COMPUTER ENGINEERING SERIES

Computer Science Security

Concepts and Tools

Ameur Salem Zaidoun

iSTE WILEY

Computer Science Security

To the memory of my parents
To my wife
To my sons, Mohamed Bairam and Iyed, and my daughter, Elaa

To all the supporters of both peace and liberty in the world

Series Editor
Jean-Charles Pomerol

Computer Science Security

Concepts and Tools

Ameur Salem Zaidoun

ISTE

WILEY

First published 2022 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.

ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

Library of Congress Control Number: 2022932502

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISBN 978-1-78630-755-2

Contents

List of Acronyms

ACL	Access Control List
AES	Advanced Encryption Standard
AH	Authentication Header
CA	Certification Authority
CLI	Command Line Interface
DES	Data Encryption Standard
DMZ	DeMilitarized Zone
DoS	Denial of Service
DSA	Digital Signature Algorithm
ESP	Encapsulating Security Payload
H-IDS	Host-based IDS
IDEA	International Data Encryption Algorithm
IDS	Intrusion Detection System
IKE	Internet Key Exchange
IoT/IoE	Internet of Things/Internet of Everything

IPS	Intrusion Prevention System
ISAKMP	Internet Security Association and Key Management Protocol
L2F	Layer Two Forwarding
L2TP	Layer Two Tunneling Protocol
MARION	*Méthodologie d'Analyse de Risques Informatiques Orientée par Niveaux* (Methodology for analysis of computing risks oriented by levels)
MD5	Message Digest 5
MEHARI	*MÉthode Harmonisée d'Analyse de RIsques* (Harmonized method for analysis of risks)
N-IDS	Network-based IDS
NAT	Network Address Translation
PAT	Port Address Translation
PPTP	Point-to-Point Tunneling Protocol
RSA	Rivest–Shamir–Adleman
SA	Security Association
SDN	Software-Defined Network
SHA1	Secure Hash Algorithm 1
SSH	Secure SHell
SSL	Secure Sockets Layer
TLS	Transport Layer Security
VPN	Virtual Private Network

Introduction

Since antiquity, humans have sought security for themselves, reserving great importance for this need, an importance that is justified by the need to protect themselves against external factors.

With the appearance of computing and its full expansion, the problem of security is evident: it is a new multifaceted problem that affects all areas of activity, users and equipment involved in the field of computing.

Security includes numerous facets:

– The first facet concerns the energetic aspect. It consists of providing and maintaining the necessary energy for the proper functioning of the computer system.

– The second facet deals with the physical aspect. It consists of physically securing the location and material infrastructure of the computer system, such as the cable and wireless systems of communication, the intermediary equipment for interconnection, transmission and security (repeaters, hubs, switches, routers, firewalls, etc.), as well as terminal processing equipment for both workstations and servers. Physical security allows the computer system to be secured against natural factors, such as flooding and fires, and human factors, such as theft.

– The third facet is an aspect specific to computer security, contrary to the first two which are applicable to any industrial field that requires energy and ownership of critical materials. The objective of this facet is to resolve specific security problems related to computer technology, in order to guarantee secure use and prevent unauthorized access, on the one hand, and

protect proprietary software and data against any unauthorized use, on the other hand.

Even though no radical, ready-made security solutions exist, they continue to be developed. However, they always remain insufficient for resolving security problems. We are faced with a continuous battle between those responsible for security, on the one hand, and those who attack, hack and intrude, on the other hand.

This battle manifests itself in the working groups on each side, which is the case for the group “Anonymous” and international organizations that take an interest in security.

Attackers never cease to seek and identify faults, create appropriate means of attack to exploit the vulnerabilities in question and overcome the countermeasures that can sometimes be deployed.

This challenge requires an investment from the other side that allows us to mobilize human material resources, in order to face attacks and diverse malicious actions that threaten the security of a computer system. Numerous approaches to security – organizational, intellectual, material and software – have been invented and continuously developed, both in quantity and quality, in recent years.

Security constitutes a very important challenge, although it is neglected due to ignorance in most cases. Rising to this challenge is not a simple task because there is a lack of a radical solution or a clear and exact approach to its application. The only way to guarantee minimal security is through the combination of numerous social, cultural and technical measures. Nevertheless, these security measures remain dynamic and depend on the circumstances. We are faced with the need for a form of technological intelligence that allows the necessary modifications and corrections to be made continuously to the security rules in use.

The intervention of experts and external parties is a valuable step of capital importance in the definition and evolution of security rules. This intervention can take the form of security audits that take place throughout a company, or a direct intervention by experts during consultations for SMEs/SMIs (Small and Medium Enterprises/Small and Medium Industries).

Numerous general recommendations must be applied to guarantee the necessary minimum with respect to security:

– First, the security service or team must configure an organigram for the company, while allocating the head of this service or team, who must have a minimum level of knowledge in matters of security, with the necessary power and logistical facilities for intervention.

– Moreover, betting on the ignorance of agents and users has never been a security measure. What is needed is the establishment of a culture in matters of security through cycles of well-planned training that addresses all personnel (managers and agents), on the one hand, and the planning of charters and displays, on the other hand.

– Finally, we must apply the necessary technical security measures, depending on the company's software assets, by using the appropriate equipment and software, as well as applying security measures. It should be pointed out that the latter recommendation, although of capital importance, can be called into question and even be useless if the first two recommendations are disregarded for one reason or another.

The problem of security is not a purely professional concern that only involves companies and institutions; it is first and foremost a social question that concerns families and society in a general sense. Indeed, the use of software and Internet access permeates society and presents many challenges, especially for children who can access any information and be influenced by social networks without any supervision from their parents. To rise to this challenge, whose gravity is not negligible, two basic measures are necessary, cultural and technical, through sensibility tasks in the home, schools and the media, in addition to the development of appropriate parental control software.

This book presents numerous subjects related to security:

– First, the crux of the issue of security will be identified, and its importance, as well as the pertinent services and mechanisms will be detailed.

– Then, security faults and problems will be identified, including those that are cultural and human, as well as the material and technical issues.

– Next, security solutions at a technical and organizational level, namely antivirus, firewalls, IDSs (Intrusion Detection Systems), and different

techniques to control access, authentication and encryption, will be specified.

– Moreover, the specific security at the level of SDN (Software-Defined Networks) and IoT/IoE (Internet of Things/Internet of Everything) sensor networks will be addressed by way of the specificity of these technologies.

– Finally, this work will be completed by a section covering the management of security and, in particular, audit security, as well as the establishment of security rules.

1

General Concepts in Security

1.1. Introduction

The recent massive evolutions in computing and communication technology have produced certain side effects, namely security problems.

Indeed, the use of networks facilitates communication and likewise, the spread of tools for attack and coordination among hackers. This is exactly the case in the development of road networks and modes of transportation that cause problems related to accidents and require investment in highway safety. Moreover, accumulated expertise from the development of software has been exploited for the development of malware.

Security attacks can be classified in two ways. The first classification is subdivided by the effect of the attack: either passive attacks, also called reconnaissance attacks, which consist of observing and divulging information via a third party, or active attacks, also known as access attacks, which are characterized by malicious acts that create doubt over information, users and communication channels. The second classification is subdivided by the means of attack: a direct attack using one's own resources and identity or an indirect attack that passes through other intermediary computers.

Numerous variations of attacks continue to appear periodically. In fact, statistics have shown the volume of danger that threatens computer systems and the resulting material waste. Both of these aspects, without a doubt, make security a question of capital importance and an indispensable necessity.

Computer Science Security: Concepts and Tools,
First Edition. Ameur Salem Zaidoun.

Faced with these problems, we must invest in security with the goal of creating a particular culture on the one hand and establishing technological solutions on the other. The former principally concerns the user of the computer, disregarding how the computer is being used or the user's knowledge, whether gained through courses, trainings, forums, manuals or posters. The latter concerns the computer system itself, addressed by establishing security measures and technical solutions.

The security of a computer system includes numerous aspects, from providing energy to physical control, and finally, the management of access and software security measures. It particularly concerns information and the entities who manipulate it. The latter, the most important and the area in which security usually gets simplified, can be classified into four points that must be satisfied to confirm whether a computer system is secure or not.

1.2. Reasons for security

Computer security has taken prominence for many years now, following the massive evolution of computers and the socialization of their use. The reasons that favored the appearance of computer security as an important subject can be divided into two large subsections: the first is technical and related to services and the technology itself; the second is social and tied to the massive, diversified and generalized use of computers.

1.2.1. *Technical issues*

The development of computing, despite its advantages, has been used maliciously to threaten security. Security was initially neglected, but became necessary to face fresh problems and avoid exploitation by hackers.

1.2.1.1. *Development of software engineering*

Software engineering has experienced a great evolution marked by the use of numerous programming languages and development techniques that are integrated in the majority of software packages in one way or another; we can cite Microsoft Office as an example, as it allows a user to develop

macros (the basis for the virus known as a macro-virus, that infects Office documents).

Over time, users have acquired more and more expertise in programming. This massive development in programming opportunities and expertise has been exploited for dishonest ends, in certain cases to create attack software (analysis tools, scanning tools) or inject tools for intrusion (viruses, worms, etc.).

Nowadays, attack tools are readily available on the Internet. They are accessible to end users and do not require any qualification from them for use. They take many forms (web applications, graphic applications, etc.). Additionally, an inexperienced user cannot distinguish them from legitimate applications and may launch them due to ignorance or negligence, jeopardizing the security of their own system. Numerous bad habits encourage this problem.

1.2.1.2. *Development of networks*

We are experiencing a great development in the spread of networks, which encourages communication and information transfer. Such opportunities and services can be misused by intruders to attack or exchange suspicious information.

The disadvantages caused by networks are the byproduct of two factors. The first is the massive use of networks in many areas, which allows everyone to be connected despite their many differences. The second concerns network technology itself, which is based on TCP/IP (Transmission Control Protocol and Internet Protocol), a protocol that despite its simplicity and efficiency, presents numerous faults insofar as it allows end users to access a large variety of tools and services.

Use

Currently, we are in an era of revolution in computing and telecommunication technology. Information can cross from one end of the world to the other in a few fractions of a second thanks to the proliferation of networks, in which the Internet plays the most important role. Because of this, a computer is not isolated; it communicates with thousands of other computers every day and either explicitly or implicitly executes thousands of

programs that could be malicious for its proper functioning. This danger is reinforced by the use of accessories, and especially external drives.

Networks represent dangerous information highways, used by hackers and attack programs, and facilitate communication. Life in an isolated house is difficult, and the act of creating a road to it makes life easier, as well as attacks by robbers.

Technological uniformity (TCP/IP)

The appearance and proliferation of the Internet favored the use of the TCP/IP family of protocols, even locally (Intranet). Insofar as the TCP/IP services and protocols are known by everyone, this implies the possibility of creating an attack that targets the known faults within TCP/IP. We are now seeing technological uniformity, and even though that is beneficial for the development of computing in general, it creates an opening that can be exploited by intruders, who can focus on developing attack tools that are specific to TCP/IP and useful in attacking any network node.

TCP/IP was created for the American Department of Defense (DoD) during the Cold War. The objective at the time was to deploy a robust communication system that could work in critical circumstances, and even if it was partially deteriorated. The proposed solution was a system with "intelligence at the terminals", which is contradictory to the principle of security, since it puts a large selection of functions within reach of end users. With the proliferation of the Internet, the problem of security became more and more imposing given the lack of a blanket solution.

1.2.2. *Social factors*

Social factors are as important as technical factors, given that we are also seeing a virtual world with all of its own characteristics developed in parallel to the real world. A new jargon has been created, for example, cybercriminal, cyber-bullying, etc. – terms that reflect the problems and characteristics inherent to the virtual world and that take place in social networks in particular.

We now speak of an information society marked by the proliferation of computing culture and the massive use of computers and mobile devices, which are considered banal tools accessible to everyone. In turn, this has

generated an environment of connected objects (IoT and IoE). The absence of rules to manage it or security charters for companies to follow facilitates the spread of information: we can thus also speak of social engineering.

1.2.2.1. *Information society*

With the proliferation and the expanded use of computers and the Internet, we are now part of an information society, outside of which no one can exist. The expression "information society" designates a society in which information technologies play a central role.

1.2.2.2. *Social engineering*

The use of social engineering technology offers a way to extract confidential information from security directors and administrators.

"Social engineering", still called "psychological subversion" in French, is a practice that consists of taking advantage of one or several people's trust with the principal objective of obtaining confidential information[1].

This attack can be carried out directly and in person, or through social networks and other means of communication. The victims are generally administrators of computer networks.

1.3. Security attacks

There are two possible classifications of attacks. The first classification is based on the nature of the act itself (passive or active). The second classification is based on the means of attack (direct or indirect).

1.3.1. *Passive/active classification of attacks*

The attacks that allow security to be called into question can be classified into two types: passive and active, depending on their impact and the degree of involvement in the communication process. We can also speak of reconnaissance attacks and access attacks.

1 Magali Jakusic – www.securite.teamlog.com.

1.3.1.1. *Passive attack (reconnaissance attack)*

The passive attack, also called the reconnaissance attack, consists of passively observing information or its characteristics, whether it is inside a computer or being transferred through a network. This attack can be led through a family of software called sniffers or network observers. These kinds of software are very easy to find on the Internet, and they allow, on the one hand, the characteristics and configurations of systems and networks to be scanned by exploiting the many faults and vulnerabilities in the systems and networks, and on the other hand, traffic to be analyzed so the flux of information can be extracted.

The reconnaissance attack can be implemented through a set of elementary actions:

– initial request to the target;

– ping sweep of the target network;

– scan port for active IP addresses;

– scan for flaws;

– exploitation of tools (appropriate sniffers and malware).

We can cite the following sniffers by way of example: Wireshark, WinPcap, WebSiteSniffer, SocketSniff, SmartSniff, Packetyzer, PacketViewer, PacketMon, CommView and IP Sniffer.

We can distinguish numerous scenarios:

– *Discovery of users*: this occurs when the identities of the participants in a particular communication are discovered.

– *Discovery of inter-message duration*: this occurs when the frequency of messages is discovered, which allows the nature of the application to be identified (interactive, etc.).

– *Discovery of information itself*: this is the discovery of the contents of communicated information. It is the most serious kind of passive attack.

1.3.1.2. *Active attack (access attack)*

An active attack, also known as an access attack, consists of acting upon information and/or the communication process, and is therefore more serious

than a passive attack. It involves the modification of data or messages, entering the network equipment or perturbing the proper functioning of the system and network.

The primary objectives in an access attack include:

– gathering data;

– gaining access;

– granting greater access privileges.

We can distinguish different scenarios:

– *Fraudulent connection*: this occurs when a connection is established with a server without the right to connect. This connection allows the attacker to make changes as if they were the owner of the account. They can erase documents, change the configuration, etc.

– *Denial of service*: this occurs when one or more authorized users are prevented from accessing a service. It consists of attacking the server and making it go offline, which paralyzes all access. This kind of attack is known as a DoS (Denial of Service). We can also speak of DDoS (Distributed DoS), which is an attack that happens when a principal computer manages to take control of numerous computers and uses them to attack victims so that they go offline.

– *Message alteration*: this consists of modifying the order of messages by capturing some of them before reinserting them later.

– *Interruptions/delays in communication*: this happens when a hacker manages to stop a connection or delay message access and transmission.

– *Modification/disclosure of message content*: the most serious act, which consists of modifying or publicizing the content of sent messages.

– *Port redirection*: this consists of redirecting traffic in a general manner or redirecting a response, operation or command to a third-party, unauthorized computer.

– *Man-in-the-middle*: this is an attack that allows a third party to intercede between two entities in communication, without the latter being aware.

– *Buffer overflow*: this attack consists of exploiting the buffer capacity, possible in a weakly written language such as C, with the purpose of placing a well-chosen code within the memory and creating a system or network failure.

– *IP, MAC, DHCP spoofing*: this attack allows the identity of an authorized computer (IP address, MAC) to be exploited for an attack or to redirect information.

– *Loss of data*: this represents the most expensive attack, given that once lost, data requires an enormous effort to be recuperated and restored to the same level, which is not efficient in the majority of cases and can cause very serious financial repercussions for a company. The points of vulnerability that can facilitate the loss of data are:

– email access/non-secure webmail;

– unchecked peripherals;

– cloud storage;

– detachable media;

– inappropriate access check that does not satisfy the minimal requirement for security.

1.3.2. *Direct/indirect classification of attacks*

On the Internet, an attack can directly target a victim with their own identity or resources, or can hide behind an intermediary, in some cases using the latter's identity and/or resources.

1.3.2.1. Direct attack

This is the simplest attack and also the easiest to detect. It consists of launching an attack request on a victim using their own identity (IP address, MAC address) and resources (memory, CPU and hard drive), as illustrated in Figure 1.1. These have become more and more rare, with the most serious hackers seeking to hide themselves or gain further abilities, given that users have developed ways to detect attack connections that are launched directly.

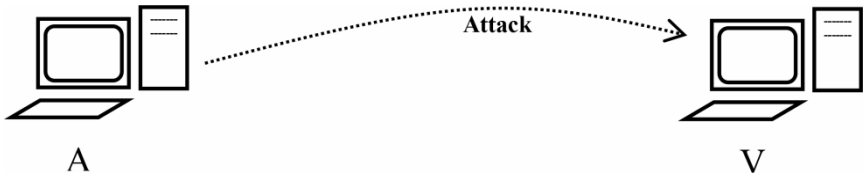

Figure 1.1. *Direct attack*

1.3.2.2. *Indirect attack*

This is the most complex attack and the most difficult to track. There are two kinds:

– The first attack is by rebound, which refers to an intermediary computer that is used for its identity or resources in an attack.

– The second attack is by response, which exploits a server as if it were an intermediary computer to attack a third one.

1.3.2.2.1. Rebound

The pirate tries to take control of the intermediary computer, which generally has better resources, as illustrated in Figure 1.2. It will be used to create a rebound such that the identity and the resources of the victim are used to attack their computer, so that the real attacker cannot be identified in any way. We speak of two attacks in a chain. New variations on attacks of this kind are based on three or more chain attacks, and in this scenario, we are dealing with two or more rebounds. The rebound is itself a victim, even though it is the starting point for the attack that targets the final victim.

It is for this reason that we must not neglect the security of a computer just because it has no use; such a computer could be used as a rebound, especially in a case where it can provide ample resources.

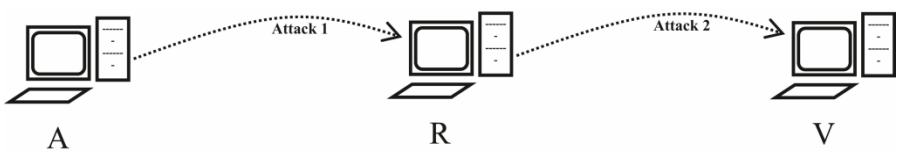

Figure 1.2. *Indirect attack by rebound*

1.3.2.2.2. Response

The attacker sends a request to a server in such a way that they send the response to a third computer, as illustrated in Figure 1.3. The attack concerns the third computer insofar as it receives a response that it did not solicit and which can introduce an error, inhibit its functioning, oblige it to react through misconduct or even take it out of service.

