third party • Can be located on or off premises • Can be used by different departments with internal billing
• Composed of two or more different cloud models (public, private, community) • Standardized or proprietary technologies that enable portability between models • Typically leveraged for load balancing, high availability, or disaster recovery	Hybrid	Community	• Owned by a group of similar organizations for use within the group • Models and features similar to a private cloud • Managed and controlled by the member organizatins • May exist on or off premises of the ownership organization

Figure 13-3: Cloud deployment models [45]

A public cloud is a model that makes cloud services available to the general public, as well as any company or organization, with no financial or planning constraints. It may be owned, controlled, and run by a corporation, an academic institution, or a government agency, or a combination of these entities. It is located on the cloud provider's premises.

Below are key benefits and features of the public cloud model:

Setup: This is very easy and inexpensive for the customer. All aspects of infrastructure, including hardware, network, licensing, bandwidth, and operational costs, are controlled and assumed by the provider.

Scalability: Even though scalability is a common feature of all cloud implementations, most public clouds are offered from very large corporations that have very broad and extensive resources and infrastructures. This allows even large implementations the freedom to scale as needed and as budgets allow, without worry of hitting capacity or interfering with other hosted implementations on the same cloud.

Right-sizing resources: Customers only pay for what they use and need at any given point in time. Their sole investment is scoped to their exact needs and can be completely fluid and agile over time based on either expected demand or unplanned demand at any given point in time.

A *private cloud* differs from a public cloud in that it is run by and restricted to the organization that it serves. A private cloud model may also be opened up to other entities, expanding outward for developers, employees, contractors, and subcontractors, as well as potential collaborators and other firms that may offer complementary services or subcomponents.

The following are key benefits and features of the private cloud model:

Ownership retention: Because the organization that utilizes the cloud also owns and operates it and controls who has access to it, that organization retains full control over it. This includes control of the underlying hardware and software infrastructures, as well as control throughout the cloud in regard to data policies, access policies, encryption methods, versioning, change control, and governance as a whole. For any organization that has strict policies or regulatory controls and requirements, this model would facilitate easier compliance and verification for auditing purposes versus the more limited controls and views offered via a public cloud. In cases where contracts or regulations stipulate locality and limitations as to where data and systems may reside and operate, a private cloud ensures compliance with requirements beyond just the contractual controls that a public cloud might offer, which also would require extensive reporting and auditing to validate compliance.

Control over systems: With a private cloud, the operations and system parameters of the cloud are solely at the discretion of the controlling organization. Whereas in a public cloud model an organization would be limited to the specific offerings for software and operating system versions, as well as patch and upgrade cycles, a private cloud allows the organization to determine what versions and timelines are offered without the need for contractual negotiations or potentially increased costs if specific versions need to be retained and supported beyond the time horizon that a public cloud is willing to offer.

Proprietary data and software control: Whereas a public cloud requires extensive software and contractual requirements to ensure the segregation and security of hosted systems, a private cloud offers absolute assurance that no other hosted environments can somehow gain access or insight into another hosted environment.

A *community cloud* is a collaboration between similar organizations that combine resources to offer a private cloud. It is comparable to a private cloud with the exception of multiple ownership and/or control versus singular ownership of a private cloud.

A *hybrid cloud* combines the use of both private and public cloud models to fully meet an organization's needs.

The cloud infrastructures consist of two or more separate (private, group, or public) cloud facilities, which are self-contained but interconnected by structured or proprietary data portability technologies (e.g., cloud busting for load balancing between clouds).

These are the key features of the hybrid model:

Split systems for optimization: With a hybrid model, a customer has the opportunity and benefit of splitting out their operations between public and private clouds for optimal scaling and cost effectiveness. If desired by the organization, some parts of systems can be maintained internally while leveraging the expansive offerings of public clouds for other systems. This can be done for cost reasons, security concerns, regulatory requirements, or to leverage toolsets and offerings that a public cloud may provide that their private cloud does not.

Retain critical systems internally: When a company has the option to leverage a public cloud and its services, critical data systems can be maintained internally with private data controls and access controls.

Disaster recovery: An organization can leverage a hybrid cloud as a way to maintain systems within its own private cloud but utilize and have at its disposal the resources and options of a public cloud for disaster recovery and redundancy purposes. This would allow an organization to utilize its own private resources but have the ability to migrate systems to a public cloud when needed, without having to incur the costs of a failover site that sits idle except when an emergency arises. Because public cloud systems are only used in the event of a disaster, no costs would be incurred by the organization until such an event occurs. Also, with the organization building and maintaining its own images on its private cloud, these same images could be loaded into the provisioning system of a public cloud and be ready to use if and when required.

Scalability: Along the same lines as disaster recovery usage, an organization can have at the ready a contract with a public cloud provider to handle periods of burst traffic, either forecasted or in reaction to unexpected demand. In this scenario, an organization can keep

its systems internal with its private cloud but have the option to scale out to a public cloud on short notice, only incurring costs should the need arise.

13.3.2 Cloud Shared Considerations

Several aspects of cloud computing are universal, regardless of the particular service category or deployment model.