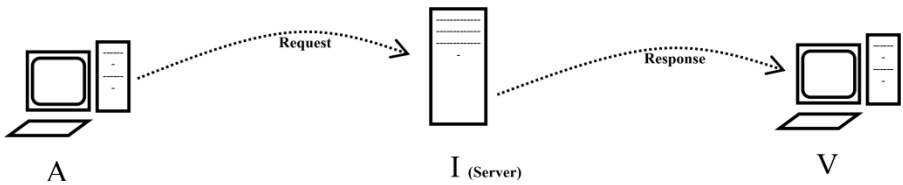

Figure 1.3. *Indirect attack by responding*

The attack by response is necessarily done through a server that will not be attacked itself, whereas the rebound attack manifests itself as two attacks in a chain.

1.3.3. *Examples of attacks*

Hackers never stop identifying or creating weaknesses in systems and networks, and here we will present a few examples.

1.3.3.1. *SQL injection*

This allows the user to connect without the password in a case where authentication is done via the following SQL request:

```
SELECT * FROM Users WHERE login='the login' AND
password='password'
```

If the user introduces "`a OR 1=1 #`" as a login and anything as the password, the request becomes:

```
SELECT * FROM Users WHERE login=a OR 1=1 # password=azerty
```

A request which is still valid, the # returns what follows as a comment. The use of the apostrophe (‘) also breaks the authentication in the case of websites developed using Jomla.

1.3.3.2. *TCP SYN*

A TCP connection is made in three steps (three-hand shake), as indicated in Figure 1.4. The computer that initiates the communication sends a request asking for a SYN C connection; the computer receiving the request sends an ACK SYN C+1 acknowledgment, as well as the SEQ S number sequence, and starts a Time Out while waiting for the acknowledgment of this ACK SEQ S+1 number sequence.

In the case where the computer that initiates the connection does not send the last message (ACK SEQ S+1) at the end of the Time Out, the other computer creates a large data file containing the context of the different connections. If the number of incomplete TCP connections goes up, the memory will be used up and the computer being attacked will go out of service. In this scenario, we speak of a TCP SYN attack.

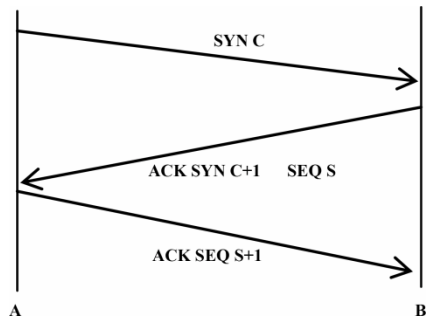

Figure 1.4. *TCP SYN attack*

1.3.3.3. *Buffer overflow*

This is a very efficient but incredibly complex attack. It consists of injecting a program by writing more information into a buffer than its size, in order to overwrite the code and insert sequences that are useful for taking control of the program being attacked.

```
int buf[20];
strcpy(buf, “a long string of code”); /*Buffer overflow*/
```

The buffer overflow error does not appear during the compilation nor systematically during the execution. It causes a crash in the case where the memory outside of the space reserved for buffering is being used by another variable. The memory used can even correspond to the entire operating system.

1.3.3.4. *Mail bombing*

This kind of attack occurs if an email account is bombarded by different messages that can, in some scenarios, be sent automatically to email addresses appearing in a contact list.

1.3.4. *Some statistics*

In this section, we will content ourselves with providing just a few key figures from the statistics obtained by the *CompTIA* (Computing Technology Industry) association[2]:

– 52% of damage is caused by human error (the human factor is the biggest, which requires sensibility training).

– 80% of attacks are done by employees and especially ex-employees, either by negligence or by irresponsibility (employee contracts should take responsibility clauses into account).

The destruction from attacks can be divided as follows:

– 44% of the damage takes the form of wasted time;

– 16% of the damage occurs as ruined software;

– 16% of the damage takes the form of lost information.

As for intrusions, we can give the following examples:

– MyDoom caused $38 million worth of losses in the USA in 2004;

– SirCam caused $1.15 million worth of losses in the USA;

– CodeRed caused $2.62 million worth of losses in the USA (250,000 infected computer systems in less than nine hours);

– Himda caused $635 million worth of losses in the USA.

2 www.comptia.org.

1.4. Security objectives

Statistics have shown the importance of the human factor in the form of negligence or ignorance for the generation and reinforcement of security weaknesses; this is the same as in the case of car accidents. Because of this, a large part of the investment in questions of security is aimed at the user to teach and sensibilize them. Additionally, it is necessary to put technical security solutions in place.

The objective of any security measure consists of mitigating the risks created by the security weaknesses, in order to minimize the attacks that exploit them, given that hackers always discover security weaknesses within networks, systems and applications.

There is no longer a radical, blanket security solution; there is only *mitigation*.

1.4.1. *Establishing a culture*

The human factor must always constitute the central axis of any investment. For security, it is clearly the most important element, given the weaknesses it can cause.

Establishing a security culture consists of making users more sensitive and aware of their responsibilities through several mechanisms:

– hold training sessions for all users of computing tools, without betting on their ignorance as a security measure;

– formulate, require signatures and follow a contract regarding use of material resources and software, as well as data and documents;

– establish a team in the company to manage security problems and sensibilize personnel using posters, clauses that should be signed, identity control measures, etc.

1.4.2. *Establishing technical solutions*

This is the technical part of security. It consists of finding solutions and establishing the appropriate security equipment and software. Its purpose is to prevent:

– the unauthorized spread of information;

– the unauthorized modification of information;

– the unauthorized use of resources.

These security measures are generally the result of an investigation led by security experts within the company. It is the job of a security audit or a consultant, as required by law.

1.5. Security fields

Computer security covers numerous complimentary fields that include the following aspects: energy, physics, organization, environment, etc.:

– the evaluation of risks;

– security rules;

– organization of security information;

– asset management (software and data);

– human resources security;

– physical and environmental security;

– management of communications and operations;

– acquisition, development and maintenance of information systems;

– access checks;

– managements of incidents concerning information security;

– conformity.

Security in the most general sense can concern numerous elements as varied as any, but we can distinguish three principal areas of security, namely, energetic, physical and software.

1.5.1. *Energy security*

In the case of a complex computer system with real-time services, it is necessary to have servers on standby and automatic back-ups. Because of this, a simple electric outage can cause considerable damage. In this

scenario, it would be necessary to launch groups of generators or inverters for the purpose of having an energy source that lasts and is independent of the sector that can be subject to periodic blackouts. We speak of energy security.

1.5.2. *Organizational and physical security*

The computer system, as reduced as it may be, risks being physically affected by human and/or natural means; we can cite, as examples, fires, floods, theft, etc.

Physical security consists of taking measures to control and physically isolate the system. It exists in two forms: high-level security concerns the organizational part and manages the different physical security strategies. Infrastructure security tries to physically secure computer equipment.

1.5.2.1. *High-level security*

High-level security covers numerous security elements:

– organization;

– security structure;

– personnel;

– management of saves;

– incident management.

The purpose of organizational security is to try to validate structures and organizations responsible for security, including standby technological units, administrators and security agents. It also allows for the verification of investments in security within companies, including training sessions and steps towards creating the necessary security culture using charters, posters and management rules. Other points fall into this aspect of security, such as the rules on backup and the management of incidents that occur within the company.

1.5.2.2. *Infrastructure security*

Infrastructure security covers numerous elements of security:

– cables;

– offices;

– server location;

– archive storage;

– technical offices;

– server cabinets;

– calculation center;

– home working sites.

This aspect concerns the physical part. It tries to physically secure computer equipment and infrastructure, as well as their locations. With regard to the locations, the security policies depend on the importance and content of the different locations. Servers benefit from a specific security policy, given their importance for the entire computer system. The home working sites represent a weakness and require specific security measures that are not to be ignored.

Physical security protects against natural and human factors and can be ensured through:

– smoke detectors;

– water detectors;

– movement detectors;

– surveillance cameras;

– clocking-in systems (card, fingerprint, etc.).

1.5.3. *Software security*

Software security concerns software and system assets that must be secured using passwords, limitations on access, electronic classification of documents depending on their importance and degree of confidentiality. This is the fundamental aspect of computer security and what it comes down to is defined by the OSI 2 (ISO 7498-2) norm.

Security concerns any computer system (network, workstation, server, etc.), as well as operating systems and applications.

1.5.3.1. *Network security*

Networks are the key element in the field of computing, and because of this, they are a potential target for attackers. This implies the need for a security investment allowing measures such as filtering, access checks and infrastructure protection to be taken.

Different network devices are involved in security, as is the case with:

– modems;

– switches (VLAN, port security);

– firewalls;

– routers (ACL);

– access points.

Network security allows remote access to be limited to different segments and network devices.

1.5.3.2. *Systems security*

Operating systems are the basic element of the software part of a computing system. Each operating system, however recent, contains vulnerabilities that can be used by attackers. Microsoft Windows systems in particular are a common target, given their widespread use.

This aspect of security also concerns telephonic and fax standards because they can be used as an unauthorized pathway for disrupting security.

1.5.3.3. *Application security*

Different software and applications are not above all criticism and cause numerous weaknesses that attackers are always able to identify, despite the patches that developers regularly provide to add the necessary correctives. Additionally, the attacks waged are specific to the software in question. Because of this, specific security precautions and preventions are necessary for different software. The classes of software that represent potential targets are:

– web services;

– email services;

– data services.

1.5.3.4. *Data security*

Even though software is of capital importance, the loss or alteration of data can have damaging effects on a computing system. It is therefore necessary to foresee the requisite security measures, typically a series of varied and complementary mechanisms, in order to protect data against unauthorized leaks or modifications.

Statistics have shown that a large part of the damage caused by attacks is due to the loss of data. Data security includes numerous measures:

– integrity control;

– encryption;

– data server redundancy;

– database security;

– data availability.

1.6. Normalization of security

Security has become an important challenge, and in order to avoid any confusion or divergence, normalization is the solution; this requires defining the necessary terminology, as well as the different requirements and solutions.

1.6.1. *Fundamental issues and general presentation*

In order to fix the issue of divergence and provide security with a solid and well-structured foundation, a standard has recently been created. It is the ISO 7498-2:1989 norm, which is part 2 of the OSI model for security architecture[3]. The international norm was created by the ISO/TC 97 technical committee.

The International Telecommunication Union (ITU) created the X.800 norm, providing a standard of security architecture for open systems.

3 http://www.iso.org.

1.6.2. *ISO 7498-2 norm*

In order to address secure communications between open systems, the second part of the ISO 7498-2 was published in 1989. It:

– gives a general description of security services and the associated mechanisms that can be provided by the reference model;

– defines where services and mechanisms can be provided in the OSI architecture.

Basic security services and mechanisms, and their appropriate locations are identified for every layer of the basic reference model. Moreover, architectural relationships between security services and mechanisms in the basic reference model are identified. Supplementary security measures can be necessary at endpoints, in installations and organizations. These measures are applicable in different contexts. Defining which security services are needed to take charge of these supplementary security measures lies outside of the field of application of this international norm.

The OSI security functions only concern the visible aspects of a communication route that allows endpoints to complete secure transfers of information amongst themselves. OSI security does not deal with the necessary security measures within the endpoint systems, installations and organizations, except when these measures have an effect on the choice and placement of security systems that are visible in the OSI. These aspects of security can be normalized, but not within the framework of OSI norms.

The current part of the ISO 7498 completes the concepts and principles defined in the ISO 7498-1, without modifying them. ISO 7498 provides the definitions of the following terms, which are applicable to the field of security:

– *Vulnerability*: a weakness or deficiency of the system or network that can be exploited or not.

– *Attack*: a malicious action allowing for the exploitation of a weakness.

– *Access control*: a precaution taken against the unauthorized use of a resource; this includes the precautions taken against the use of a resource in an unauthorized way.

– *Access control list*: a list of entities authorized to access a resource; this list includes access rights tied to the entities.

– *Accountability*: a property guaranteeing that the actions of an entity cannot be attributed to anyone except that entity.

– *Active threat*: the unauthorized and deliberate threat of modification to the state of the system.

– *Authentication information*: the information used to establish the validity of a declared identity.

– *Authentication exchange*: a mechanism whose purpose is to guarantee the identity of a user by information exchange.

– *Authorization*: the attribution of rights, including the permission to access the access rights data.

– *Availability*: a property of being available and usable on demand by an authorized entity.

– *Capacity*: a token used as an identifier for a resource, such that the possession of the token gives access rights to this resource.

– *Path*: a pathway used to transfer information.

– *Cryptogram (ciphered text)*: data obtained through the use of encryption. The semantic content of resulting data is incomprehensible.

A cryptogram can be reinserted into a new encryption to produce a *superencryption*.

– *Plain text*: intelligible data whose semantics are understandable.

– *Confidentiality*: a property of information that is neither available nor shared with unauthorized people, users or processes.

– *Identity document*: transferred data used to establish the declared identity of a user.

– *Cryptographic analysis*: the analysis of a cryptographic system and/or its comings and goings, in order to deduce the confidential variables and/or sensitive information (including the *plain text*).

– *Value of the cryptographic check*: the information obtained via a cryptographic transformation (see *cryptography*) on a unit of data.

The control value can be obtained in one or more steps and comes from a mathematical function that uses the key and a unit of data. It allows us to verify the integrity of the unit of data.

– *Cryptography*: the discipline, including the principles, means and methods for transforming information, with the objective of hiding its content, preventing its modification from remaining undetected and/or preventing its unauthorized use.

Cryptography determines encryption and decryption methods. An attack aimed at the principles, means and methods of cryptography is called a cryptographic analysis.

– *Data integrity*: a property guaranteeing that data has not been modified or destroyed in an unauthorized fashion.

– *Data source authentication*: confirmation that the source of received data is indeed the one that has been declared.

– *Decryption*: the inverse operation of a reversible encryption.

– *Denial of service*: the inability to access resources for authorized users or the introduction of a delay for the processing of critical operations.

– *Electronic signature*: data added to a unit of information, or the cryptographic transformation of a unit of data, allowing the recipient to prove the source and integrity of the unit of data, protecting against counterfeits (by the recipient, for example).

– *Encryption*: the cryptographic transformation (see *cryptography*) of data, producing a *cryptogram*.

Encryption can be irreversible. In this case, the corresponding decryption cannot be completed; this is the case for a hash function.

– *End-to-end encryption*: *encryption* of data within, or at the level of the endpoint source; the corresponding *decryption* can only be made within, or at the level of the destination endpoint.

– *Identity-based security policy*: security rules based on the identities and/or the attributes of users, a group of users or entities acting in the name of users; these can also be based on the identities/and or attributes of resources/objects to which they seek access.

– *Key*: a series of symbols commanding *encryption* and *decryption* operations.

– *Key management*: the production, storage, distribution, elimination, archiving and application of keys following the *security policy*.

– *Link encryption (link by link)*: a particular application of *encryption* to each link in a system.

Link encryption implies that information is presented as plain text, as it passes through the relay entities.

– *Modification detection*: a mechanism used to detect modifications, whether accidental or intentional, of a unit of information.

– *Dissembling*: when one entity pretends to be another.

– *Notarization*: saving data with a secure third party, allowing the verification of its exactitude (content, origin, date, delivery) at a later date.

– *Passive threat*: the threat of unauthorized sharing of information without the state of the system being modified.

– *Password*: confidential authentication information, usually composed of a chain of characters.

– *Homologous entity authentication*: confirmation that a homologous entity of an association is indeed the declared entity.

– *Physical security*: the measures taken to ensure the protection of resources from deliberate or accidental threats.

– *Privacy*: the right of individuals to control or act on information concerning them, which can be collected or stored, as well as their rights regarding the people through whom and to whom this information can be shared.

Given that this term is tied to privacy law, it cannot be very precise, and its use should be avoided except for security needs.

– *Repudiation*: the act of denying participation in exchanges, either entirely or in part, by one of the entities involved in a communication.

– *Router control*: the application of rules during a routing process in order to choose or avoid certain networks, connections or specific relays.

– *Rule-based security policy*: security rules based on the general rules imposed upon all users. These rules are generally based on a comparison of the sensitivity of resources that need to be accessed with the attributes of users, a group of users or entities acting in the name of users.

– *Security audit*: independent review and examination of activity and information saved by the system, in order to verify the exactitude of system controls, to ensure their compliance with the established security policy and usage procedures, detect security breaches and recommend the appropriate modifications to controls, policy and procedures.

– *Security audit record*: collected data that could be used to facilitate a security audit.

– *Security certificate*: a badge tied to a resource that names or designates the security attributes of this resource (the resource can be a unit of information).

The badge and/or the association of the badge with the resource can be implicit or explicit.

– *Security policy*: the group of criteria used to provide security services.

A complete security policy necessarily covers subjects that are outside of the scope of the OSI.

– *Security service*: a service provided by a group of open systems that guarantees the security of systems and information transmission.

– *Selective protection of fields*: the protection of certain specific fields in a sent message.

– *Sensitivity*: the characteristic of a resource reflecting its value or importance and, in some cases, its vulnerability.

– *Threat*: a potential violation of security.

– *Traffic analysis*: the deduction of information by observing the flux of data (presence, absence, quantity, direction, frequency).

– *Data flow confidentiality*: confidentiality service providing protection from traffic analysis.

– *Padding*: the production of parasitic instances of communication, parasitic units of information and/or parasitic data in the units of data.

– *Confidence functionality*: the desired functionality with regard to certain criteria, such as those defined by a security policy.

1.7. Security services

By *security service*, we mean a security need that must be satisfied. A computer system is qualified as secure if it satisfies the principal conditions known as security services:

– authentication:

- origin of data;

- between entities;

– confidentiality:

- of data;

- of traffic;

– integrity:

- of data;

- of traffic;

– non-repudiation:

- of the origin;

- of the recipient;

– access control;

– service availability;

– traceability.

Authentication and integrity can be grouped together in the term authenticity.

These services apply to system entities and/or information that is stored locally, in the middle of being processed or gets transmitted through a network.

Ensuring computer security within a company necessarily consists of providing the different needs mentioned above. Certain services appear more critical than others, depending on the nature of traffic and the information being manipulated. The system and network administrator, and more generally, the security manager, must define priorities in satisfying different security services. A priority can change depending on the circumstances.

Any violation or threat to one or numerous security services necessarily brings a threat to the security of the system or company in question.

1.7.1. *Authentication*

Contrary to identification, which serves to mark or otherwise distinguish an entity among others, authentication is a proof of identity; it serves to ensure that the entity in question is really the entity that just identified itself.

The act of saying that "I am X" is an identification; this must be followed by a means for proving that we are really "X", which is authentication.

There are numerous types of basic authentication, namely: I know, I have and I am. They can be set up individually or coupled.

1.7.1.1. *I know*

This is the simplest and most well-known method. It consists of authenticating ourselves via a secret combination of characters; this is the password associated with a login. The login is unique and serves as identification. The password is secret and serves as authentication.

A password must satisfy a group of rules in order to avoid any kind of sharing or brute force, and withstand attempts at intrusion. As such, a password must satisfy the following rules:

– non-trivial (abc, qwerty, 123, etc.);

– not so complicated that it is difficult to learn;

– should never appear anywhere physically (piece of paper, notebook, etc.) or digitally (file, email, etc.);

– does not come from a dictionary;

– does not correspond to a personal characteristic (last name, first name, license plate number, etc.);

– is not used twice (Facebook account, email account, etc.);

– a combination of capital letters, lowercase letters, numbers and special characters;

– periodically modified in order to limit how long it is used for.

A good practice for choosing a password is to choose a word from the dictionary or that comes from another language (such as Arabic) and formulate it using Latin letters, or perhaps even numbers (e.g. 3a55alem), making sure to use obvious spelling errors (e.g. conbuter for computer) or inverted syllables (e.g. terpucom for computer).

1.7.1.2. *I have*

Authentication in this scenario is guaranteed by the possession of a private physical object. Microchipped cards can be used as a means of checking personnel.

This method is the most primitive and easiest to use; it presents some blind spots tied to the possibility of being able to give the object in question to someone else.

This is the case for authentication that allows access to a private locale using a visitor's badge for people from outside a company.

For bank cards, we have two means of authentication in sequence: I have (the card) and I know (the code).

1.7.1.3. *I am*

This is the strongest means of authentication. It is tied to some of the personal characteristics of a person, such as a digital imprint (we speak of biometric authentication) or characteristics of the eyes or a hand (form, heat, etc.).

Such a method can be used to control access to places or critical locales. It is also useful for authentication on a computer or system.

1.7.2. *Confidentiality*

This is the act of ensuring that manipulated information does not get divulged by any unauthorized third-party entity. It is a protection against the spread of critical information, using the technique of encryption, in order to make the information illegible and incomprehensible by other, non-relevant entities.

1.7.3. *Integrity*

This is a protection against the modification of manipulated information. It is a means of detecting whether a block of information has undergone a modification, however simple, using hash techniques, which consist of generating a representative message that is unique and dependent on the information in question.

1.7.4. *Non-repudiation*

This is the act of guaranteeing that none of the correspondents (sender or recipient) in the transmission of information will be able to deny the transaction.

1.7.5. *Traceability and access control*

Traceability consists of being able to follow access to sensitive computer resources (time of connection, list of actions, etc.). Access control consists of limiting access to different shared resources, both material and software.

1.7.6. *Service availability*

This is the act of maintaining the proper functioning of a service, in order to satisfy the requests of authorized users in all scenarios. It is a protection of the server itself via measures and techniques that detect and neutralize attempts by external entities to interrupt service.

1.8. Security mechanisms

By *security mechanisms*, we mean a well-defined measure or technique allowing one or more security services to be provided.

Computer security mechanisms are the means allowing for the provision of security services. The ISO 7498-2 norm allows us to specify:

– encryption using shared a key;

– encryption using a private/public keys;

– integrity of data with a hash function;

– authentication;

– access control;

– digital signature;

– notarization.

1.8.1. *Encryption*

Encryption is a technique that has been used for centuries. It entails the modification of a text to make it incomprehensible, except for the person who generated it. The branch of mathematics that deals with this subject is called cryptography. This mechanism allows confidentiality services to be provided.

Encryption is based on two basic techniques: substitution (a table of correspondences between letters and numbers) and shift, also known as the Caesar cipher, which allows letters to be substituted in a circular manner.

Since its appearance, this mechanism has gone through several evolutions: to begin with, the first solutions were based on the choice of an algorithm that would be used for encryption, as well as for decryption. Then, a new generation of encryption solutions used a symmetrical key (the shift value in the case of the Caesar cipher). This key would be used for both encryption and decryption. Finally, encryption with asymmetrical keys was created. This entails a pair of keys used for encryption and decryption, with

the purpose of avoiding the problem of sharing, which is the security weakness in the case of the first two generations of encryption.

We can cite the following encryption algorithms as examples: DES, 3DES, AES, RSA, Blowfish, etc.

Encryption was never an absolutely necessary solution insofar as any cryptogram can be analyzed to obtain the corresponding plain text. We speak of cryptoanalysis, which is a technique based on linguistic statistics. As an example, we can think of Turing's work during World War II, which allowed the Allies to find the plain texts of military messages sent by the enemy's governors to their troops, thanks to Turing's cryptoanalysis machine.

1.8.2. *Integrity check*

An integrity check is a mechanism based on the hash algorithm that allows integrity service to be provided.

A hash algorithm is a non-bijective function that generates a chain with a unique, fixed size, in other words a hash code made from a chain of variable length. The latter can be a packet, message, web page or other; it is the thing that proves its integrity.

The hash code represents a signature insofar, as any change in the original message provides a different hash code, which allows for the detection of any change to the secure entity, and consequently confirms its integrity. Of the many hash algorithms that are available, the most important are MD5 (16 bytes) and SHA1 (20 bytes).

1.8.3. *Access check*

An access check is the mechanism allowing for the provision of an authentication service. Authentication techniques can be classified into three large families tied to possession, knowledge of an entity or personal characteristics (biometric, speech, etc.).

Authentication is necessary to ensure local or distant access to any service; it is the best way to limit access and identify responsibilities.

The most commonly used authentication technique is the one based on passwords. Such a technique requires that conditions be put on the choice, lifetime and usage of a password, which is the key element and a critical element of any computer system.

1.8.4. *Electronic signature*

A digital signature (sometimes called an electronic signature) is a mechanism that guarantees the integrity of an electronic document and authenticates its author, by analogy with the handwritten signature on a paper document. A mechanism for digital signatures must:

– allow the reader of a document to identify the person or company that made the signature;

– guarantee that the document has not been altered between the moment when the author signed it and the moment when the reader looked at it.

1.8.5. *Notarization*

Electronic notarization is the certification of different steps in the evolution of an electronic document, with the aim to:

– guarantee the contents, origin, date and destination of an electronic message between two parties in an exchange;

– securely archive digital documents.

Electronic notarization allows evidence of electronic exchanges to be verified and stored, and for a trusted, certified third party to store them electronically (as with a notary). This technique improves the security of exchanges and electronic storage due to the fact that it offers different mechanisms for monitoring and archiving transactions, whether sent or received (integrity, origin, date and destination of data).

1.9. Good practices

Computer security can be ensured through good practices that attenuate security risks and make attacks ever more difficult to carry out. Good practices in terms of habits and technology include:

– coming up with a written security policy;

– training employees with regard to social engineering risks and developing strategies to validate their identities over the phone, email, or even in person;

– physical access checks for systems;

– using strong passwords and changing them regularly;

– encrypting sensitive information and protecting it with passwords;

– limiting connection privileges to the operating system;

– developing security devices and software;

– saving documents and testing saved documents regularly;

– deactivating unused services and ports;

– keeping patches updated by installing them every week or every day, in order to avoid buffer overflow and attacks due to privilege escalation;

– periodically planning security audits to test the network and systems, and making sure to apply the recommendations that come from these.

1.10. Conclusion

Computer security is currently considered as the solution to the problems discussed here, as illustrated by statistics. Weaknesses and vulnerabilities are numerous and varied, and they cover every level of computing, from devices to software, from infrastructures and structures to users and administrators.

Computer security allows us to cover numerous fields and plan organizational, cultural and technical solutions, in order to give the elements of solutions for the varied weaknesses and vulnerabilities.

Security, whose objective is to ensure a computer system remains in proper working order, touches upon several external and internal points, physically and in software, material and human, etc.

The technical aspect of computer security in the strict sense comes down to the magic term ACID, a French acronym for the four points to satisfy if you want to guarantee proper functioning: authentication, confidentiality, integrity, and *Disponibilité* or availability. But in order to verify these points, we must be familiar with the weakness and vulnerabilities.

2

Security Weaknesses

2.1. Introduction

TCP/IP is the default norm that appeared as a normalization of something that already existed, ARPA-Net, without any modification or consideration of numerous points, and in this case, security measures.

Moreover, given both its simplicity and the exponential development of the Internet, TCP/IP was most frequently used and established itself as the norm. Indeed, it is rare or even impossible for a computer engineer to find a norm other than TCP/IP.

The history of TCP/IP is that of the Internet or even of its ancestor, ARPANet. The Internet, however, due to its history and the circumstances of its appearance, became the distributed solution. We speak of "intelligence at the limits", and this is the principal reason behind all of the weaknesses discussed, given that advanced and complex functions are made available to the final user.

The use of TCP/IP locally (Intranet) presents the same security weaknesses, or even more serious ones, given the restricted size and ease of information sharing.

The considerable progress made in the field of computer engineering facilitated the appearance of malicious programs, or malware, and intrusion tools that allow information to be destroyed, configurations to be modified, the system to be put out of order, etc., whether directly or indirectly, by facilitating access to a computer or even by diminishing its resources.

Computer Science Security: Concepts and Tools,
First Edition. Ameur Salem Zaidoun.

The physical and non-physical destruction caused by malware is always increasing, presenting repercussions on computing systems and requiring ever greater efforts and time for the restoration of proper functioning to systems after an attack.

2.2. Weakness in the TCP/IP

TCP/IP, used widely, is a family of protocols that, given its history and through its very structure, is very vulnerable to attacks.

2.2.1. *ARPANet, the ancestor of the Internet*

ARPANet is the acronym for Advanced Research Projects Agency Network, a private network used by the American army.

ARPANet was developed in the context of the Cold War as a means of private communication for the American DoD. At the time, the objective was to have a network that could work even if it was partially damaged by a possible attack. It was necessary to eliminate the centralization of communication, as was the case in other military strategies.

ARPANet is, on the one hand, a private communication network that was entirely under the control of the American army, and, on the other hand, a distributed solution. The distribution was of both the functionality and the tools. There was no central controlling it; the only guarantee at the time was its private nature, which is no longer the case today due to the Internet, which has become an incontrollable public communication network, the access to which cannot be managed nor limited by anyone.

2.2.2. *The Internet and security problems*

The problem of security appeared with distribution, but it was neglected because all of the nodes in the ARPANet were under control.

In 1974, the TCP/IP was created to make the ARPANet uniform; the system is still in use today.

In 1980, ARPANet was divided into two distinct networks, one was military (MILNET: Military Network, which would become the DDN:

Defense Data Network) and the other was for universities (NFSNet: National Science Foundation Network), which the military left to the civilian sphere. The idea of the builders was to have decentralized computing.

On January 1, 1983, ARPANet adopted TCP/IP, which would become the basis of the Internet, which underwent considerable evolutions before becoming integral to the industry; it brought with it the problem of security, which has become increasingly relevant.

Academia was the seed in which the Internet developed. The problem of security began to appear there but given that the information being shared was not critical, it was not considered a subject of debate and research.

The Internet developed considerably, which pushed the industry to adopt it widely as a means of communication. Given that people send critical information and that Internet users are ever more numerous, diverse and anonymous, the security problem has now become serious. At this stage, it has become impossible to backtrack; the Internet is an unavoidable necessity, and we must find security solutions.

2.2.3. *The Internet and the ability to analyze*

Among the different security services, confidentiality stands out. It can be called into question by simple spyware targeting information as well as users. This act can be followed by an intervention that modifies the information or even the configuration. Spying and intervention can be accomplished with the right software.

2.2.3.1. *Scanning tools*

Scanning tools are surveillance software covering the system and the network. They are tools that enable the identification of entities within a computing system or a network such as physical and electronic addresses, names, open ports, operating systems including patches and weaknesses, applications, registries, protocols used, interconnected equipment, and different configurations.

There is a large panoply of scanning tools available on the Internet that can be easily downloaded and exploited by attackers and spies, which makes the act of scanning banal and easily within reach. This puts the security of

systems and computer networks at risk, making access easier, which in turn allows the limits and weaknesses of systems and networks to be identified. This then puts hackers in a position to program and launch targeted attacks that exploit the observed vulnerabilities.

A computer system exposed to scans will present a great weakness since it will be surveilled by attackers and network observers who can decide to act and call the configuration of the system and/or network into question.

2.2.3.2. *Analysis tools (sniffers)*

These are tools that allow us to analyze and/or modify information about a computer system. The spied information can either reside on the computers or circulate through the network. There are two families of scanners: passive and active:

– A *passive scanner* (for reconnaissance) limits itself to spying on information without stepping in to modify it. This is a passive attack.

– An *active scanner* (for access) updates information, which allows errors to be introduced, the configuration to be changed, information to be lost, etc.

As mentioned, on the Internet, there is a vast panoply of scanning software, both passive and active. These tools are easily accessed by attackers to observe traffic and saved information at network nodes, compromising it at times through unauthorized updates. Some of the sniffers available on the Internet are Wireshark, MSN Sniffer, ICQ Monitor Sniffer, Link Sniffer, EtherDetect, Jitbit Network, and EffeTech http Sniffer.

2.3. Weaknesses due to malware and intrusion tools

Malware, or malicious software, sometimes called harmful software, is a program developed with the purpose of causing harm to a computer system without the consent of the user whose computer is infected.

Currently, there is a wide array of this kind of software, of which a large part can be found on the Internet. Malware can be a virus, a worm, a Trojan horse, as well as other threats. We can distinguish spyware from key loggers and rootkits, etc.

2.3.1. *Viruses*

These are the most well-known tools for direct intrusion. The real name given to this type of program is the self-replicating code, but by analogy with the field of medicine, the word “virus” has been given to it.

The production of viruses has gradually increased over time. The first virus appeared in Pakistan and was created by the Amjed brothers in the 1980s, approximately.

A virus can execute various actions, ranging from an undesirable message to reformatting the disk. It can update the configuration of systems or networks and even compromise systems, communication equipment and devices.

2.3.1.1. *Definition*

A virus is a small program that self-replicates (copies itself) in order to infect other programs and cause the destruction of information.

2.3.1.2. *Characteristics*

A virus has two intrinsic characteristics:

– self-replication;

– the destruction of information.

The first characteristic reflects the capacity of a virus to copy itself from one place to another, to change repertoire, disk, computer, etc., without any intervention or external assistance.

The second characteristic reflects the malicious capacity of this piece of code that causes data, programs, configurations, etc. to be compromised.

A virus can gain other characteristics if it allows for a precisely defined behavior that acquires supplementary capacities, making it able to hide and resist system protections. We can distinguish two principal characteristics: mutation and polymorphism.

Mutation

A virus is called a mutant if it creates numerous versions of itself (we speak of variants) that differ in their behavior, namely the messages shown

the execution traces. Such a virus is difficult to verify since it changes in appearance without changing either form or action, which allows it to maintain its signature.

Polymorphism

A virus is considered polymorphous if it exists in numerous forms and presents itself with numerous file formats (.exe, .bat, .sys). It is like a chameleon: it takes a form that blends with the repertoire where it exists. Such a characteristic allows the virus to change its signature, which makes the task of detection by appropriate systems increasingly difficult.

2.3.1.3. *Different forms*

There are seven principal families of viruses:

– *Joke*: this kind of virus makes nonsensical messages appear with the sole purpose of annoying the user.

– *False alarm*: in this scenario, the virus displays alarm messages to announce non-existent security problems or virtual system problems.

– *Test*: this type of virus allows certain critical information to be spied on and discovered by asking users questions.

– *Macro*: in this scenario, we speak of macro-viruses, that is, viruses based on macros. It is a type of virus that is particular to Microsoft (Office) applications, occurring when a macro that destroys information and reproduces itself is created.

– *Batch*: these are viruses in the form of batch files (.bat).

– *Executable*: these are viruses in the form of executable files or dynamic libraries (.exe or .dll).

– *System*: these are viruses in the form of system files (.sys or .vxd).

2.3.1.4. *Examples*[1]

2.3.1.4.1. Klez

The Klez virus, which appeared at the start of 2002, refers to new variants of the virus that keep appearing (Klez.e, Klez.g, Klez.h, Klez.i, Klez.k, etc.). It uses four modes of propagation:

1 http://virus.wikidot.com.

– the Web;

– shared folders;

– the weaknesses of Microsoft IIS servers;

– the exchange of files.

In particular, it affects users of Microsoft Outlook on the operating systems Windows 95, 98, Millennium, NT4, 2000 and XP, as well as users of Microsoft Internet Explorer.

The Klez virus obtains the list of addresses stored as contacts in Microsoft Outlook, Eudora and instant messaging software (ICQ), and then sends an email to all the recipients using its own SMTP server.

The Klez virus is thus able to generate emails with an empty body and whose subject is chosen randomly among a selection of a hundred or so predefined themes, and adds an executable attachment to this email containing a variant of the virus.

2.3.1.4.2. Magistr

The Magistr virus is a polymorphous worm (i.e. a virus that propagates itself through the network and whose form, or more exactly its signature, modifies itself continuously) that propagates itself using email. It especially affects users of the messaging clients Microsoft Outlook, Eudora and Netscape under the operating systems Windows 95, 98, Millennium and 2000.

The Magistr virus seeks contact list files that are stored in the system (with .WAB and .DBX/.MBX extensions for clients of Outlook and Eudora, respectively) in order to select the recipients of the message.

The subject and body of the message sent by the Magistr worm are chosen randomly by taking an excerpt from a file found on the infected computer's drive. The Magistr virus adds a copy of itself to the message with a name containing an extension (or a double extension) like .com, .bat, .pif, .exe or .vbs. It can also potentially erase all of the information contained in the CMOS, the BIOS or in the hard disk.

It can thus cause serious damage to the system and the information stored on it. Moreover, it is capable of deactivating the personal firewall ZoneAlarm using the WM_QUIT command.

2.3.2. *Worms*

A worm is an intrusion tool that propagates itself through the network. It is a network virus whose primary characteristic is tied to the way in which it propagates through the network.

2.3.2.1. *Effects*

Currently, worms mostly replicate themselves using emails (and notably with the messaging client Outlook) thanks to the attached files containing instructions that train all of the email addresses contained in the contact list to be collected so that copies can be sent to all of these recipients.

Most of the time, these worms are scripts (generally VBScript) or executable files sent as an attachment that initiates once the recipient clicks on the attached file.

2.3.2.2. *Prevention methods*

In this way, all files that are executable or interpretable by the operating system can potentially infect a computer. It is simple to protect ourselves from an infection by a worm by being distrustful of files sent as attachments.

Files containing the following extensions in particular are potential sources of infection:

386, ACE, ACM, ACV, ARC, ARJ, ASD, ASP, AVB, AX, BAT, BIN, BOO, BTM, CAB, CDR, CHM, CLA, CLASS, CMD, CNV, COM, CPL, CPT, CSC, CSS, DLL, DOC, DOT DRV, DVB, DWG, EML, EXE, FON, GMS, GVB, HLP, HTA, HTM, HTML, HTT, INF, INI, JS, JSE, LNK, MDB, MHT, MHTM, MHTML, MPD, MPP, MPT, MSG, MSI, MSO, NWS, OBD, OBJ, OBT, OBZ, OCX, OFT, OV?, PCI, PIF, PL, POT, PPT, PRC, PWZ, QPW, RAR, SBF, SCR, SH, SHB, SHS, SHTML, SHW, SMM, SYS, TAR.GZ, TD0, TGZ, TLB, TSK, TSP, TT6, VBE, VBS, VBX, VOM, VS?, VWP, VXE, VXD, WBK, WBT, WIZ, WK?, WPC, WPD, WML, WSC, WSH, XML, XLS, XLT, ZIP.