- Interoperability: Interoperability is the ease with which one can move or reuse components of an application or service. The underlying platform, operating system, location, API structure, or cloud provider should not be an impediment to moving services easily and efficiently to an alternative solution. An organization that has a high degree of interoperability with its systems is not bound to one cloud provider and can easily move to another if the level of service or price is not suitable. This keeps pressure on cloud providers to offer a high level of services and be competitive with pricing or else risk losing customers to other cloud providers at any time. With customers only incurring costs as they use services, it is even easier to change providers with a high degree of interoperability because long-term contracts are not set. Further, an organization also maintains flexibility to move between different cloud hosting models, such as moving from public to private clouds, and vice versa, as its internal needs or requirements change over time. With an interoperability mandate, an organization can seamlessly move between cloud providers, underlying technologies, and hosting environments, or it can split components apart and host them in different environments without impacting the flow of data or services.
- Performance, Availability, and Resiliency: The concepts of performance, availability, and resiliency should be considered de facto aspects of any cloud environment due to the nature of cloud infrastructures and models. Given the size and scale of most cloud implementations, performance should always be second nature to a cloud unless it is incorrectly planned or managed. Resiliency and high availability are also hallmarks of a cloud environment. If any of these areas falls short, then customers will not stay long with a cloud provider and will quickly move to other providers. With proper provisioning and scaling by the cloud provider, performance should always be a top

concern and focus. In a virtualized environment, it is easy for a cloud provider with proper management to move virtual machines and services around within its environment to maintain performance and even load. This capability is also what allows a cloud provider to maintain high availability and resiliency within its environment. As with many other key aspects of cloud computing, SLAs will determine and test the desired performance, availability, and resiliency of the cloud services.

- Portability: Portability is the key feature that allows data to easily and seamlessly move between different cloud providers. An organization that has its data optimized for portability opens up enormous flexibility to move between different providers and hosting models, and it can leverage the data in a variety of ways. From a cost perspective, portability allows an organization to continually shop for cloud hosting services. Although cost can be a dominant driving factor, an organization may change providers for improved customer service, better feature sets and offerings, or SLA compliance issues. Apart from reasons to shop around for a cloud provider, portability also enables an organization to span its data across multiple cloud hosting arrangements. This can be for disaster recovery reasons, locality diversity, or high availability, for example.
- Service Level Agreements (SLAs): Whereas a contract will spell out the general terms and costs for services, the SLA is where the real meat of the business relationship and concrete requirements come into play. The SLA clearly sets out minimum standards for operation, accessibility, procedures, customer satisfaction, security checks and requirements, auditing and monitoring and potentially many other fields that determine enterprise and performance. Failure to meet the SLA requirements will give the customer either financial benefits or form the basis for contract termination if acceptable performance cannot be rectified on behalf of the cloud provider.

13.3.3 Security

Of course, every system or application's security should always be a top priority. There will be a lot of management and stakeholder apprehension about using newer technologies in a cloud setting, and many would be uncomfortable with the concept of having corporate and confidential data stored in private data centers and not under direct control of internal IT

personnel and hardware. Different software and systems will have their own specific security standards and controls, depending on company policies and any legislative or contractual requirements. Another challenge exists with large cloud environments that likely have very strong security controls but will not publicly document what these controls are so as not to expose themselves to attacks. This is often mitigated within contract negotiations through nondisclosure agreements and privacy requirements, although this is still not the same level of understanding and information as an organization would have with its own internal and proprietary data centers.

The main way a cloud provider implements security is by setting baselines and minimum standards, while offering a suite of add-ons or extensions to security that typically come with an additional cost. This allows the cloud provider to support a common baseline and offer additional controls on a per-customer basis to those that require or desire them. On the other hand, for many smaller companies and organizations, which would not typically have extensive financial assets and expertise, moving to a major cloud provider may very well offer significantly enhanced security for their applications at a much lower cost than they could get on their own. In effect, they are realizing the economies of scale, and the demands of larger corporations and systems will benefit their own systems for a cheaper cost.

13.3.4 Privacy

Cloud privacy needs special attention because of the large number of regulational and legal requirements, which can vary greatly in usage and location. The fact that laws and regulations can vary depending on where the data is stored (data at rest) and where it is exposed and consumed adds to the complexity (data in transit). In cloud environments, especially large public cloud systems, data has the inherent ability to be stored and moved between different locations, from within a country, between countries, and even across continents.

Often the Cloud providers have in place mechanisms for maintaining systems in geographical locations that comply with customer requirements and rules, but Cloud Security Professional is responsible for verifying the proper functioning of these mechanisms. The customer and the cloud provider must clarify contractual requirements, but strict SLAs and the ability to verify compliance are important also. In particular, European countries have strict privacy

regulations that a company must always be cognizant of or else face enormous penalties that many other countries do not have; the ability of the cloud provider to properly enforce location and security requirements will not protect a company from sanctions and penalties for compliance failure because the burden resides fully on the owner of the application and the data held within.