Under Windows, it is recommended that the “mask extensions” function be deactivated because this function can mislead a user regarding the real extension of a file. So, a file with the extension .jpg.vbs could appear as a file with a .jpg extension.

2.3.3. *Spam*

This is an intrusion tool tied to an email, presenting itself in the form of an undesirable message.

Spamming, which constitutes approximately 90% of email traffic, consists of sending out emails in massive amounts, generally for advertising (also called "junk mail") to a large number of people who have not asked for this kind of advertising to be sent, thus filling message servers and inboxes with useless, unsolicited and usually misleading advertisements. Spammed emails currently constitute almost half of all emails circulating throughout the globe.

Recent surveys taken since 2003 show that the propagation of spam represents the greater part of the total number of emails worldwide.

2.3.3.1. *Effects*

Spam brings about numerous effects:

– a waste of time (and thus money) for users, who must sort their email and clean out their inboxes more frequently;

– the risk of overlooking an important message, "hidden" among multiple spam messages;

– the "corruption" of unaware users with attractive but naturally fake offers;

– moral offenses, via messages with sexual, political or religious advertisements;

– a waste of the network bandwidth, which it uselessly consumes, monopolizing a large part of the bandwidth.

2.3.3.2. *Prevention methods*

The proper measures that we should take to counteract spam are:

– Do not respond to spam messages (do not threaten them, etc.), for that would only make matters worse since it would confirm to spammers that you are indeed receiving their emails, and also that you are receptive to this type of message.

– Configure your email client, given that messaging software and webmail services sometimes allow us to block certain undesirable senders. Moreover, certain messaging clients systematically delete the emails in question.

– Do not give your email address on dubious web forms because many of these websites provide these addresses to spammers. It would even be useful to have two or more email addresses, one of which is reserved for online identification and discussion groups.

2.3.4. *Software bomb*

This is a case of hidden intrusion that starts suddenly.

It is a program that begins after a system event (startup, starting a particular program, etc.) or even according to a preestablished schedule. A good illustration of this is the Chernobyl attack tool, which starts up on the same date of the Chernobyl nuclear catastrophe.

2.3.5. *Trojan horse*

This is an indirect intrusion too. It is more complex and more serious than a virus. Its use is tied to the network.

2.3.5.1. *Definition*

This is a program that hides inside another program and opens ports. It allows someone to take control of a computer, execute programs and start commands without the user knowing.

A Trojan horse in and of itself does not do anything malicious, but it allows other entities to share information and/or modify it by opening ports through the network.

2.3.5.2. *Symptoms*

An infection by a Trojan horse causes numerous symptoms, of which we can mention the following:

– uncontrollable movement of the mouse;

– opening and closing of windows;

– abnormal activity in the modem or any other connection equipment, reflecting abnormal traffic on the network;

– loss of control of the computer.

2.3.5.3. *Examples*

Table 2.1 presents known examples of Trojan horses with the relevant ports that they manage to open without the knowledge of the user, creating hidden communication pathways.

Trojan	Port	Trojan	Port
TransScout	2002	SpySender	1807
TransScout	2003	Shockrave	1981
TransScout	2004	BackDoor	1999
TransScout	2005	TransScout	1999
Ripper	2023	TransScout	2000
Bugs	2115	TransScout	2001
HVL Rat5	2283	Trojan Cow	2001
Striker	2565		

Table 2.1. *Examples of Trojans and relevant ports*

2.3.6. *Spyware*

Software that spies, or spyware, is malicious software that installs itself in a computer or mobile device with the purpose of collecting and transferring information about the environment in which it is installed, very often without the user knowing. The rise of this type of software is strongly tied to that of the Internet, which serves as its means for transmitting information.

Spyware is software intended to discover the navigating habits of a user. Installed at the same time as smaller, free software downloaded off the Internet or received via email, spyware generally starts up with the computer and executes background tasks, appropriating a part of the system's resources. Its essential function is to collect as much information as possible about the navigating habits (or even purchases) of the computer's user in order to make a detailed user profile.

Certain spyware uses cookies to collect data relative to certain users in the case where browsers allow such options to be activated.

Profiles created in this way, without the knowledge of the user and thus without advanced permission, are then used to send targeted advertising (e.g. modifications to pages shown, the addition of links), or are even sold in the form of files containing qualified prospects for email campaigns.

EXAMPLES.– Adayairespy, AdwarePunisher, AdwareSheriff, AlphaCleaner, AVGold, BargainBuddy, BraveSentry, MalwareWipe, PestTrap, PSGuard, Quicknavigate.com, Security iGuard, Smitfraud, SpyAxe, SpyGuard, SpyHeal, SpySheriff, Spyware Soft Stop, Spyware Vanisher, SpywareQuake, SpywareSheriff, Startsearches.net, UpdateSearches.com, Virtual Maid, Win32.puper, WinHound

2.3.7. *Keylogger*

A keylogger is spyware or a peripheral that electronically spies on the user of a computer. The purpose of this tool is to spy on a user's private computer use.

A keylogger is a configuration in charge of saving the typed keys of a keyboard without the knowledge of the user.

Certain keyloggers are capable of recording the URLs of pages visited, emails that have been viewed or sent, files that have been opened and at times can even create a video that retraces all computer activity.

Insofar as keyloggers save all the keystrokes of a keyboard, they can be useful to people with illicit intentions, seeking the passwords of users at a workstation. This means that particular attention is therefore needed when using a computer that is not secure (an open access workstation at a company, a school or a public space).

2.3.8. *Adware*

Adware is a type of malicious software that drowns a user in endless advertising windows that are potentially dangerous for the device. It allows its editor to generate advertising revenue and is generally installed along

with different kinds of open-access software available for free download, and without the knowledge of the user.

EXAMPLES.– 1ClickDownloader, 4-you.net Search, 7search, AnywhereMe Toolbar, Arcade Safari, Auto-Lyrics, AutoCompletePro Toolbar, BigSeekPro Toolbar, Blekko Search, Boby Lyrics, Bomlabio, Bonanza, BrandProfiles, Browse for the Cause, Browse to Save, BrowseBeyond, BrowseFox, BrowserSeek Search, Btosjs Info, Bubble Dock, Bueno Search, Buify, BuscaID Search, BuumpMe, Buzzdock, BuzzSearch, Certified Toolbar, ChatZum, Clickorati, Conduit Search, ContinueToSave, CoolLyrics, Coupon Alert Toolbar, Coupon Cactus, Coupon Caddy, Coupon Chaser, Coupon Companion, Coupon Genie, Coupon Locker, Coupon Matcher, Coupon Pigeon, Coupon Printer, Coupon Samurai, Coupon Server, Coupon Slider, Findwide Search, First Address Bar, Gigantic Savings, Ginyas Companion, Glarysoft Toolbar, Glindorus, GlobaSearch, Goong Search, GoOnSearch, Govome Search, GreyGray, Guffins Toolbar, Hot Search Toolbar, Hotspot Shield, Ievbz Search, iLivid Search, Illoxum, iMesh Toolbar, IMinent Toolbar, Infomash Search, Kozaka, Lyrics Bot, Lyrics-Fan, LyricsSing, Nav-Links, Oyodomo.

2.3.9. *Other malware*

In addition to the principal types of malware already presented here, there are other examples of this kind of malicious software that are always evolving and appearing as new types.

– *Ransomware*: ransom software, ransomware or extortion software is malicious software that takes hostage of personal information and/or systems. The release of this data/system depends on the payment of a sum of money (usually in bitcoin) called a ransom.

– *Scareware*: part of a class of malicious software that includes fake security software, ransomware and other fraudulent software that suggests paying to download fake software.

– *Phishing*: also known as swindling, this is a technique used by fraudsters to obtain personal information with the objective of stealing an identity.

– *Rootkits*: also called an “activity dissimulator tool”, “furtive malware”, or even a “hacker’s administrator toolkit” and sometimes a “kit”, is an ensemble of techniques implemented by one or more pieces of software,

with the objective of obtaining and prolonging (usually unauthorized) access to a computer in the most furtive way possible.

2.3.10. *Comparison of intrusion tools*

Intrusion tools present a diverse array of similarities; however, they have certain differences that are summarized in Table 2.2, which provides their intrinsic characteristics, the nature of the action in question and the symptoms of each tool.

Intrusion tool / Characteristics	Virus	Trojan horse	Worm	Software bomb
Intrinsic characteristics	– Self-replication – Destruction of information	– Hidden in a program – Opens a port	– Propagation through a network	– Starts after an event
Nature of actions (direct/indirect)	Direct	Indirect	Direct	Direct
Symptoms	– Parasitic messages – Parasitic folders	– Mouse movement – Opening/closing of windows – Loss of control of the computer – Interruption of usage	– Parasitic messages – Parasitic folders	No symptoms until it starts

Table 2.2. *Comparison of different intrusion tools*

2.4. Conclusion

Even though it is simple to set up, TCP/IP comes with numerous security weaknesses; given its widespread use, it is necessary to invest in security in order to avoid or at least minimize the problems presented.

TCP/IP is vulnerable to scanning and analysis tools that are widely available to end-users and exist in various forms on the Internet. The use of a scanning or analysis tool is a necessary step before proceeding to an attack.

Such weaknesses facilitate both the propagation of and attacks by intrusion tools. Intrusion tools and malware are, in a general sense, attack programs created to perturb the proper functioning of a system. There are currently four families of intrusion tools (virus, Trojan horse, worm, software bomb), but hybrid intrusion tools can exist as well; these act according to the characteristics of numerous families at the same time and are the most serious and most complex tools.

The development of software engineering has also facilitated the creation of spyware tools that are controlled by the intruder.

3

Authentication Techniques and Tools

3.1. Introduction

Authentication is the most fundamental aspect of security. It entails the definition of access rights and the identification of the source of an attack in the case of a problem. It allows access to various local and remote services to be controlled and limited. Authentication is a key mechanism of security since it constitutes the first barrier of security against potential attacks. The most important challenge for an attacker with authentication is to find the password that allows them to do what they want and thus compromise the system in question.

Authentication is accomplished through three possible techniques:

– The first technique, called "I have", is tied to the possession of an object that represents a proof of identity. This is the case with a badge or a card. This technique is most often used to ensure physical security and control access to critical locales.

– The second technique, called "I know", is tied to the knowledge of a secret combination which is the password. This is the most common method of providing authentication for a computing system.

– The third technique, called "I am", is the strongest form of authentication insofar as it is dependent on a personal characteristic of the user being authenticated (fingerprint, speech, etc.).

Computer Science Security: Concepts and Tools,
First Edition. Ameur Salem Zaidoun.

In certain scenarios, numerous techniques can be combined for better security, which is the case with banking cards ("I have" the card and "I know" the code).

The password must require a certain number of criteria in order to prevent it from being disclosed, which would potentially cause irreparable damage to the system or the entity requiring security. It must also have a limited lifetime.

In order to ensure authentication in a controlled and centralized way, AAA services were invented to offer a certain number of complementary functions that provide both authentication and an access log.

Authentication as a security service can be offered on network equipment via Telnet and SSH protocols, without any need for an AAA server. This method of authentication includes limits on the database of accounts that can be used to access the security configuration itself. Because of this, the configuration of an AAA server allows the deficiencies described above to be resolved and provides two additional functionalities, namely authorization and accountability, hence AAA.

An AAA service can use a local database, used only for authentication, or a remote database used to provide the three services: authentication, authorization and accountability. There are two possible modes of AAA: local or on a server.

3.2. Theoretical concepts of authentication

In order to avoid confusion, we must distinguish authentication from identification, two terms with different meanings and representing two complementary aspects of security. These two aspects can be provided in a coupled or separate manner.

3.2.1. *Identification*

This is the act of distinguishing ourselves from others by presenting our identity, which can be in the form of a name, an IP address, a physical address or a MAC address. Whatever is used as identification must be

unique in its context in order to be able to distinguish it, as is the case of a login or any other user identity.

3.2.2. *Authentication*

This is proof of identity through numerous techniques and is a way of guaranteeing that the entity presenting an identification is indeed the entity in question.

3.3. Different types of authentications

Depending on the entity to be authenticated and its place in the network, there are two different types of authentications, local and remote.

3.3.1. *Local service authentication*

To access local services, the most commonly used means of authentication is via a login and password. The login is unique for purposes of identification and the password is secret for purposes of authentication.

A password must possess a certain number of characteristics:

– It must be of reasonable size, which depends on the importance of the entity to secure.

– It must include a certain amount of varied content (capital and lowercase letters and special characters).

– Simple passwords should be avoided.

– Passwords from the dictionary should be avoided.

– Overly complex passwords that are difficult to memorize without having to write them down somewhere physical or electronic should be avoided.

– Passwords should be modified periodically according to their importance and use.

– Passwords should not be used for more than one account (email, credit card, etc.).

A realistic example of a good password is teurn@dior2C)21 (a password from the phrase “ordinateur2021”, *ordinateur* being French for “computer”. The syllables “or”, “di”, “na”, “teur” are inverted, “@” is used in the place of “a” and the number “0” is replaced by the sequence “C)”, a complex password that is easy to remember without needing to transcribe it digitally or physically).

In order to avoid any fraudulent connection based on trial and error, connection systems require a key to be entered (which appears in the form of an image) to ensure that the login and password provided were manually entered.

Certain authentication systems can be vulnerable to SQL injection attacks. These allow the required password to be circumvented since it can be ignored if it contains a special character that is used for comments (# or ‘ and others).

3.3.2. *Network authentication*

In order to access a remote service, each user must be authenticated. The password required for authentication must never be sent as plain text on the network. Indeed, any information circulating through the network can be captured by an observer (sniffer), and is the biggest challenge to overcome.

3.3.2.1. *Issues with remote authentication*

When accessing a remote service through a network, the most common form of authentication as well as the easiest is to send a login and password in plain text through the network, as presented in Figure 3.1. This technique provides both identification and authentication. This was the case for a wide array of webmail clients until the first years of the 21st century.

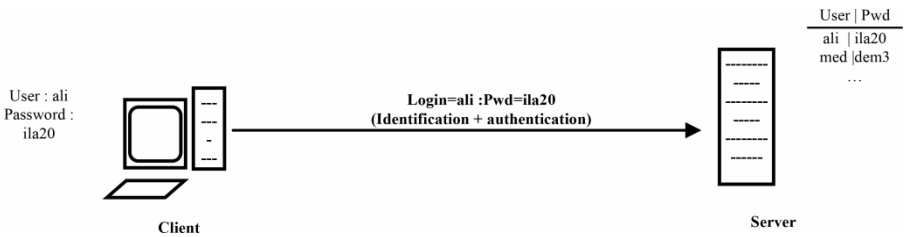

Figure 3.1. *Authentication by login and password in plain text*

By using a network analyzer, it became easy to detect a password, which called security into question. Because of this, in order to authenticate securely over a network it became necessary to avoid using this technique and sending a password in plain text through the network.

3.3.2.2. *Remote authentication by challenge message*

This consists of identifying someone first, and then authenticating that person by using the result of a hash function applied to a combination of the login, the password and a random challenge message sent via the server during authentication. It is accomplished in several phases.

– A client sends their login to the server to identify themselves.

– The server verifies the validity of the login.

– In the case of a valid identification, the server randomly generates a challenge message and sends it to the client.

– The client generates a chain of characters based on the following three elements: the login, the password and the challenge message. Then, applying an appropriate hash function to the chain generated in this way, it calculates a new chain that will be sent to the server.

– The server recreates the calculation using the same hash function applied to the same combination (login, associated password memorized by the server and the challenge message sent to the client). It then compares the result to the information that the client sent. If they are the same, the client will be authenticated.

The use of a different challenge message from one connection to the next prevents the hash code from being reused in a fraudulent authentication later on.

A hacker who watches the network can manage to detect the login, the challenge message and the hash code. These parameters are not enough to extract the password, however, even in the case where the formula combining the different elements (login, password, challenge message) and the hash function being used are known.

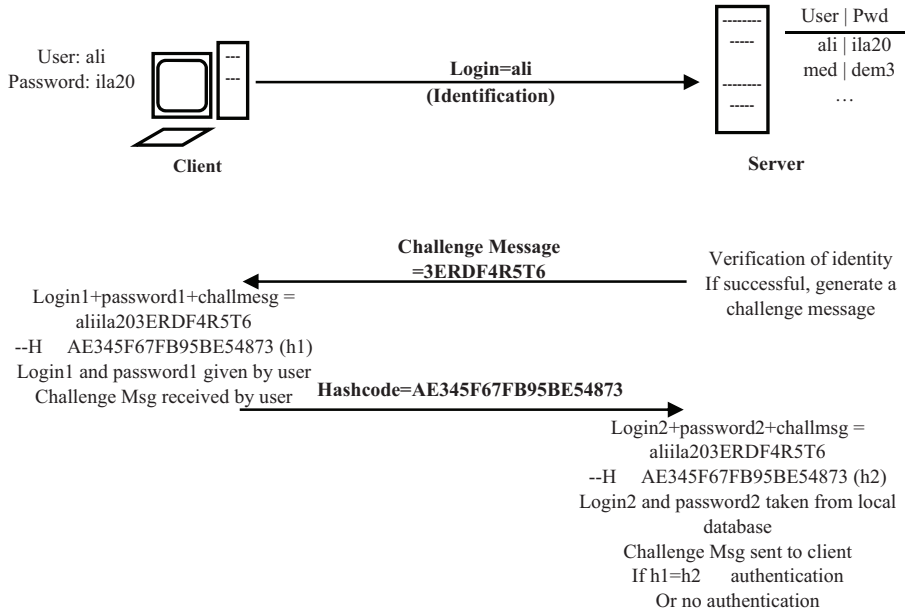

Figure 3.2. *Challenge message authentication*

3.3.2.3. *Remote authentication by single-use password*

Called a "one-time password", this technique is also based on the use of a hash function. It consists of changing the password of a connection to another one according to the following conditions:

– Passwords are generated using a root password that will be the last one used.

– A password that has already been used comes from the password that will follow.

– Passwords that have already been used cannot, under any circumstances, allow for any passwords that will be used ulteriorly to be determined.

Given H, a hash function, and n, a very large whole number, it is possible to be authenticated n+1 times by using just one chosen password root:

– The first authentication is completed using a password arrived at by applying the H function n times to the root password.

– The second authentication is completed using the password arrived at by applying the H function (n-1) times to the root password.

– etc.

– The nth authentication is completed using the password arrived at by applying the H function to the root password only once.

– The last authentication is completed using the root password.

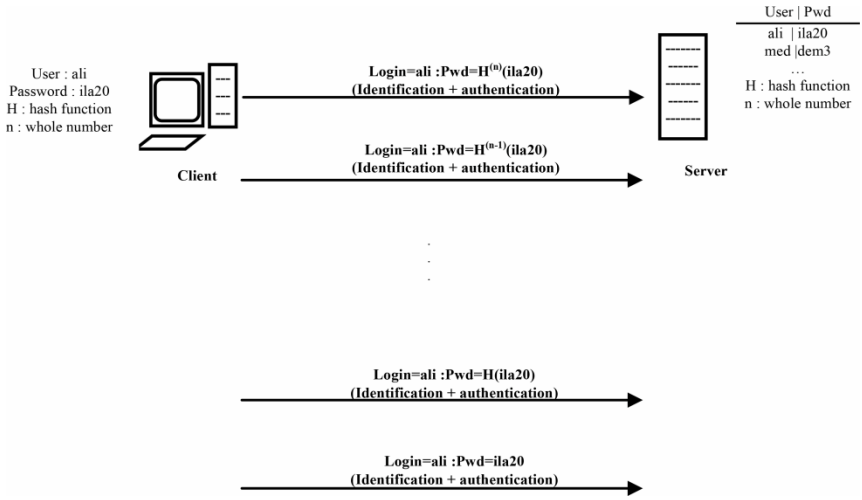

Figure 3.3. *Authentication by a one-time-use password*

A network observer who has managed to obtain all of the passwords already used will not be able to use them for any fraudulent authentication or to determine the root password chosen by the user.

3.3.2.4. *Telnet and SSH*

Telnet and SSH are two network protocols that are used to connect to a remote computer. We connect either to this system within a network using terminal emulators or on the Internet, controlling the system with remote commands.

Telnet is a protocol to remotely access a server or some equipment (router, switch, etc.) whose password is sent as plain text through the network, giving a network user or a sniffer the possibility of capturing it and using it for potentially fraudulent activity.

SSH, the acronym for secure shell, is a similar but more secure protocol insofar as the password cannot be captured via the network.

With Cisco equipment, it is possible to limit remote access to the Telnet protocol using the following configuration:

```
Router(config)#line vty 0 4
Router(config-line)#login
Router(config-line)#transport input telnet
```

Remote access to the SSH protocol can be limited via the following configuration:

```
Router(config)#line vty 0 4
Router(config-line)#login
Router(config-line)#transport input ssh
```

With Huawei equipment, it is possible to limit remote access to the Telnet protocol using the following configuration:

```
[Huawei]user-interface vty 0 4
[Huawei-ui-vty0-4]protocol inbound telnet
```

Remote access to the SSH protocol can be limited via the following configuration:

```
[Huawei]user-interface vty 0 4
[Huawei-ui-vty0-4]protocol inbound ssh
```

3.4. AAA service

AAA is the acronym for authentication, authorization and accounting.

Authentication is a function that allows an identity to be validated, with the objective of limiting access to a well-defined system. It responds to the question, “Who are you?”

Authorization is a function that determines the services which an authenticated entity may access, again with the objective of limiting access rights. It responds to the question “What rights do you have?”

Accounting is a function that allows all activity (access and completed tasks) to be recorded, with the objective of ensuring traceability for any possible subsequent analyses. It responds to the question "What have you done?"

AAA can be configured locally or based on authentication data hosted on a remote server.

3.4.1. *Local AAA*

Local AAA allows for only one function, which is authentication.

AAA authentication in the local mode is composed of the following steps:

– The user establishes a connection with the equipment.

– The equipment asks the user for a username and a password, authenticating the user by verifying this in a local database.

Local AAA authentication must be configured for smaller-sized networks. There are networks that allow one or two routers to provide access to a limited number of users. This method uses local usernames and passwords stored on a router. The system administrator must fill in the local security database by specifying the username and password profiles for each user who may log in.

The local AAA method of authentication is similar to using the local connection command. AAA also provides a means for configuring the methods of recording authentication.

3.4.1.1. *Local AAA authentication configuration*

On a Cisco router, AAA authentication in local mode consists of creating users, activating the AAA service and specifying the local database (case-sensitive) as a method for the default list of authentication logins:

```
Router(config)#username Bairam secret cisco
Router(config)#username Iyed password class
Router(config)#username Elaa password test
Router(config)#aaa new-model
Router(config)#aaa authentication login default local-case
```

It is also possible to create a named authentication method using the local database plus the password "enable" and fixing the application field for this method (remote SSH access):

```
Router(config)#aaa authentication login AUTH-SSH local
enable
Router(config)#line vty 0 4
Router(config)#login authentication AUTH-SSH
```

The general syntax of the configuration for an authentication method is:

```
Router(config)#aaa authentication login {default| LIST-NAME
} method1 [method2] ...
```

The local methods that can be applied are:

– *enable*: this consists of using the password "enable" for authentication;

– *local*: this consists of using the local database of users for authentication;

– *local-case*: this consists of using the local database of users (case-sensitive) for authentication;

– *none*: no authentication used.

On a Huawei router, the AAA authentication configuration in local mode consists of activating the AAA service, creating users, creating an authentication blueprint and specifying that it is in the local mode:

```
[Huawei]aaa
[Huawei-aaa]local-user Bairam password cipher huawei
[Huawei-aaa]local-user Iyed privilege level 12 password
cipher haina
[Huawei-aaa]local-user Elaa password cipher test
[Huawei-aaa]authentication-scheme LISTE-AUTHEN
[Huawei-aaa-authen-LISTE-AUTHEN]authentication-mode local
```

This method of authentication can be used for remote access to equipment, for example, using:

```
[Huawei]user-interface vty 0 4
[Huawei-ui-vty0-4]authentication-mode aaa
```

3.4.2. *Server AAA*

Server AAA authentication is composed of the following steps, as shown in Figure 3.4:

– The user establishes a connection with the equipment (1).

– The equipment asks the user for a username and password (2).

– The equipment transmits the username and password to a server (3).

– The server authenticates the user according to its database (4).

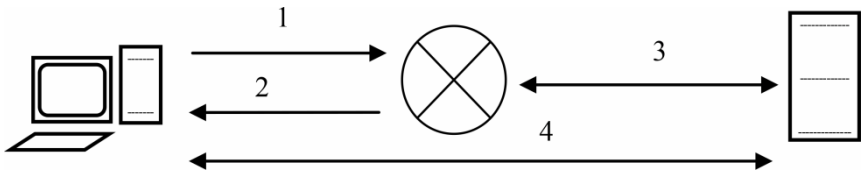

Figure 3.4. *AAA server authentication*

	RADIUS	TACACS+/HWTACACS
Function	Combines authentication and authorization but separates accounting, which allows for less flexibility	Separates AAA according to its architecture; ensures implementation modularity from the security server
Standard	Open (standard RFC)	Proprietary of Cisco/Huawei
Transport Protocol	UDP	TCP
CHAP Authentication	Unidirectional challenge and response from the RADUIS security server to the RADIUS client	Bidirectional challenge
Permitted Protocols	ARA and NetBEUI are not supported	All protocols
Confidentiality	Password encryption only	Encryption of the entire package
Personalization	No option to authorize router commands by user or group	Authorizes router commands by user or by group
Accounting	Extensive	Limited

Table 3.1. *Comparison of RADIUS and TACACS+/HWTACACS*

Communication between the network equipment, here a router, and the AAA security server is managed by one of the following protocols:

– *RADIUS*: a standard protocol.

– *TACACS+/HWTACACS*: a proprietary Cisco protocol/a proprietary Huawei protocol.

A comparison of the protocols is presented in Table 3.1.

The focus of the next two sections is the necessary configuration of servers and the activation of appropriate authentication methods on Cisco and Huawei equipment, respectively.

3.4.2.1. *Cisco server AAA authentication*

The server configuration of AAA authentication with CLI (Command Line Interface) is done by following the steps below:

– activate AAA;

– specify the IP address of the ACS server;

– configure the secret key;

– configure the authentication to use the RADIUS or TACACS+ server.

Activation of the AAA service is completed via:

```
Router(config)#aaa new-model
Router(config)#
```

The configuration of the TACACS+ and RADIUS server consists of:

```
Router(config)#tacacs server LE-SERVEUR-T
Router(config-server-tacacs)#address ipv4 192.168.1.10
Router(config-server-tacacs)#single-connection
Router(config-server-tacacs)#key TACACS-P@55w0rd
Router(config-server-tacacs)#exit
Router(config)#
Router(config)#radius server LE-SERVEUR-R
Router(config-radius-server)#address ipv4 192.168.1.11 auth-
port 1812 acct-port 1813
Router(config-radius-server)#key RADIUS-P@55w0rd
Router(config-radius-server)#exit
Router(config)#
```

The creation of a default authentication method making use of the RADIUS protocol is done through the configuration:

```
Router(config)# aaa authentication login default group
radius
```

For TACACS+, a named authentication method, for example, can be created using the following command:

```
Router(config)# aaa authentication login AUTH-TACACS group
tacacs+
```

3.4.2.2. *Huawei server AAA authentication*

The configuration of a RADIUS server is done via the following commands:

```
[HUAWEI] radius-server template SERVER-R
[HUAWEI-radius-SERVER-R] radius-server authentication
10.7.66.66 1812 weight 80
[HUAWEI-radius- SERVER-R] radius-server accounting
10.7.66.66 1813 weight 80
[HUAWEI-radius-SERVER-R] radius-server shared-key cipher
Radius@2020
[HUAWEI-radius-SERVER-R] radius-server retransmit 2
[HUAWEI-radius-SERVER-R] undo radius-server user-name
domain-included
[HUAWEI-radius-SERVER-R] quit
```

The activation of the RADIUS authentication consists of creating an authentication blueprint named AUTH-R and configuring the authentication protocol to use RADIUS authentication as the active mode of authentication:

```
[HUAWEI]aaa
[HUAWEI-aaa] authentication-scheme AUTH-R
[HUAWEI-aaa-authen-AUTH-R] authentication-mode radius
[HUAWEI-aaa-authen-AUTH-R] quit
```

The configuration of the HWTACACS server is done with the following commands:

```
[HUAWEI] hwtacacs-server template SERVER-T
[HUAWEI-hwtacacs-SERVER-R] hwtacacs-server authentication
10.7.66.66 49
```

```
[HUAWEI-hwtacacs-SERVER-R] hwtacacs-server authorization
10.7.66.66 49
[HUAWEI-hwtacacs-SERVER-R] hwtacacs-server accounting
10.7.66.66 49
[HUAWEI-hwtacacs-SERVER-R] hwtacacs-server shared-key cipher
Hwtacacs@2020
[HUAWEI-hwtacacs-SERVER-R] quit
```

The activation of the HWTACACS authentication consists of creating an authentication blueprint named AUTH-T and configuring the authentication protocol to use HWTACACS authentication as the active mode of authentication:

```
[HUAWEI]aaa
[HUAWEI-aaa] authentication-scheme AUTH-T
[HUAWEI-aaa-authen-AUTH-T] authentication-mode hwtacacs
[HUAWEI-aaa-authen-AUTH-T] quit
```

3.4.2.3. *AAA server authorization and accounting configuration*

Authorization is a means of granting or denying authenticated users access to certain zones and programs on the network, unlike authentication, which guarantees that a device or a final user is legitimate.

TACACS+/HWTACACS allows for the separation of authentication from authorization, while RADIUS does not separate authentication from authorization.

Under Cisco, the general syntax of the configuration of an authorization method is as follows:

```
Router(config)#aaa authorization {network|exec|commands
level} {default|LIST-NAME} method1 [method2] ...
```

And that of an accountability method is as follows:

```
Router(config)#aaa accounting {network|exec|connection}
{default|LIST-NAME} {start-stop|stop-only|none} [broadcast]
method1 [method2] ...
```

For example, an authorization and accounting method can be created using the following commands with the RADIUS and TACACS+ protocols, respectively:

```
Router(config)#aaa authorization exec default group radius
Router(config)#aaa accounting exec ACCT-METH stat-stop
groupe tacacs+
```

Under Huawei, the creation of an authorization blueprint named AUTHORIZ-T is followed by the configuration of the authorization protocol using HWTACACS authorization as the mode of authorization:

```
[HUAWEI-aaa] authorization-scheme AUTHORIZ-T
[HUAWEI-aaa-author-AUTHORIZ-T] authorization-mode hwtacacs
[HUAWEI-aaa-author-AUTHORIZ-T] quit
```

The creation of an accounting blueprint named ACCT-R is followed by the authorization protocol using the RADIUS authorization as the mode of authorization:

```
[HUAWEI-aaa] authorization-scheme ACCT-R
[HUAWEI-aaa-author-ACCT-R] authorization-mode radius
[HUAWEI-aaa-author-ACCT-R] accounting start-fail online
[HUAWEI-aaa-author-ACCT-R] quit
```

3.5. Conclusion

Authentication, whether it is local or through the network, must be done in a secure manner by choosing a viable password and using techniques that avoid sending the password in plain text.

Authentication is generally paired with integrity. We speak of authenticity, which ensures both the identity of the sender or generator, on the one hand, and the validity of the information sent or generated, on the other.

The AAA service that we have discussed in this chapter allows for access to different equipment and services to be managed in a methodical and modular way, keeping track of all of these activities. The configuration of these services on different equipment is a very important security measure that is generally neglected by management in favor of limited and outdated methods of authentication.

4

Techniques and Tools for Controlling Access, ACL and Firewalls

4.1. Introduction

Network traffic is always increasing in large measure and making the load on routers moving information around more and more difficult, and the quantity of data more and more voluminous. Moreover, a large part of this information constitutes parasitic elements controlled by hackers and unauthorized entities. The appropriate solution for this is to filter traffic and only allow legitimate parties to communicate through a network, which on the one hand limits the quantity of data processed, and on the other hand, eliminates potential sources of attack. These filters can be configured on routers and are named ACLs (access control lists).

To protect a critical location (laboratory, control room, etc.) in a company, it must be isolated and access to the location must be logged. In computing, measures for securing a system are similar and are done through a firewall that allows it to be isolated and for access to be tracked. A firewall is the most important security tool; it is the key element in any security measure. Moreover, the principle of firewalling and more specifically, filtering, is present in numerous security tools, and here, in the network connection equipment (Router, Level 3 Switch) in the form of ACLs.

Computer Science Security: Concepts and Tools,
First Edition. Ameur Salem Zaidoun.

4.2. Access control list

ACL is the acronym for access control list; it is a list of ordered filters called access control entry, which is applied to traffic leaving or entering through the LAN/WAN interface of a router.

ACLs allow packets to be filtered following the criteria defined by the user. It ensures that packets are filtered as they enter or leave a router interface depending on:

– source IP;

– destination IP;

– source port;

– destination port;

– protocol: IP, TCP, UDP, ICMP, etc.

An access control list is a router configuration script that controls whether packets and even frames are authorized or refused passage, following the criteria stipulated in their header. They are also used to select the type of traffic to be analyzed, transmitted or processed via other methods.

Within the router, only one ACL can be defined per protocol, per interface and per direction:

– protocol (IPv4, IPv6);

– interface (GigabitEthernet0/0, etc.);

– direction (incoming or outgoing traffic).

4.2.1. *ACL classification*

There are three possible classifications for ACLs, and these depend on the direction of traffic being managed by the ACL, on the type of ACL itself (its structure and complexity) and on the way the ACL is defined (numbered or named).

4.2.1.1. *Incoming/outgoing ACL*

Access control lists can apply to incoming or outgoing traffic.

– *Incoming access control lists*: incoming packets are processed before being routed to the outgoing interface. An incoming access control list is efficient because it reduces the routing load in case the packet is abandoned. If the packet is authorized after the tests, it is routed.

– *Outgoing access control lists*: incoming packets are routed to the outgoing interface and then processed via the outgoing access list.

4.2.1.2. *Standard/extended ACL*

There are two principal ACLs, defined by the criteria which used to control routing. We distinguish:

– *Standard access control lists*: these allow filters based only on IP sources. Basic nomenclature is used for Huawei equipment.

– *Extended access control lists*: these allow filters based on almost any field in IP, TCP and UDP headers. Advanced nomenclature is used for Huawei equipment.

For Huawei routers, it is also possible to define ACLs according to the MAC address; this is a layer 2 ACL.

4.2.1.3. *Numbered/named ACL*

In an ACL definition, a name can be given to it, or a number can be given instead. We distinguish:

– *Numbered ACL*: the ACL is identified by a number chosen while defining it.

For Cisco, the number allows the type of ACL to be identified as follows:

- (1–99) and (1300–1999): standard IP access control list;

- (100–199) and (2000–2699): extended IP access control list.

For Huawei, the number allows the type of ACL to be identified as follows:

- 2000–2999: basic IP access control list;

- 3000–3999: advanced access control list;

- 4000–4999: layer 2 access control list.

– *Named ACL*: the ACL is identified by a name chosen while defining it and by indicating the type (standard or extended):

- names can include alphanumeric characters;

- it is recommended that the name be written in capital letters;

- the names cannot contain spaces or punctuation marks; they must begin with a letter;

- entries in the access control list can be added or deleted.

4.2.2. *ACL configuration in Cisco*

An access control list is a sequential and ordered group of authorization or refusal instructions, called access control entries (ACE).

When network traffic crosses an interface configured with an access control list, the router compares information figuring in the packet to each rule in the ACL. The action specified by the first corresponding ACE is applied to the packet in question.

4.2.2.1. *Generic mask*

An IPv4 access control entry includes the use of a generic mask to filter IPv4 addresses.

A generic mask takes the form of an IPv4 address, and the bits that constitute it are interpreted as follows:

– *a bit at 0*: this allows for the establishment of a correspondence to the value of the bit at the corresponding IP address;

– *a bit at 1*: this allows the value of the bit at the corresponding IP address to be ignored.

As examples of generic masks associated with the IP address 192.168.1.1., we can distinguish:

– *0.0.0.255*: this allows the last byte to be ignored, which allows all addresses from the network 192.168.1.0/24 to be accepted;

– *0.0.0.7*: this allows the three last bits to be ignored, which allows all addresses from the network 192.168.0/29 to be accepted;

– *0.0.0.254*: this allows the first seven bits of the last byte to be ignored, which allows all the uneven addresses of the network 192.168.1.0/24 to be accepted;

– *0.0.0.0*: this allows the total correspondence of bits to be verified (with no bits being ignored), which allows only the address 192.168.1.1 to be accepted;

– *255.255.255.255*: this allows all bits to be ignored, which allows all IP addresses to be accepted.

The generic mask 0.0.0.0, which only allows one address to be accepted, can be replaced by the word host.

The generic mask 255.255.255.255, which allows all addresses to be accepted, can be replaced by the word any.

4.2.2.2. Configuration of numbered ACLs

The number attributed to an ACL indicates the type of ACL in question:

– from 1 to 99 and 1300 to 1999: *standard ACLs*;

– from 100 to 199 and 2000 to 2699: *extended ACLs*.