13.3.5 Design and Apply Data Security Strategies

Several toolsets and technologies are commonly used as data security strategies:

- Encryption
- Hashing
- Key management
- Tokenization
- Data loss prevention
- Data de-identification
- Application of technologies
- Emerging technologies

These range from the encryption of data and the prevent companies of unauthorized access, to the masking and tokenization of data to render it protected in the event that it is leaked or accessed.

Design and Plan Security Controls

The Cloud Protection Professional should focus on many aspects, as described here, in order to achieve a sound security strategy and overall governance.

- Physical and Environmental Protection: this applies to all physical devices and components of infrastructure. The actual data centre's physical assets include servers, refrigeration units, power distribution systems, network equipment, physical racks, cables, as well as real-world physical equipment and auxiliary systems in the facilities, such as battery backups, power lines, power generators, fuel tanks and the

surrounding peripheries. The power and network connections depend on the data center and also the end-point for users and customers, such as mobile systems, workstations, laptops, tablet systems and any other customer systems, outside the data centre property.

- System and Communication Protection: Whilst the cloud infrastructure is virtualized to the user, the underlying hardware and structures are the same as those in a conventional data center. Depending on the type of cloud hosting model employed, there are varying degrees of customer exposure and responsibilities. While the cloud provider is responsible for the underlying hardware and network regardless of cloud service model, the remaining services and security responsibilities either lie with the customer or are split between the customer and cloud provider. It is incumbent on the Cloud Security Professional to clearly know the demarcation lines between the customer's and cloud provider's responsibilities, which should be clearly articulated in the contracts and SLAs. As with any application, the protection of data is a primary concern, and different methods are needed for different states of data:
    - Data at rest: the key data security at rest is through encryption technology.
    - Transit data: with transit data, network isolation and the use of encrypted transport mechanisms like TLS are the principal methods of security.
    - Used data: Used data is secured by the use of encrypted systems, digital signatures, dedicated network pathways through secure API calls and web services.
- Virtualization Systems Protection: The components and systems that make up the virtualized infrastructure are the most obvious and attractive targets for attackers. The management plane, with full control over the environment and exposed APIs for allowing administrative tasks, is the most visible and important aspect to fully protect. It is made up of a series of APIs, exposed function calls and services, as well as web portals or other client access to allow its use. Any and all of these points are potential vulnerabilities. The Cloud Security Professional needs to develop a holistic picture of threats and vulnerabilities at each level and for each component of the management plane. All administrative access must be tightly controlled as well as regularly and comprehensively audited. Very detailed logging should take place not only at the level of each piece of the virtualization infrastructure, but also at the level of the web portal

or wherever the client accesses the management plane. All logs should be captured in a way where they are removed, indexed, and evaluated away from the actual system itself. This allows log preservation if the actual component is compromised, with sufficient administrative access to modify or delete the logs that are local in nature.

- Identification, Authentication, and Authorization in a Cloud Infrastructure: Like applications, cloud systems require identification, authentication, and authorization. However, this need is also extended to include nonperson entities such as devices, clients, and other applications. Federation is another important aspect of cloud computing, especially with public clouds that have a large number of customers and users. This enables the use of "home" or "native" systems, without the need of a user base to identify and authenticate the provider. A common base is used by various organisations to allow applications to accept their credentials while retaining their autonomy so that they can integrate their identity systems.

  The typical relationship flow between the user, identity provider, and relying party is shown in Figure 13-4.

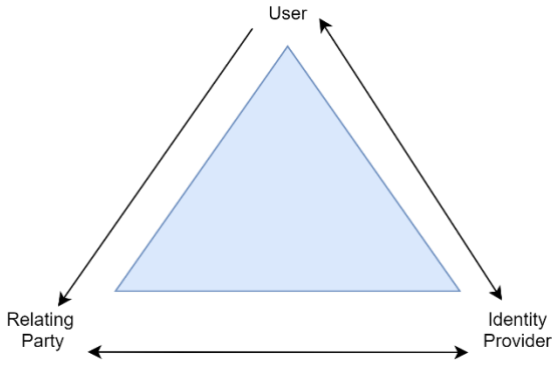


Figure 13-4: The relationship flow within a federated system [45]

13.4 SmartGrids (Scada systems)

Smart grids can include OT or industrial control systems (ICS). ICS refers to a range of operational and control systems, like SCADA, distributed control systems (DCS) system, and other control system configurations, including programmable logic control systems (PLCs), typically applied in industrial applications. Figure 13-5 shows a plan for developing the smart grid cybersecurity.