The general syntax is:

```
R(config)#access-list ACL-NUM {permit|deny} ...
```

As examples of standard ACLs:

```
R(config)#access-list 10 permit 192.168.1.0 0.0.0.255
R(config)#access-list 11 permit host 192.168.2.1
R(config)#access-list 11 permit host 192.168.3.254
R(config)#access-list 12 deny any
```

– ACL 10 authorizes packets with a source address from the network 192.168.1.0/24;

– ACL 11 only authorizes packets with a source address of either 192.168.2.1 or 192.168.3.254;

– ACL 12 denies access from all packets independently of their source addresses.

Extended ACLs provide filtering according to protocol (IP, TCP, UDP, ICMP, etc.), source IP addresses, destination and even both source and destination ports, TCP and UDP. A good illustration of this is:

```
R(config)#access-list 100 permit tcp 192.168.1.0 0.0.0.255
any eq 80
R(config)#access-list 101 deny ip any
R(config)#access-list 102 permit udp any host 192.168.4.100
eq 49
```

– ACL 100 allows for the authorization of TCP segments whose source address belongs to the network 192.168.1.0/24, representing Web traffic (relative to port 80, which can be replaced in the ACL configuration by www);

– ACL 101 allows all IP traffic to be blocked;

– ACL 102 allows all UDP traffic oriented towards computer 192.168.4.100 to be authorized (with port 49 corresponding to the TACACS+ service).

An implicit entry is added at the end of the ACL to block all other traffic.

Once the ACL has been configured, the administrator must specify the location (interface or access line of a router) in order to apply it. A standard ACL is applied as closely as possible to the destination and an extended ACL is applied as closely as possible to the source.

The command **ip access-group** is used to apply an ACL at the interface with the following syntax:

```
R(config-if)#ip access-group ACL-NUM| ACL-NAME {in|out}
```

As an example of this, ACL 101 can be applied to entering traffic from the GigabitEthernet 0/0/0 interface from router R as follows:

```
R(config)#interface GigabitEthernet 0/0/0
R(config-if)#ip access-group 101 in
```

The comma **ip access-class** is used to apply an ACL at the access line with the following syntax:

```
R(config-line)#ip access-class ACL-NAME {in|out}
```

As an example of this, ACL ADMIN-PLAGE-IP can be applied to filter authorized computers when accessing router R remotely:

```
R(config)#ip access-list
R(config)#line vty 0 4
R(config-if)#ip access-class ADMIN-PLAGE-IP in
```

4.2.2.3. *Configuration of named ACLs*

The configuration of a named ACL (which is the only option for configuring IPv6) consists in identifying the ACLs by their names while specifying whether it is a standard or extended ACL.

The general syntax is:

```
R(config)#ip access-list {standard|extended} NOM-ACL
```

For a standard ACL, the filtering rules are added by using the command with the following syntax:

```
R(config-std-nacl)#deny|permit IP-ADDRESS GENERIC-MASK
```

An example of this is:

```
R(config)#ip access-list standard NO-ACCESS-HOST-1
R(config-std-nacl)#deny host 192.168.1.1
R(config-std-nacl)#permit any
R(config-std-nacl)#exit
R(config)#interface g0/0
R(config-if)#ip access_group NO-ACCESS-HOST-1 out
```

The ACE (access control entry) rules are organized according to identifiers beginning with 10 and in multiples of 10 – 10, 20, 30, etc. – and adjusted as follows:

– deleting an ACE (20, for example):

```
R(config-std-nacl)#no 20
```

– redefining a deleted ACE:

```
R(config-std-nacl)#20 deny host 192.168.1.20
```

– inserting an ACE by indicating its identifier at the start:

```
R(config-std-nacl)#15 deny host 192.168.1.3
```

– adding a new ACE:

```
R(config-std-nacl)#permit any
```

We thus obtain the following list of ACEs:

```
10 deny host 192.168.1.1
15 deny host 192.168.1.3
20 deny host 192.168.1.20
30 permit any
```

For an extended ACL, the filtering rules are added using the command with the following syntax:

```
R(config-ext-nacl)#deny|permit {ip|icmp|udp|tcp} {SRC-IP-
ADDRESS GENERIC-MASK | any | host SRC-IP-ADDRESS}
[{eq|neq|gt|lt|range} PORT(S)-SRC] {DEST-IP-ADDRESS GENERIC-
MASK | any | host DEST-IP-ADDRESS} [{eq|neq|gt|lt|range}
PORT(S)-DEST]
```

For an example of an extended named ACL, we will create two lists:

– the first will be applied to traffic entering from the GigabitEthernet 0/0 interface and will let us authorize computers from the network 192.168.10.0/24 to only access Web traffic: http (port 80) and https (port 443);

– the second list will be applied to traffic leaving the same interface and will let us authorize all entering IP packets that are part of an already established connection.

```
R(config)#ip access-list extended SURFING
R(config-ext-nacl)#permit tcp 192.168.10.0 0.0.0.255 any eq
80
R(config-ext-nacl)#permit tcp 192.168.10.0 0.0.0.255 any eq
443
R(config-ext-nacl)#exit
R(config)#ip access-list extended BROWSING
R(config-ext-nacl)#permit tcp any 192.168.10.0 0.0.0.255
established
```

```
R(config-ext-nacl)#exit
R(config)#interface g0/0
R(config-if)#ip access-group SURFING in
R(config-if)#ip access-group BROWSING out
```

4.2.2.4. *IPv6 ACL*

IPv6 ACLs are uniquely named lists that function as extended IPv4 ACLs.

The configuration of an IPv6 access control list is done in three steps:

– Using the general configuration mode, the following command is used to create the IPv6 access control list:

```
R(config)#ipv6 access-list LIST-NAME
```

– In the named access control list configuration mode, the instructions `permit` and `deny` are used to specify one or more conditions determining whether a packet is transferred or abandoned.

```
R(config-ipv6-acl)#deny | permit {ipv6|icmp|udp|tcp}
{source-ipv6/prefix | any | host source-ipv6}
[{eq|neq|lt|gt|range} Source-Port-Number(s)] {destination-
ipv6/prefix | any | host destination -ipv6}
[{eq|neq|lt|gt|range} Destination-Port-Number(s)]
```

An IPv6 ACL is applied to traffic entering or exiting an interface with the command:

```
R(config-if)#ipv6 traffic-filter NOM-ACL in|out
```

As an example of an IPv6 ACL:

– the first instruction names the IPv6 access list NO-R3-LAN-ACCESS;

– the second instruction authorizes computer 2001:DB8:CAFE:30::1 to access any Web service;

– the third instruction refuses IPv6 packets from 2001:DB8:CAFE:30::/64 going to an IPv6 network;

– the fourth instruction authorizes all other IPv6 packets.

```
R(config)#ipv6 access-list NO-R3-LAN-ACCESS
R(config-ipv6-acl)#permit tcp host 2001:DB8:CAFE:30::1 any
eq 80
R(config-ipv6-acl)#deny ipv6 2001:DB8:CAFE:30::/64 any
R(config-ipv6-acl)#permit ipv6 any any
```

The ACL NO-R3-LAN-ACCESS is applied to traffic entering from the GigabitEthernet 0/0 interface with the following commands:

```
R(config)#interface g0/0
R(config-if)#ipv6 traffic-filter NO-R3-LAN-ACCESS in
```

4.2.3. *ACL configuration for Huawei*

With a Huawei router, it is possible to configure three types of ACLs, as described in Table 4.1.

Types	Value range	Parameters
Basic ACL	2000–2999	– Source IP address
Advanced ACL	3000–3999	– Source IP address – Destination IP address – Protocol – Source port – Destination port
Level 2 ACL	4000–4999	– MAC address

Table 4.1. *Types of ACLs for Huawei*

An ACL based on a number or a name can be created. An ACL is composed of many lists of rules containing authorization or denial clauses.

To create an ACL, the following parameters must be specified:

– When creating an ACL based on a number, the ACL number must be specified. The ACL number designates the type of ACL. For example, the ACL with a number between 2000 and 2999 is a basic ACL, and the ACL with a number between 3000 and 3999 is an advanced ACL.

– When creating an ACL based on a name, the name of the ACL must be specified. The number or the type of named ACL can be specified. If the

number of a named ACL is not specified, the system automatically attributes a named ACL number.

The general syntax for creating an ACL using either a number from the appropriate range that reflects the type of ACL or a name chosen by the administrator specifying the type of ACL or the appropriate number is either:

```
[R]acl [number] <ACL-Num>
[R-acl-<type>-<ACL-Num>]rule permit|deny …
```

or:

```
[R]acl name <ACL-Name> basic|advance|link|user|<ACL-Num>
[R-acl-<type>-<ACL-Num>]rule [<rule-id>] permit|deny …
```

With the command rule, the different filtering rules are created using the appropriate parameters for each type of ACL.

The rules, once created, are numbered 5, 10, 15, 20, etc. and are tested in this order for any packet; the first rule that matches will be applied. It is possible to integrate rules with well-chosen numbers.

The value of the default step is 5. It can be modified by the command:

```
[R-acl-<type>-<ACL-Num>]step <step-value>
```

To re-establish the default step, we use the command:

```
[R-acl-<type>-<ACL-Num>]undo step
```

4.2.3.1. *Basic ACL*

A basic ACL, which is attributed a number within the range of 2000 to 2999, is an ACL whose filtering rule is based only on the source IP address.

The configuration syntax of a basic ACL for Huawei is:

```
[R]acl [number] <ACL-Num(from 2000 to 2999)>
[R-acl-basic-ACL-Num]rule [<rule-id>] permit|deny source
<IP-ADDRESS> <Wildcard>
[R]interface <Type-Interface> <Interface-Num>
[R-InterfaceTypeInterfaceNum]traffic-filter inbound|outbound
acl <ACL-Num>
```

An ACL can also be defined in a different way, by attributing a name to it. In this case, the type of ACL must be specified (basic in this case) or by giving it a number from the range of values between 2000 and 2999.

```
[Huawei]acl name <ACL-Name> basic|<ACL-Num(from 2000 to
2999)>
```

4.2.3.2. *Advanced ACL*

An advanced ACL, which is attributed a number within the range of 3000 to 3999, is an ACL whose filtering rules are based on numerous parameters: protocol, source IP address, destination IP address, source port and destination port.

The configuration syntax of a basic ACL for Huawei is:

```
[R]acl [number] <ACL-Num(from 3000 to 3999)>
[R-acl-advance-ACL-Num]rule [<rule-id>] permit|deny
<Protocol> …
[R]interface <Type-Interface> <Num-Interface>
[R-InterfaceTypeInterfaceNum]traffic-filter inbound|outbound
acl <ACL-Num>
```

An ACL can also be defined in another way, by attributing a name to it. In this case, the type of ACL must be specified (advanced in this case) or by giving it a number from the range of values between 3000 and 3999.

```
[R]acl name <ACL-NAME> advance |<ACL-Num(from 3000 to 3999)>
```

The advanced ACL rules can be configured according to protocols for each IP. The parameters vary by protocol type.

When the type of protocol is ICMP, the format of the command is:

```
[R-acl-advance-ACL-Num]rule [ rule-id ] { deny | permit } {
protocol-number | icmp } [ destination { destination-address
destination-wildcard | any } | { { precedence precedence |
tos tos } * | dscp dscp } | { fragment | first-fragment } |
logging | icmp-type { icmp-name | icmp-type [ icmp-code ] }
| source { source-address source-wildcard | any } | time-
range time-name | ttl-expired | vpn-instance vpn-instance-
name ] *
```

When the type of protocol is TCP, the format of the command is:

```
[R-acl-advance-ACL-Num]rule [ rule-id ] { deny | permit } {
protocol-number | tcp } [ destination { destination-address
destination-wildcard | any } | destination-port { eq port |
gt port | lt port | range port-start port-end } | { {
precedence precedence | tos tos } * | dscp dscp } | {
fragment | first-fragment } | logging | source { source-
address source-wildcard | any } | source-port { eq port | gt
port | lt port | range port-start port-end } | tcp-flag {
ack | established | fin | psh | rst | syn | urg } * | time-
range time-name | ttl-expired | vpn-instance vpn-instance-
name ] *
```

When the type of protocol is UDP, the format of the command is:

```
[R-acl-advance-ACL-Num]rule [ rule-id ] { deny | permit } {
protocol-number | udp } [ destination { destination-address
destination-wildcard | any } | destination-port { eq port |
gt port | lt port | range port-start port-end } | { {
precedence precedence | tos tos } * | dscp dscp } | {
fragment | first-fragment } | logging | source { source-
address source-wildcard | any } | source-port { eq port | gt
port | lt port | range port-start port-end } | time-range
time-name | ttl-expired | vpn-instance vpn-instance-name ] *
```

When the type of protocol is GRE, IGMP, IP, IPINIP or OSPF, the format of the command is:

```
[R-acl-advance-ACL-Num]rule [ rule-id ] { deny | permit } {
protocol-number | gre | igmp | ip | ipinip | ospf } [
destination { destination-address destination-wildcard | any
} | { { precedence precedence | tos tos } * | dscp dscp } |
{ fragment | first-fragment } | logging | source { source-
address source-wildcard | any } | time-range time-name |
ttl-expired | vpn-instance vpn-instance-name ] *
```

For an example of an advanced named ACL, we will create an access control list that will be applied to traffic entering through the GigabitEthernet 0/0 interface and will authorize computers from the network 192.168.10.0/24 to access Web traffic only: http (port 80) and https (port 443):

```
[R]acl 3001
[R-acl-advance-3001]rule permit tcp destination 192.168.10.0
0.0.0.255 destination-port eq 80 eq 443 source any
[R] interface GigabitEthernet 0/0
[R- GigabitEthernet0/0]traffic-filter inbound acl 3001
```

4.2.3.3. *Level 2 ACL*

A level 2 ACL, which is attributed a number within the range of 4000 to 4999, is an ACL whose filtering rules are based on information about the connection level, including the MAC source, the VLAN source ID, the type of level 2 protocol and the destination MAC address.

The configuration syntax of a level 2 ACL for Huawei is:

```
[R]acl [number] <ACL-Num(from 4000 to 4999)>
[R-acl-basic-ACL-Num] rule [ rule-id ] { permit | deny } [ {
ether-ii | 802.3 | snap } | l2-protocol type-value [ type-
mask ]]
[R]interface <Type-Interface> <Num-Interface>
[R-InterfaceTypeInterfaceNum]traffic-filter inbound|outbound
acl <ACL-Num>
```

An ACL can also be defined in another way, by giving it a name. In this case, the type of ACL must be specified (link, in this case) or by attributing a number to it from the range of values between 4000 and 4999.

```
[Huawei]acl name <ACL-NAME> link|<ACL-Num(from 4000 to
4999)>
```

4.3. Firewall

A firewall is software (for a router or terminal equipment: PC, server, workstation) or physical equipment that allows a computer, network or sub-network to be secured.

A firewall has three principal functions: filtering, tracing and in some cases, network address translation (NAT).

4.3.1. *Filtering function*

This function isolates the entity that is to be secured by filtering the entities that access it through the MAC address, IP address, Web address (URL), port number, protocol, etc.

Numerous rules can be applied:

– denying some and authorizing the rest (allowing access by default) is an "optimistic" policy;

– authorizing some and denying the rest (preventing access by default) is a "pessimistic" policy;

– allowing the user to decide along the way whether to authorize or deny an entity is an interactive policy.

The filtering policy as well as the filters must be chosen by the administrator and updated periodically. The strength of a firewall depends on the precision and exactitude of its filtering configuration.

Depending on the entity to filter, there are two modes for filtering, either at a high or low level.

We can also distinguish two types of filters by the way they function, either with or without states.

4.3.1.1. *Low-level filtering*

Also called packet filtering, this allows packets to be filtered and sometimes even frames depending on their principal characteristics:

– source/destination MAC address;

– source/destination IP address;

– protocol (IP, ICMP, UDP, TCP, etc.);

– source/destination port, etc.

4.3.1.2. *High-level filtering*

Also called application filtering or even content filtering, this concerns applications and Web pages. It is a kind of filter based on high-level parameters:

– execution, network access, access to the register log for applications, called application filtering;

– URL addresses, called URL filtering;

– list of key words present in the Web page, called content filtering.

4.3.1.3. *Stateless filtering*

Stateless filtering consists of filters based only on information concerning the entity to be filtered, independently of its state or progress. Even though it is superficial, this filtering does not take up many resources from the CPU and memory.

4.3.1.4. *Stateful filtering*

Stateful filtering consists of maintaining the context of different sessions in the memory. It is possible in this case to dynamically modify the filtering rules. A good illustration of this is the authorization of packets that are responses to connections initialized from within the network and for packets coming from a connection attempt from outside the network to be refused, as illustrated in the following figure.

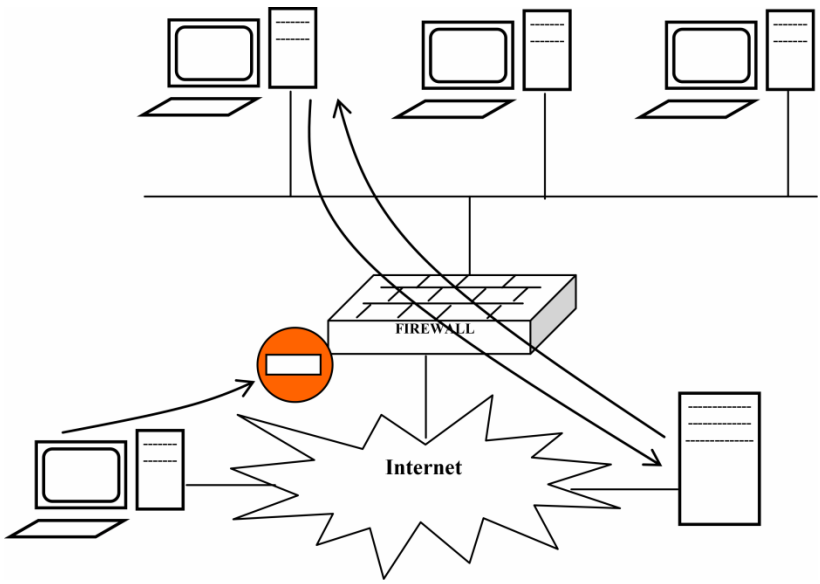

Figure 4.1. *Firewall supporting stateful filtering. For a color version of this figure, see www.iste.co.uk/zaidoun/computer.zip*

4.3.2. *Functionalities of tracing and NAT*

In order to identify a source of attack, it is necessary to record all access events and attempts at access. The information that is generated in this way is called the log. The administrator must frequently check the log to identify sources of attack and analyze the state of communication.

This daily log of different activities is necessary for the administrator, who can analyze them manually or by using appropriate tools to detect any unauthorized access or access attempts.

NAT is the acronym for network address translation. It is a router or firewall function that secures a network. It hides local IP addresses (private addresses) during external communication (on the Internet) and replaces them with public addresses. The purpose is to protect certain computers that are inaccessible directly from the exterior, save routable IP addresses and facilitate network maintenance.