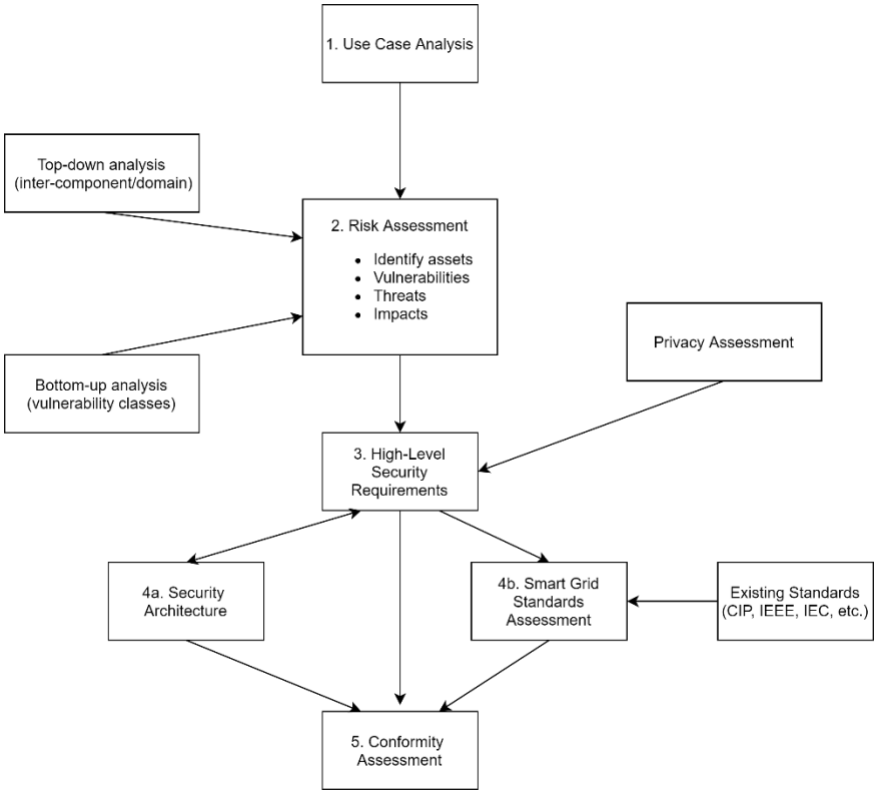


Figure 13-5: Smart Grid Cybersecurity Development Strategy [46]

Three types of cyber-physical attacks can be distinguished:

1. Cyber-informed physical attacks. When an adversary uses information obtained by cyber means to prepare and conduct a more effective physical assault. An enemy, for example, has chosen to destroy components inside a substation while not knowing which substation or components will have the greatest effect. He could physically attack a substation and specific lines if he could access sensitive information or aggregate unprotected data through cyber means that told him that a particular substation was on a very congested route and which lines were at full capacity. This could have a much greater effect than a random substation assault.
2. Cyberattacks that enhance the effects of physical attacks. A cyberattack is used by an attacker to amplify or exploit the effects of a physical attack. An adversary tampering with the integrity of protective relay settings before launching a physical attack on power lines is one example. Although the original settings were intended to contain the effects of a fault, the changed settings allow the fault to cascade through the entire network. Another example is when an adversary conducts denial-of-service

attacks on the availability of networks and facilities that enable repair operations after a physical attack. These attacks obstruct reconstruction, resulting in longer outages.

3. Physical damage caused by a cyber-system. An attacker uses a cyber-system to exploit physical equipment in order to cause physical damage. A natural gas generator's burner control system is an example of this. An adversary or a reckless operator may try to ignite the natural gas flow without an ignition source present in this situation. The adversary could ignite the ignition source when the burner device fills with natural gas, resulting in an explosion.

13.4.1 Logical Security Architecture

Millions of additional components will be added to the electric grid as a result of smart grid technologies. Many of these components will be crucial to interoperability and reliability, will interact bi-directionally, and will be responsible for ensuring the security, integrity, and availability (CIA) of power systems. The following is a list of CIA definitions:

- *Confidentiality: Keeping permitted controls on information access and disclosure in place, including safeguards for personal privacy and proprietary data. Unauthorized disclosure of information constitutes a breach of confidentiality.*
- Integrity refers to the prevention of unauthorized information alteration or destruction, as well as the assurance of information non-repudiation and authenticity. The unauthorized alteration or destruction of information is referred to as a loss of integrity.
- Availability: Ensuring that information can be accessed and used in a timely and accurate manner. The obstruction of access to or use of information or an information system is referred to as a lack of availability.

13.4.2 Logical Security Architecture Key Concepts and Assumptions

Since threats and technologies change, a smart grid's conceptual security architecture is still changing. The architecture subgroup established the following core concepts and assumptions as the logical security architecture's base.

- **Defense-in-depth strategy:** Protection can be applied in layers, with one or more security measures introduced at each layer. The aim is to reduce the chances of one of the defense components being compromised or bypassed. The fundamental concepts are that individuals, method, and technology are all required; none of these elements can be eliminated on their own.
- **Defense-in-breath strategy:** Product design and production, manufacturing, packaging, assembly, system integration, delivery, operations, maintenance, and retirement are all security activities that are planned across the system, network, or subcomponent life cycle. Throughout the life cycle, the aim is to define, control, and reduce the risk of exploitable vulnerabilities.

- **Power system availability:** The primary goal of power system engineering and operations is to ensure that electricity is delivered safely and reliably. The architecture and capabilities of existing power systems have been effective in providing this availability for defense against inadvertent behaviour and natural disasters. Current power system capabilities may be used to resolve cybersecurity concerns.

- **Microgrids:** Implied hierarchy in availability and resilience eliminates potential peer-to-peer negotiations between microgrids. According to microgrid models, availability begins in a local microgrid, and stability is achieved by aggregating and interconnecting such microgrids. Microgrids may act as islands or as part of a larger network; islands are essential where vital operations must be sustained.

- **WASA (Wide Area Situation Awareness):** WASA is frequently exchanged between business entities; such information should be defined and protected in accordance with SOA security principles. Exchange of WASA between provider and aftermarket customer (Co-op or Aggregator), utility and emergency management, or adjacent bulk providers are examples of such interactions.

13.4.3 INL National SCADA Test Bed Program (NSTB): Control System Security Assessment

The test bed assessment procedure is extremely flexible, and it can be adapted to the shared needs of the industry partners. The steps in a standard method are as follows:

- Create a contract that specifies the working arrangement (scope, staff, equipment, services, and cost sharing) and guarantees the security of sensitive data.
- Work with an industry partner to obtain equipment and training.
- With the help of an industry associate, set up equipment.
- Run experiments to discover cyber-vulnerabilities.
- Provide an industry partner with a comprehensive test study.
- Create a report that can be distributed to Web forums, conferences, and user groups.

The aim of the NSTB program is to share the knowledge gathered through the evaluations with those who may be impacted. The program collaborates with the industrial partner to decide what knowledge gathered or extracted from the evaluation process can be shared outside of the relationship. Without the written permission of the industrial partner, no information is disclosed.

13.5 IoT (SmartCities)

Smart cities are an interconnected initiative of emerging technology aimed at transforming the environment into a digital one. They're designed to be a one-stop shop for city people. Smart cities have six key characteristics, according to the EU (2007): smart economy, smart people, smart mobility, smart governance, smart environment, and smart living. The characteristics of a smart city are summarized in Figure 13-6. A dynamic market encourages new business opportunities, creativity, and foreign investments in the smart economy. People who are smart are essentially social and human capital. They are educated, lifelong learners, social, adaptable, innovative, open-minded, and eager to contribute to society. Transportation and information and communication technologies (ICTs) are two components of smart mobility. It provides secure transportation, seamless accessibility, and ICT infrastructure availability. Smart governance, or e-democracy, guarantees citizens' interest in

policy and decision-making. It promotes the use of public and social services. The key focus of smart environment is natural resource management. It provides a pollution-free, environmentally friendly, and safe climate. Smart living is about improving one's quality of life. There are a variety of social, cultural, health, comfort, protection, and individual factors that go into it.

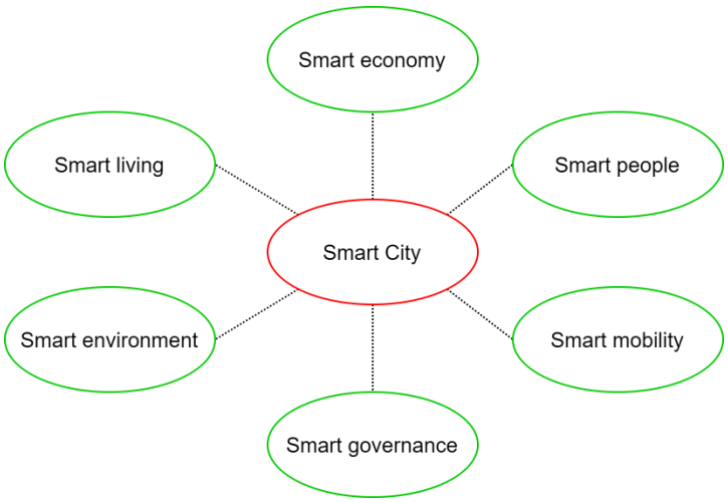


Figure 13-6: The characteristics of a smart city

13.5.1 IoT Applications

Smart city infrastructures are expected to deploy a wide range of IoT-based applications including:

- Smart home,
- Smart grid,
- Intelligent Transport System Applications (ITS), and
- Real-Time Monitoring and Safety Alert (RTMSA).

A summary of these applications is illustrated in Figure 13-7.