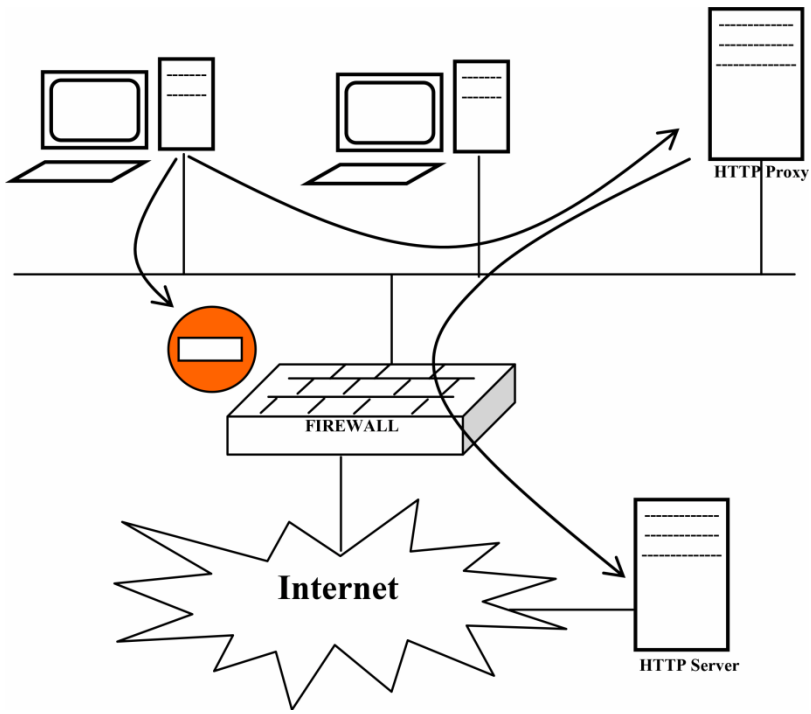

Figure 4.2. *Movement of data with a proxy*

There are two ways of translating addresses, static and dynamic. The static translation of addresses (one-to-one) creates a correspondence between a private address and a public address. The dynamic translation of addresses (N-to-M with M<N) consists in attributing public addresses (from an address pool) as needed. Port pools are also used (PAT: port address translation) to identify the relevant internal address when a packet comes from the exterior, since an external address could be allocated to numerous computers.

NAT is a more evolved technique than using a proxy. A proxy, as shown in Figure 4.2, is an intermediary that relays an application (http) to its users. The use of the application is only open to the proxy and workstations must go through the proxy.

4.3.3. *Firewall architecture*

The general architecture of a firewall is depicted in Figure 4.3.

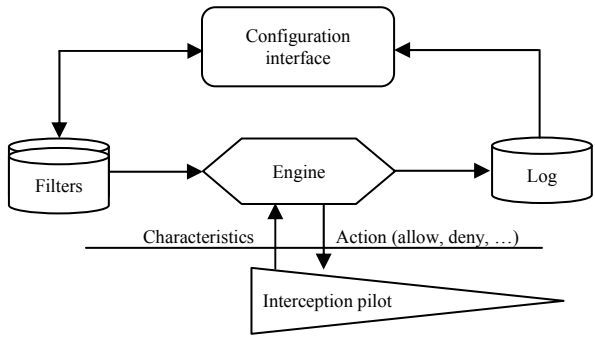

Figure 4.3. *Firewall architecture*

This security solution is composed of three parts:

– interception pilot;

– engine;

– configuration interface.

This allows two data forms to be altered:

– filter list;

– daily logs.

4.3.3.1. *Interception pilot*

This is a lower-level system component that allows packets, frames or entities to be captured and for decisions to be applied to them. This component is responsible for the detection and application of an action. It takes the form of a system process that must have the necessary privileges, which allows it to act at the key moment and avoid any actions of the entity in question to get through.

4.3.3.2. *Engine*

This is a higher-level component that analyzes the set of filters (generic) set up by the administrator and decides what action should be applied to an entity depending on its characteristics.

The configuration of the firewall appears as a list of different filters, both complex and complimentary, which means the appropriate action is not intuitive nor easy to identify. Such a characteristic requires a deeper analysis to be able to resolve the task of filtering a well-defined entity. This is the principal function of the engine.

4.3.3.3. *Configuration interface*

This allows the administrator to choose appropriate filters. There are two types of interfaces, appearing as either text through an administrative console (CLI: command line interface) or graphics through a user-friendly graphic interface (GUI: graphic user interface). The first option requires an administrator with expertise.

4.3.3.4. *Filter database*

This is a database that contains the configuration of the firewall. The filters are defined generically, and the engine is in charge of defining the order of priority going from the most specific to the most general. The action applied is determined from the entities in question, the circumstances and the general policy in use within the firewall.

4.3.3.5. *Logs*

These are the records of access and access attempts saved by the firewall and any other event concerning the firewall. The records are archived (by system date and time) so the administrator can use them to identify a well-defined problem.

4.3.4. *How a firewall works*

The interception pilot captures the entity (packet, application, etc.) and sends its characteristics (source, destination, etc.) to the engine, which determines the action (authorize, deny, etc.) by seeking the filter to apply to this entity. The group of applicable filters is chosen by the administrator during configuration.

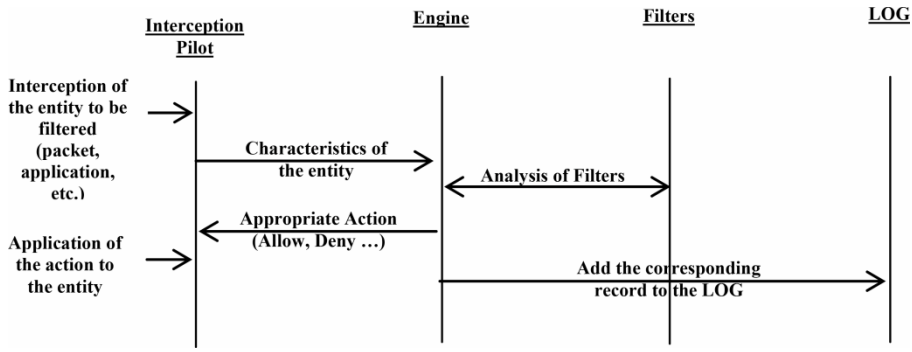

Figure 4.4. *Firewall filtering*

4.3.5. *Firewall classifications*

There are two possible types of classification for firewalls. The first classification is according to the level of filtering with respect to the TCP/IP stack, and the second with respect to the field of action, that is, the part that needs to be secured by the firewall.

4.3.5.1. *Filtering by level of action*

This classification responds to the question: at what level does the filtering occur?

We speak of packet filtering or low-level filtering in the first case and applied filtering or high-level filtering in the other case.

Packet firewall

In this class of firewalls, the entity that is to be filtered is the packet or sometimes the frame. It comes into play at levels 2, 3 and 4 of the OSI model.

These are filtered with respect to:

– physical source and destination addresses (@MAC);

– logical source and destination addresses (@IP);

– protocols used (IP, ICMP, TCP, UDP, SLIP, PPP, etc.);

– source and destination port numbers, etc.

Application firewall

In this class of firewalls, the entity that is to be filtered is the application. This comes into play at level 7 of the OSI model.

Applications are filtered with respect to:

– Web traffic (URL, keywords, cookies, etc.);

– application rights (execution, network access, log access, disk access, etc.).

4.3.5.2. *Filtering by field of action*

This classification responds to the question: what is the firewall securing?

The first generations of firewalls were made to secure a network (Intranet) from the outside (Internet) given that the latter was considered the principal source of attacks. Over time, statistics have shown that 80% of attacks come from within. Firewalls that secure a computer appeared to be the solution.

In order to avoid configuration difficulties related to PC firewalls, distributed firewalls were composed of agents and an administration server.

Classic firewall

Any physical firewall that secures a network is considered a classic firewall. We also speak of box firewalls. This equipment is used between the local network to be protected and the external network (Internet), the potential source of attack.

PC firewall

This is software that secures a computer and secures others from it. The reason for the existence of this new generation of firewalls comes from the fact that statistics have shown that 80% of attacks come from an internal

source. The PC firewall is used as an antivirus in homes, and there are numerous versions available for free on the Internet.

Distributed firewall

This is a recent network solution that secures different computers using a centralized configuration. There is a filtering agent for each computer that functions as a PC firewall and whose configuration comes from the central computer in charge of collecting the log. The administrator does not need to go computer-by-computer to reconfigure and analyze the log, nor do they have to leave this task to the user, who is not necessarily trained to do this.

4.3.6. *Stateful firewall*

A stateful firewall is a firewall that provides filtering based on the state of the network connection. The objective is to secure the local network against any exterior access attempt (from the Internet). This function allows for a distinction between packets that are part of a response to an internal request (authorized) and a packet from a new connection initiated from the outside (denied). For this, the firewall inspects traffic initiated from the local network so it can know how to distinguish the response to an internal request from the rest of the entering traffic and determine which action should be applied.

One possible implementation through a Cisco router (IOS updated to support the firewall) is based on ACLs. This is done by:

– Choosing internal and external interfaces.

For example, GigabitEthernet 0/0 for the internal interface; Serial 0/0/0 for the external interface.

– Configuring the ACLs for each interface.

For example, an ACL blocking entering traffic from any external interface:

```
R(config)#ip access-list extended OUTSIDE
R(config-ext-nacl)#deny ip any any
R(config)#interface Serial 0/0/0
R(config-if)#ip access-group OUTSIDE in
```

Or, for example, an ACL allowing authorized entering traffic from any external interface to be defined:

```
R(config)#ip access-list extended INSIDE
R(config-ext-nacl)#permit tcp any any eq 80
R(config-ext-nacl)#deny ip any any
R(config)#interface GigabitEthernet 0/0
R(config-if)#ip access-group INSIDE in
```

– Defining the inspection rules.

For example, creating an inspection rule:

```
R(config)#ip inspect name FWSF http
```

– Applying the inspection rule to an interface.

For example, applying the inspection rule to entering traffic from any internal interface:

```
R(config)#interface GigabitEthernet 0/0
R(config-if)#ip inspect FWSF in
```

4.3.7. *Zone-based firewall*

A ZPF, or zone-based policy firewall is a firewall that provides a flexible configuration that is independent of the ACLs and physical interfaces.

4.3.7.1. *General presentation of the notion of zones*

This new type of configuration in which interfaces are attributed to security zones and the firewall strategy is applied to traffic circulating in these zones consists of:

– determining zones;

– establishing an inter-zone policy;

– developing a physical infrastructure;

– identifying subgroups within zones and combining traffic requirements.

The Cisco IOS ZPF can take one of three actions:

– *Inspect*: this allows the packets to be inspected with stateful Cisco IOS.

– *Drop*: analogous to the command "deny" in an ACL, this deletes a packet. A daily log option is available to record rejected packets.

– *Pass*: analogous to the command "permit" in an ACL, this authorizes a packet. The successful action does not follow the state of connections or sessions in the traffic.

It is possible to create two or more zones, and a physical interface may or may not be affected by a zone; a specific zone called a self-zone corresponds to the router itself and contains all the IP addresses attributed to these interfaces.

Once the security zones have been created, the administrator can create pairs of zones and then test the security for some or all of these pairs of zones.

The action (pass, drop or inspect) applied to traffic between interfaces, as presented in Table 4.2, depends on the source and the destination on the one hand, and, on the other hand, the configuration of the ZPF firewall, in this case the existence of a pair of zones among the aforementioned zones or a security policy that will be applied to the zone-pair in question.

Source interface: member of a zone?	**Destination interface: member of a zone?**	**Does the zone-pair exist?**	**Does a policy exist?**	**Action**
NO	NO	-	-	**PASS**
YES	NO	-	-	**DROP**
NO	YES	-	-	**DROP**
YES (Zone **X**)	YES (Zone **X**)	-	-	**PASS**
YES (Zone **X**)	YES (Zone **Y**)	NO	-	**DROP**
YES (Zone **X**)	YES (Zone **Y**)	YES	NO	**DROP**
YES (Zone **X**)	YES (Zone **Y**)	YES	YES	**INSPECT**
YES (Self-Zone)	YES (Zone **Y**)	NO	-	**PASS**
YES (Self-Zone)	YES (Zone **Y**)	YES	NO	**PASS**
YES (Self-Zone)	YES (Zone **Y**)	YES	YES	**INSPECT**
YES (Zone **X**)	YES (Self-Zone)	NO	-	**PASS**
YES (Zone **X**)	YES (Self-Zone)	YES	NO	**PASS**
YES (Zone **X**)	YES (Self-Zone)	YES	YES	**INSPECT**

Table 4.2. *Filtering rules for traffic between interfaces with a ZPF firewall*

4.3.7.2. *Configuration and implementation of a ZPF firewall*

The configuration includes five steps:

– Zones are created using the command "`zone security <zone-sec-name>`":

```
R(config)#zone security PRIVATE
R(config-sec-zone)#exit
R(config)# zone security PUBLIC
```

– Traffic is identified using the command "`class-map type inspect`":

```
R(config)#class-map type inspect {match-all|match-any}
<class-map-name>
R(config-cmap)#match access-group {acl-name|acl-name}
R(config-cmap)#match protocol <protocol-name>
R(config-cmap)#match class-map <class-map-name>
```

The option "`match-all`" means packets must satisfy all criteria mentioned while the option "`match-any`" will permit a packet that satisfies any one of the criteria from those mentioned.

If we want to create a class-map that includes all Web traffic, then it is configured as follows:

```
R(config)#class-map type inspect match-any WEB-TRAFFIC
R(config-cmap)#match protocol http
R(config-cmap)#match protocol https
R(config-cmap)#match protocol dns
```

– An action can be defined using the command "`policy-map type inspect <policy-map-name>`":

```
R(config)# policy-map type inspect PRIV-2-PUB-POLICY
R(config-pmap)#class type inspect WEB-TRAFFIC
R(config-pmap-c)#inspect
```

This command specifies a class-map that identifies the relevant traffic and chooses the action applied (drop, pass or inspect).

– A zone-pair can be identified using the command "`zone-pair security <zone-pair-name> source <zone-sec-name> destination <zone-sec-name>`" and attribution of a security policy using the command "`service-policy type inspect <policy-map-name>`":

```
R(config)# zone-pair security PRIV-PUB source PRIVATE
destination PUBLIC
R(config-sec-zone-pair)# service-policy type inspect PRIV-2-
PUB-POLICY
```

– Physical interfaces are attributed to different zones using the command "`zone-member security <zone-sec-name>`":

```
R(config)#interface GigabitEthernet 0/0
R(config-if)# zone-member security PRIVATE
R(config-if)#exit
R(config)#interface Serial 0/0/0
R(config-if)# zone-member security PUBLIC
```

4.3.8. *Firewall examples*

This section will present an example of a physical firewall and two examples of software firewalls.

4.3.8.1. *Cisco PIX*

Cisco PIX (private internet exchange) is the box firewall from the company Cisco Systems.

The range of Cisco PIX combines robust firewall and VPN functions as well as intelligent network services in one compact and reliable platform. Cisco PIX can be configured using either a console cable or via Web or Telnet access from the network. It is also possible to connect from a telephone line using a modem.

4.3.8.2. *Iptables*

Iptables is a freely accessible software on Linux that a system administrator can use to configure chains and rules in the firewall of a central computer (which is made of Netfilter modules).

Different programs are used depending on the selected protocol: Iptables is used for the IPv4 protocol, Ipt6tables for IPv6, Arptables for ARP (Address Resolution Protocol) or Ebtables for Ethernet frames.

This type of modification must be restricted to system administrators. As a result, access to a root account is needed to use it. Other users are barred from using the program.

On the majority of Linux versions, Iptables can be started with the command "/usr/sbin/iptables"; this is documented in the manual pages of iptables and ip6tables, which can be consulted via the commands "man iptables" or "man ip6tables".

Iptables is also frequently used to refer to low-level components (kernel level). X_tables is the name of the central module, which is more general, and which contains the shared code for the four protocols. This is also the module that provides extension APIs. As a result, X_tables refers to the entire firewall in general usage (IPv4, IPv6, arp, eb).

4.3.8.3. *ISA server*

An ISA (Internet Security and Acceleration) Server is a Microsoft firewall; it is a company server firewall with extendable cache for the Web. It offers security functions by strategy, for the acceleration and management of network connections. ISA servers have two intertwined modes: a multi-level firewall server and a high-performance cache server for the Web. The firewall provides:

– filtering by packet, circuit and application levels;

– states analysis to examine the data going through the firewall;

– control of the access strategy and traffic routing.

The cache improves the performance of the network and the final user experience by storing frequently-used Web content. The firewall or cache services can be deployed separately on dedicated servers or integrated on one computer.

4.3.8.4. *Cisco ASA firewall*

The Cisco Adaptive Security Appliance (ASA) provides a complete and tested firewall alternative. It offers superior upgrade possibilities, a large array of technological solutions and security that is both efficient and permanent, created to respond to the needs of a great variety of uses.

ASA supports many services (routing, AAA, etc.) in addition to the firewall function. It makes use of a specific syntax, which is different from that used in other Cisco equipment. Moreover, ACLs in ASAs are defined and applied by using different commands from those used by routers.

Cisco has recently started to sell different models of its firewall, for example: ASA 5505, ASA 5506-X, ASA 5512-X, ASA 5525-x, ASA 5555-X, etc. in addition to its ASA module in the form of a card that can be inserted into another compatible system.

ASA can be used via different types of architecture and can make use of the Active Directory service from Microsoft. It can moreover be used to create a DMZ zone.

4.4. The concept of a DMZ

DMZ is the acronym for a demilitarized zone. It is part of a local network containing services that are accessible to the outside, such as a Web service. It is a zone that is monitored but without the strong filtering of the rest of the more secure local network.

The usefulness of the DMZ is to isolate the services that are visible outside of the rest of the network with the purpose of defining a different filtering policy with a stateful firewall filter. Any traffic initiated outside gets authorized for the DMZ and denied for the rest of the network; in the other direction, only the outgoing traffic from the secured part gets authorized.

4.4.1. *Implementation of topologies*

There are two possible ways to create a DMZ topology. The first is done by creating the DMZ as a branch of the firewall. The second consists of demarcating a DMZ between two firewalls (internal and external).

4.4.1.1. *Topology with one firewall*

The DMZ segment is attached to a branch of the LAN that is independent of the firewall (some firewalls come with a branch reserved for the DMZ).

In the case where the firewall only allows one LAN entry, we opt for VLAN technology to separate the DMZ from the rest of the local network. We can also separate the secured area in numerous physical or virtual (VLAN) segments with the objective of defining different security policies. This is what happens, for example, when separating internal services from workstations.