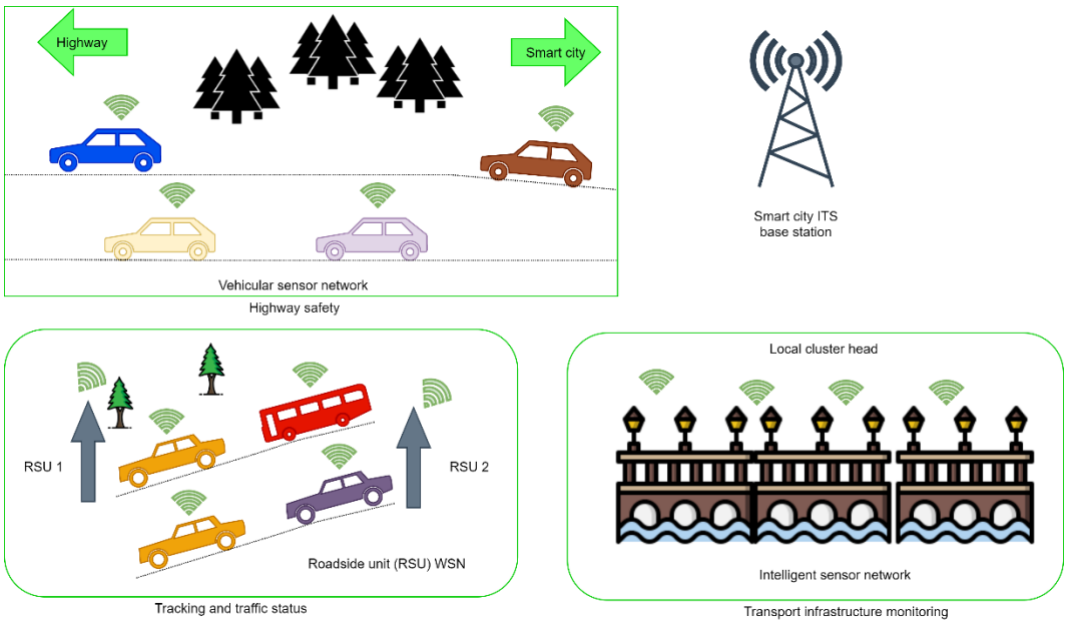


Figure 13-7: A summary of applications in the ITS [47]

13.5.2 Possible Attacks

Physical and cyber attacks are the two most common ways to target IoT. Physical damage, physical tampering, environmental tampering, and physical intrusion are the most common physical attacks. These attacks are summarized in Table 13-2. Invasive or noninvasive physical tampering is possible. Invasive attacks enable attackers to gain access to and change the devices' internal structure. Attackers obtain physical information without physically manipulating sensors in a noninvasive attack. Physical intrusions into the deployment area will set the stage for the other three forms of attacks. There are several possibilities, including the use of fake sensors to corrupt data, the use of unauthorized IoT devices to collect sensitive information, and the use of malicious sensor nodes to disrupt routing processes.

Table 13-2: Summary of common physical-attacks [47]

Physical-attack type	**Impacts on WSNs**	**Remarks**
Physical destructions	Sensing unavailable	The prime targets are the sensors that are deployed in

		an open environment: ITS, RTMSA applications
Physical tampering	Sensor malfunctioning (invasive), physical information disclosed (noninvasive, side-channel attacks))	All types of applications can be targeted. Physical intrusion in the deployment area is a prerequisite
Environment tampering	Unreliable/misleading data	The prime targets are the sensors that are deployed for monitoring and control of environmental parameters: smart home and substation applications
Physical intrusion	Creates environment for physical and cyber-attacks, deploys unauthorized sensing devices	All types of applications can be targeted

The key categories of cyber-attacks are routing attacks, denial of service and cyber intrusion. Routing attacks can take various forms: sybil, wormhole, sinkhole and selective transmission. Fake identities of the same nodes are created in the Sybil attack. Under multiple identities the same compromised nodes work. Several malicious nodes work together during the wormhole attack. They eavesdrop, move and broadcast in a different area. A fake base station is built in the sinkhole attack to direct all traffic towards a compromised entity. Data can therefore be lost and manipulated. In selective forward attack, malicious sensor nodes transmit only a selected portion of the messages received. This is to the detection process. Various types of DoS attacks are available. The idea is that the network resources are unnecessarily consumed and useful operations blocked. A distributed DoS (DDoS) may exist where several nodes collaborate and cause a service failure. Hello Flood Angriffe is one example of DoS Attack, which transmits Hello messages using high power for network use.

The attack is driven by the cloning of nodes. Cyber intrusion allows unauthorized sensor access. The common cyber attacks can be found in Table 13-3.

Table 13-3: Summary of common cyber-attacks [47]

Cyber-attack type	Impacts on WSNs	Remarks
Routing attacks: Sybil, wormhole, sinkhole, selective forwarding	Adversaries capture actual data (break of confidentiality), corrupts data (break of integrity), misleads sensing operations (break of availability)	The prime targets are the large-scale sensor networks: ITS, RTMSA applications
Service denial: DoS, DDoS, flooding	Unnecessary consumption of resources: battery (affected lifetime), bandwidth, processor, add delays in routing (affected time criticality)	The prime targets are the large-scale sensor networks: ITS, RTMSA applications
Insider attack	Forgery, initiates other types of attacks	All types of applications can be targeted
Cyber intrusion	Compromise, privacy leak, initiates other types of attacks	All types of applications can be targeted

13.5.3 Solution Approaches

For highly motivated attackers, Smart cities can be an easy goal. Financial benefits, privacy, blocking, vandalism and sabotage are the most common reasons. These motivations can lead to cyber and physical attacks of both kinds. In smart cities, wireless sensor systems (WSNs) have three key security requirements, similar to other communication networks:

- Confidentiality: protection from unauthorized parties for sensitive data and relevant information.
- Integrity: prevention of unauthorized data and related information modification.
- Availability: ensuring uninterrupted access for authorized parties to sensed data and relevant information.

At least one of these fundamental requirements is violated in a successful attack. The implementation of thoroughly designed security solutions can achieve these requirements. In this context, we use the term "relevant information" to refer to sensitive information other than sensory data like the energy status of sensors. Security solutions for WSNs include energy usage, communication, computer overheads, storage capacity and costs, among other design factors. The research challenge is always simultaneously considering low energy, low overhead and low cost. We discuss solutions that are appropriate for smart cities in the following:

- **Cryptography** is the most important method of enhancing cyber security, confidentiality and message integrity. Two basic encryption techniques exist: symmetrical key encryption and public key encryption. The same key is used for encrypting and decrypting messages in symmetric encryption. Two various keys are used for encrypting and decrypting messages in public key encryption. These keys are respectively referred to as public and private keys. Because of the resource constraints of the sensor nodes, the WSN context is a difficult environment to implement crypothec technology. The TTP may either be a main distribution center (KDC) (CA). A KDC has the function of distributing keys among communicator parties.

  WSNs are used for continuous and long term sensing operations in an intelligent city climate. In such cases, cryptographic keys should be regularly refreshed. Dynamic key management is governed by three performance metrics: I memory needs, (ii) bandwidth use and (iii) power usage. The need for memory depends on the size and number of security credentials that can be stored, such as keys and identities. The use of bandwidth depends on the size and number of messages that are exchanged for key management. Power consumption depends on computer processes and key management message transmission profiles. The above efficiency metrics effectively introduce limitations to the cryptographic architecture of WSNs with limited resources.

- **Intrusion Detection System:** As intrusion events can in many ways lead to dangerous activities, deploying IDS is an efficient safety measure. In WSNs, malicious sensor nodes are primarily identified by IDSs. IDSs use battery and computer tools. This entails extra battery life and computer overheads. Additional resources are needed for sensor nodes to maintain the sensing quality level. As WSNs work under resource-limited constraints, it is difficult to design and enforce IDSs. An IDS decides on four different types: (i) true positive, (ii) true negative, (iii) false positive, and (iv) false negative. These decisions are taken through conduct study of sensor nodes. Decisions are constructive when an IDS categorizes an interference behaviour. Decisions are negative when an IDS categorizes behaviour. When an IDS is acting right, "real" decisions are made. The 'fake' decisions, on the other hand, are decided if an IDS is wrong. A low rate of false decision is generated by an effective IDS. In several cases, IDSs are classified. These categories are specifically applied in their implementation. During behavioral analysis two simple identification methods are used: I anomaly and (ii) signature. The conduct of each node is compared to a predefined normal activity in the anomaly-based detection. If the variance reaches an intrusion value, it is observed. An intrusion. Using traffic analysis, the behaviour of sensor nodes is extracted.
    - **Watchdog System:** The watchdog system definition was derived from trust administration in WSNs. It was initially proposed in MANET and subsequently adopted in WSNs. Although the purpose is the same, the operating process of the IDSs is different. A WSN selects several intermediate nodes as watchdogs. In promiscuous manner, every watchdog overhears the message of its neighbours. It reports misbehaviour to the neighbour if it detects any irregularity in message transmission or corrupt message from a neighbour.
    - **Game Theoretic Deployment:** The theory of gaming is a mechanism for enhancing security both in cyberspace and in physical fields. It is primarily used to calculate optimal protection implementation strategies in resources. The protection of a given amount of capital is maximized in an optimal strategy. Game theory is particularly helpful in the development of cyber physical intrusion management solutions

13.6 Referencing

[43] Wrightson, T. (2014). *Advanced persistent threat hacking: the art and science of hacking any organization*. McGraw-Hill Education Group.

[44] Souppaya, M., & Scarfone, K. (2016). User's Guide to Telework and Bring Your Own Device (BYOD) Security. *NIST Special Publication*, *800*, 114.

[45] 2019. CCSP Certified Cloud Security Professional All-in-One Exam Guide (2sd Edition). Daniel Carter. McGraw-Hill Education.

[46] Lee, A., & Brewer, T. Guidelines for Smart Grid Cyber security, Volume 1, Smart Grid Cybersecurity Strategy, Architecture, and High Level Requirements, NISTIR 7628 Revision 1. Available at: http://dx.doi.org/10.6028/NIST.IR.7628r1

[47] Stouffer, K., Falco, J., & Scarfone, K. (2011). Guide to industrial control systems (ICS) security. *NIST special publication*, *800*(82), 16-16. Available at: http://dx.doi.org/10.6028/NIST.SP.800-82r2

[48] Song, H., Fink, G., & Jeschke, S. (2017). *Security and privacy in cyber-physical systems*. Wiley-IEEE Press.

14 Bibliography

[1] W. B. L. B. M. D. &. B. A. K. Stallings, Computer security: principles and practice, (pp. 978-0). Upper Saddle River, NJ, USA: Pearson Education., 2018, 4 edition.

[2] M. Bishop, Computer Security Art and Science", 2nd Edition, Addison-Wesley Professional, ISBN: 9780134097145, November 2018.

[3] C. Easttom, Computer security fundamentals, Pearson IT Certification., (2019).

[4] CISSP, Official (ISC)² guide to the CISSP CBK, Fourth Edition.

[5] ISO/IEC 15288:2008, «Online Browsing Platform (OBP),» [En línea]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:21827:ed-2:v1:en. [Último acceso: 24 November 2020].

[6] ISO/IEC 21827:2008, «Online Browsing Platform (OBP),» [En línea]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:21827:ed-2:v1:en. [Último acceso: November 2020].

[7] R. Ross, M. McEvilley y J. Oren, «Engineering Principles for Information Technology Security (A Baseline for Achieving Security), Revision A,» *NIST Special Publication (SP) 800-27 Revision A,* November 15, 2017.

[8] «Cisco Secure Development Lifecycle,» [En línea]. Available: https://www.cisco.com/c/dam/en_us/about/doing_business/trust-center/docs/cisco-secure-development-lifecycle.pdf. [Último acceso: 23 November 2020].

[9] International Organization for Standararization, «ISO/IEC 154408,» [En línea]. Available: https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html. [Último acceso: November 2020].

[10] «The ISO 27000 Directory,» [En línea]. Available: http://www.27000.org/. [Último acceso: 23 November 2020].

[11] Open Web Application Security Project, «WEB SECURITY TESTING GUIDE 4.1,» OWASP.

[12] G. Rajendran, «Security in the embedded system: Attacks and Countermeasures,» *International Conference on Recent Trends in Computing, Communication and Networking Technologies,* 2019.

[13] W. Stallings, Information Privacy Engineering and Privacy by Design", 1/e,.

[14] K. J. V. O. P. V. S. Menezes AJ, Handbook of applied cryptography..

[15] W. Stallings, Cryptography and Network Security: Principles and Practice (7th Edition). [Reference chapters 11 & 13], USA: Pearson, , 2016.

[16] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Cryptographic_hash_function.

[17] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Secure_Hash_Algorithms.

[18] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Digital_signature.

[19] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Chain_of_trust.

[20] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Digital_rights_management.

[21] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Cryptanalysis.

[22] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Attack_model.

[23] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/Replay_attack.

[24] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/MD5.

[25] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/SHA-1.

[26] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/SHA-2.

[27] «Wikipedia,» [En línea]. Available: https://en.wikipedia.org/wiki/SHA-3.

[28] W. Stalling, Network Security Essentials: Applications and Standards,, Fourth Edition.

[29] M. M. S. .. M. M. Jernigan, CompTIA Security Certification Guide., (2018).

[30] «Network Architecture Diagrams,» [En línea]. Available: https://www.uml-diagrams.org/network-architecture-diagrams.html.

[31] CLOUDFLARE, « TLS 1.3 - Enhanced Performance, Hardened Security,» [En línea]. Available: https://www.cloudflare.com/it-it/learning-resources/tls-1-3/.

[32] NIST, «Information Technology Laboratory. Computer Security Resource Center,» [En línea]. Available: https://csrc.nist.gov/glossary/term/security_concept_of_operations.

[33] K. A. G. T. &. M. K. Scarfone, Sp 800-61 rev. 1. computer security incident handling guide., 2008.

[34] A. &. M. V. Bahga, Internet of Things: A hands-on approach. Vpt., 2014.

[35] tutorialspoint, «Learn Software Testing terms,» [En línea]. Available: https://www.tutorialspoint.com/software_testing_dictionary/dynamic_testing.htm.

[36] L. Johnson, Security Controls Evaluation, Testing, and Assessment Handbool, Elsevier, 2016.

[37] NIST Special Publication 800-15, Technical Guide to Information Security Testing And Assessment..

[38] O. G. t. t. C. CBK, Certified Information Systems Security Professional, Adam Gordom, Fourt Edition.

[39] incibe, «Secure Mobile Application. Introduction. Unit 5».

[40] OWASP Foundation, “OWASP CODE REVIEW GUIDE V1.1,” 2008.

[41] A. Shostack, Threat modeling desingning for security, WILEY, 2014.

[42] N. Mead, «Seven Principles for Software Assurance,» Software Engineering Institute, 24 october 2016. [En línea]. Available:

https://insights.sei.cmu.edu/sei_blog/2016/10/seven-principles-for-software-assurance.html. [Último acceso: 17 December 2020].

[43] T. Wrightson, « Advanced persistent threat hacking: the art and science of hacking any organization.,» McGraw-Hill Education Group., 2014.

[44] M. &. S. K. Souppaya, « User's Guide to Telework and Bring Your Own Device (BYOD) Security.,» NIST Special Publication, 800, 114., 2016.

[45] D. Carter, CCSP Certified Cloud Security Professional All-in-One Exam Guide (2sd Edition)., McGraw-Hill Education., 2019.

[46] A. &. B. T. Lee, Guidelines for Smart Grid Cyber security, Volume 1, Smart Grid Cybersecurity Strategy, Architecture, and High Level Requirements., Publication(NIST SP) 800-114.

[47] H. F. G. &. J. S. Song, Security and privacy in cyber-physical systems., Wiley-IEEE Press., 2017.

[48] K. F. J. &. S. K. Stouffer, «Guide to industrial control systems (ICS) security. NIST special publication, 800(82), 16-16.,» NIST Sepecial Publication 800-82, Revision 2 , 2015.