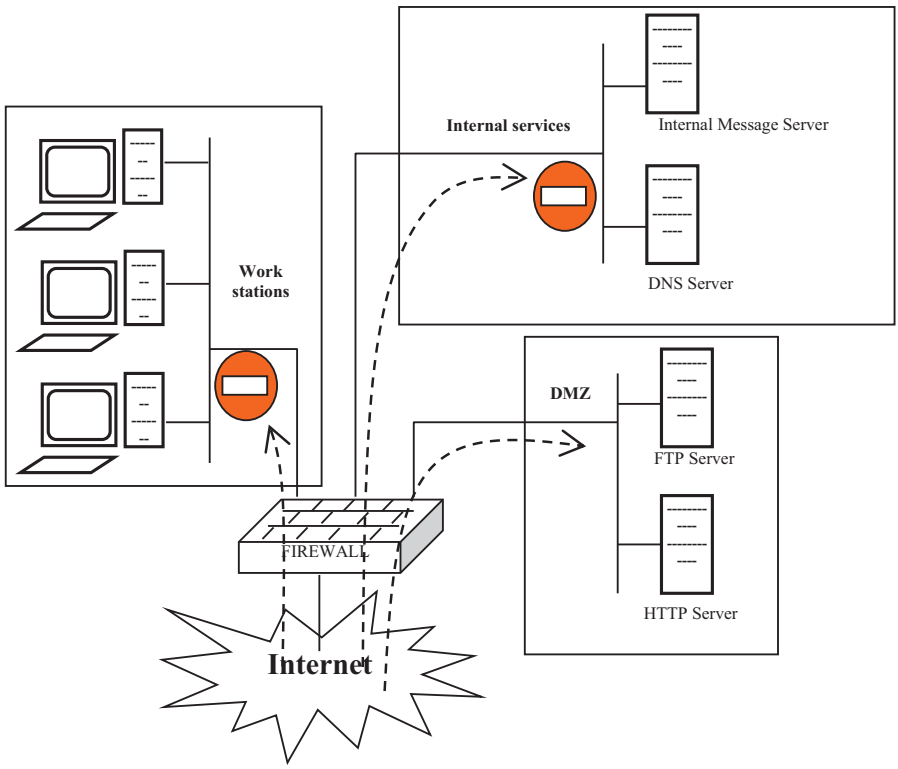

Figure 4.5. *DMZ implementation with one firewall*

In the case of only one LAN branch, we have a unique physical segment containing all the internal equipment. We opt for software segmentation via VLANs (virtual local area network). There are three VLAN technologies.

The VLAN technology using a physical port attributes each port of a switch to a VLAN (identified by its number). This is the simplest but most rigid option in the sense that a computer risks changing the virtual segment by changing the connection port.

The VLAN technology using a MAC address offers security by port and assigns a fixed MAC address to each port. Otherwise, it learns and creates virtual segments (DMZ, etc.), which allows it to emulate a physical distribution of segments.

Table 4.3 shows the general filtering policies applied by the firewall between the three segments and the Internet, each one with respect to the others.

From / To	Workstations	Internal services	DMZ	Internet
Workstations		ALLOW (Access possible from within)	ALLOW (Access possible from within)	ALLOW (Internet access possible)
Internal services	DENY (No users)		DENY (No users)	DENY (No users)
DMZ	DENY (No users)	DENY (No users)		DENY (No users)
Internet	DENY (No services)	DENY (Internal services)	ALLOW (Services accessible from outside)	

Table 4.3. *Traffic filtering in DMZ implementation with one firewall*

4.4.1.2. *Topology with two firewalls*

The segment for the DMZ is demarcated between two firewalls, one external firewall that filters traffic between the Internet and the local network, and one internal firewall that filters the traffic in the secured part even more, which makes access to that part even more difficult (by passing via two firewalls).

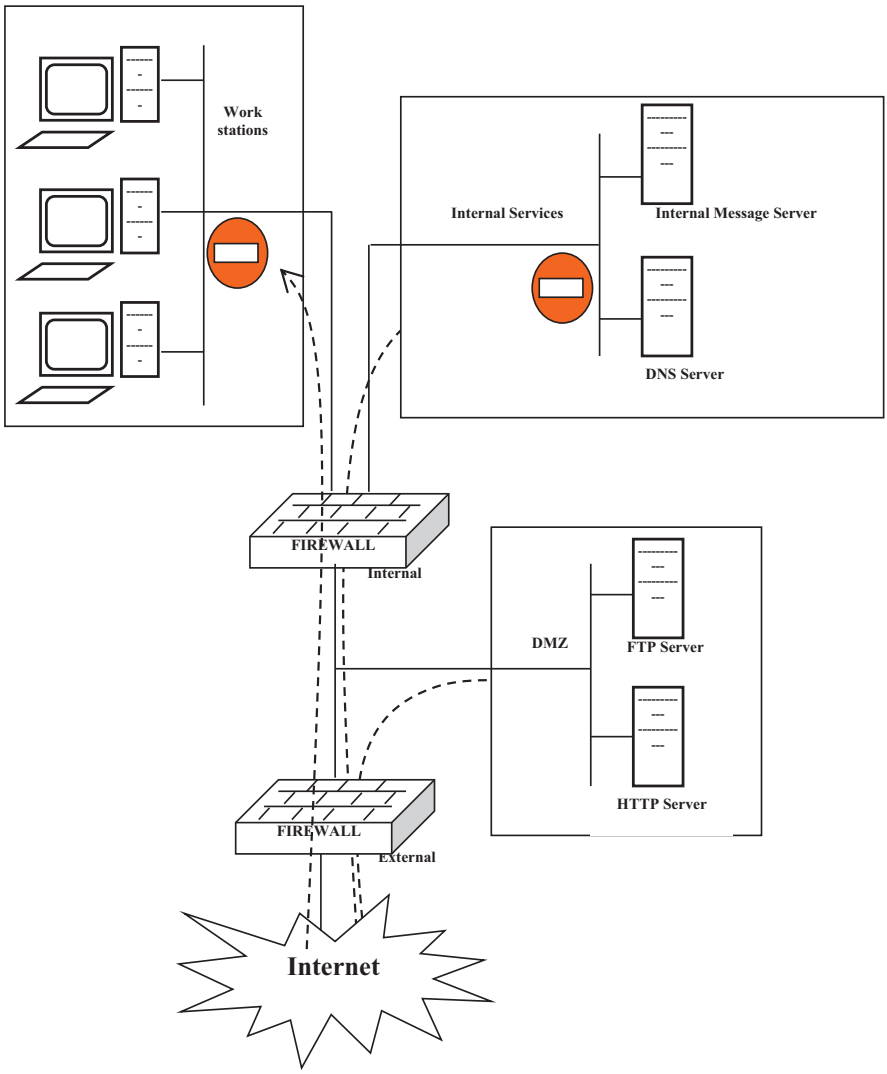

Figure 4.6. *Firewall implementation with two firewalls*

4.5. Conclusion

Controlling access physically as well as virtually is a necessary security measure that reduces the risk of fraudulent access and allows all connections and connection attempts to be monitored. This is provided, as a software component, by ACLs and firewalls.

ACLs are a primitive method of controlling and limiting network traffic. They provide a level of both physical and virtual security to the computing system. This offers diverse static and dynamic filtering functions at many levels. Although they are important and efficient, an evolution of this principle was developed using dedicated equipment called a firewall.

Firewalls are an indispensable security solution for a company. The use of a physical firewall is necessary to secure company-wide remote access, while the use of virtual firewalls remains a necessary security measure to secure different servers and workstations, as well as the physical and virtual assets of a company. However, a firewall can never resolve the problem of security without being combined with other solutions, such as antivirus software and IDS.

5

Techniques and Tools for Detecting Intrusions

5.1. Introduction

Viruses are the most serious, widespread and well-known kind of intrusion. A large number of malwares, especially viruses, are produced every day. For this kind of problem, a protection tool is necessary, hence antiviruses have been developed as a specific security solution.

Firewalls are a general security solution that limits the probability of attacks, whereas antiviruses are a specific security solution whose objective is to protect computers from the virus. However, these two solutions remain limited in guaranteeing the security of a computer system. Therefore, the use of an intrusion detection system is necessary to ensure security, as well as to overcome the insufficiencies of both the firewall and the antivirus. It is a more complex solution that uses diverse means to detect and disinfect intrusion tools.

5.2. Antivirus

This is software that detects, isolates and destroys viruses on hard drives, external drives and memory.

5.2.1. *Functions of an antivirus*

An antivirus has three functions: detection, isolation and destruction

Computer Science Security: Concepts and Tools,
First Edition. Ameur Salem Zaidoun.

– *Detection*: an antivirus must be able to detect known viruses by analyzing files and searching for their signatures. In some cases, it can detect unknown viruses using heuristics and by their behavior.

– *Isolation*: this involves quarantining a virus or an infected program to prevent it from either reproducing or damaging the computer system.

– *Destruction*: files related to the virus are deleted to limit its reproduction and propagation.

5.2.2. *Methods for detecting a virus*

The first generations of antiviruses were limited to only known viruses by searching for the viral base, whereas new generations have become increasingly intelligent in detecting viruses, even if their signatures are not found in the viral base, by using techniques from industrial research.

5.2.2.1. *Detection of known viruses*

Known viruses can be detected using their signature, which is a chain of characters that represents the virus (which is the result of a hash function). This emphasizes the need for regular updates in order to include the signatures of new viruses for their detection.

5.2.2.2. *Detection of unknown viruses*

The detection of unknown viruses is a difficult and inefficient task, which is based on a collection of complementary methods as follows:

– *Behavioral analysis*: this involves detecting viral behavior within programs, for example, the appearance of parasitic files or the loss of data.

– *Integrity control*: this involves saving the signatures of different programs and applications in a computer and periodically inspecting changes that can occur due to a virus attack.

– *Multi-level generic detection*: this is a rather complicated technique for detecting polymorphous viruses whose signature changes in form.

5.2.3. *Actions taken by an antivirus*

Even though the use of an antivirus does not require expertise, it does require some basic tasks for it to work properly.

– *Analysis*: the user can, at any moment, explicitly request for an analysis of the disk, an external drive or a file to specifically detect whether it contains a virus.

– *Updates*: the user must periodically download updates in order for new viruses to be taken into account. A delay of one day can mean the absence of protection from many viruses. When the antivirus is being installed, an update must be performed because it will only account for the viruses known up to the date it was put on the market. If it appeared one year ago, it will not provide protection against several thousand viruses.

5.2.4. *Antivirus components*

An antivirus is made up of four parts: a scanner, a monitor, a viral base and a log.

– *Scanner*: it analyzes files on demand in a designated area of a computer. It must be efficient, and given its complexity, it uses a large amount of resources. This is an exhaustive analysis responding to an explicit request by the user. The analysis encompasses files, saves, the register log and, in some cases, the memory content, in order to detect and destroy memory-resident viruses.

– *Monitor*: it analyzes files briefly as they are accessed. It must be quick and uses the minimum amount of resources in order to stop viruses immediately without blocking the computer. It covers four principal activities: email, the Web, downloading and the system.

– *Viral base*: this is a database composed of groups of signatures for different known viruses. Periodic updates are needed to expand the database with new viruses that have just been developed. The updates can be done either online with an Internet connection or offline by executing the appropriate patch.

– *Log*: this is composed of different records used to save the history, activities and actions taken by the antivirus.

5.2.5. *Antivirus and firewall comparison*

An antivirus, which is a specific solution, can be compared with a firewall, which is a general solution, as presented in Table 5.1.

Security tool / Characteristics	Antivirus	Firewall
Nature	Specific	General
Form	Software	Hardware or software
Degree of expertise needed	Minimum knowledge	Advanced administrator experience
Functions	Detection, isolation and destruction of viruses	Filtering and tracking (network or applications) or NAT
Degree of complexity	Simple	Complex
Necessary periodic actions	Periodic updates (preferably automatic)	Configuration and analysis of the log
Use	Widespread, domestic	Modest, professional

Table 5.1. *Comparison between an antivirus and a firewall*

5.3. Intrusion detection systems

An intrusion detection system (IDS) is a mechanism meant to identify abnormal or suspect activities in the analyzed object (a network or a host). It thus provides knowledge of successful intrusions as well as failed attempts. IDSs, which have gone through considerable evolutions, include many forms and can act at many levels.

5.3.1. *IDS purposes*

An IDS ensures the identification of all intrusive behavior in an environment and promptly provides a notification of this behavior.

An IDS is a monitoring system based on varied and complementary techniques. It must announce any detections it makes in order to be able to react at the right moment and find the necessary remedy or prevention measures so that the attack can be evaded or deactivated.

5.3.2. *IDS components and functions*

An IDS is made up of many elements:

– *Production stack*: this consists of a group of integrated TCP/IP modules in the majority of network operations on the system platform. It detects IPv4 and IPv6 intrusions and extrusions.

– *Service stack*: this consists of a group of integrated modules in the service and support of the system platform. It detects IPv4 intrusions and extrusions and is the first response to the production stack.

– *IDS task*: this is the component that treats events generated by the production stack and the service stack.

– *Graphic interface*: this allows the events to be visualized from an audit record.

– *IDS notification*: this is the alert component that sends messages via email.

– *Audit record*: this maintains the log.

IDS components and the environment in which they operate are presented in Figure 5.1.

The IDS works in the following manner:

– When the production stack or the service stack detects an advanced intrusion or an extrusion, it sends an event to the IDS task.

– The IDS task takes events in order one by one and ensures that each event corresponds to one condition (from the port table). The IDS task also maintains statistics on intrusion and extrusion events.

– The IDS gives notifications of events that go beyond the thresholds set in the strategy files.

– If there is a notification of an event, an intrusion surveillance record is created in the audit log.

– The IDS graphic presents intrusion events from the audit log folders.

– If an email and a notification message have been configured in the IDS properties page, the IDS notification will send an email to the specified address and a message to a message file.

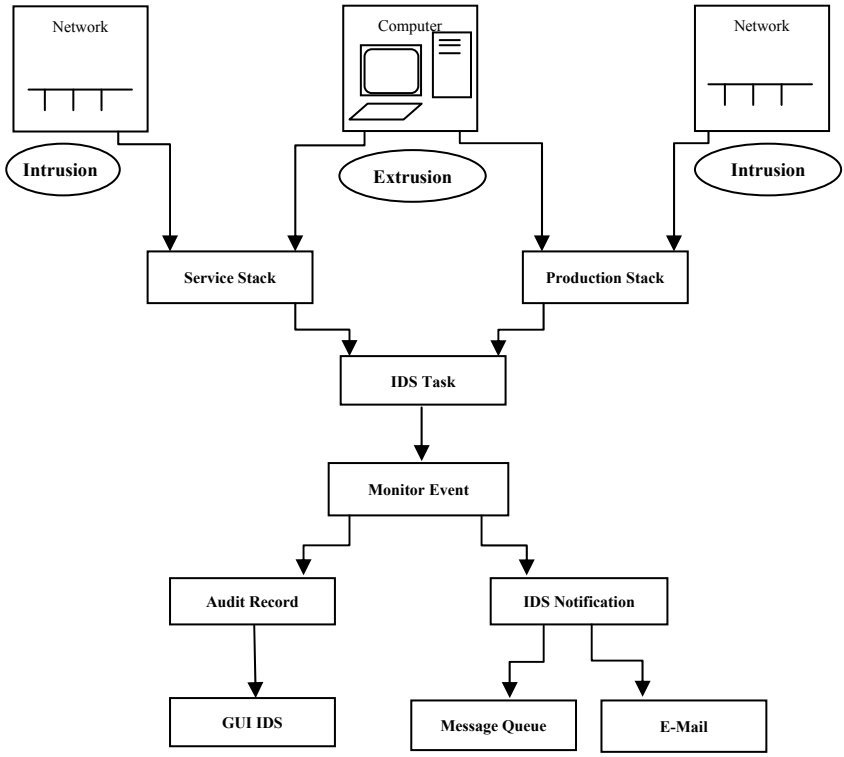

Figure 5.1. *IDS components and functions*

5.3.3. *IDS classification*

IDSs can be classified according to either the field of action (computer or network) or the level of protection (simple detection or prevention).

5.3.3.1. *Classification by field of action*

Depending on what needs to be secured, a computer or a network, we distinguish an H-IDS (Host-based IDS) and an N-IDS (Network-based IDS).

H-IDS

H-IDSs, or computer intrusion detection systems, are IDSs dedicated to monitoring equipment and operating systems. An H-IDS gathers information provided to it by the equipment or the operating system. For this purpose, there are different approaches: signatures and behavior.

An H-IDS behaves like a daemon or a standard service on a system host that detects suspicious activity by comparing it to a norm. If the activity is very different from the norm, an alert is generated. The computer can be monitored in various ways:

– *computer activity*: number and lists of processes, users, resources used, etc.;

– *user activity*: times and duration of connections, commands used, messages sent, programs started, usage beyond a predefined perimeter, etc.;

– *malicious activity* by a worm, virus or Trojan horse.

Another type of H-IDS seeks out intrusions in the system kernel and any modifications made therein. This technique is called protocol analysis, a technique that does not require searching a signature database.

N-IDS

N-IDSs, or network-based intrusion detection systems, are IDSs dedicated to monitoring the security state of a network. N-IDS analysis can be divided into three general subsections: capture, signatures and alerts.

– *Capture*: this captures network traffic in real time. Most N-IDSs use a standard capture packet library called libpcap, which is available on most platforms.

– *Signatures*: this is a library of attack signatures (classified by context) used in a manner similar to an antivirus. Thus, an N-IDS is efficient when it is familiar with the attack, but inefficient when it is not. Commercial and free tools have evolved to provide signature personalization, which allows attacks of which only one part of the elements is known to be dealt with. Tools based on signatures require more regular updates.

– *Alerts*: these are generally saved in the form of a log. A norm exists for organizing the content so that different security elements can be altered. This

format is called the IDMEF (Intrusion Detection Message Exchange Format), described in the RFC4765. The IDMEF offers an infrastructure so that the IDS does not have to send alerts. This allows the IDS to focus on describing the information it contains without having to record it for later user visualization.

The N-IDS has the advantage of being a system in real time, which means attacks targeting numerous computers at once can be discovered. The disadvantage is the high rate of false positives generated.

5.3.3.2. *Classification by level of protection*

Intrusion management systems can provide detection and/or prevention. There are three families of systems: IDS, IPS and IDS/IPS.

IDS

This is the simplest form insofar as it only provides detection. The mechanism identifies an abnormal or suspicious activity on the analyzed object (a network or host) and thus provides the knowledge of intrusion attempts on a company.

IPS

An IPS, or intrusion prevention system, is a mechanism that protects systems from intrusions. IT is evolved from the IDS and provides both detection and prevention. It has the same detection role as an IDS except that the system can take steps to diminish the risk of impact from an attack. The IPS is an active IDS that detects automatic scanning, and can automatically block ports.

IDS/IPS

This is a new generation of IDS/IPS, which are hybrid detection/prevention solutions for intrusions in a computer and network.

5.3.3.3. *Comparison between different types of IDS/IPS*

Table 5.2 presents a comparison between the H-IPS and the N-IPS by presenting their advantages and disadvantages.

	Advantages	Disadvantages
Host-based IPS	– Provides specific protection for the computer – Offers high-level protection for the operating system and applications – Protects the computer after messages are decrypted	– Depends on the operating system – Requires installation on all computers
Network-based IPS	– Provides better cost – Is independent from the operating system	– Is incapable of analyzing encrypted traffic – Must stop malicious traffic before it reaches computers

Table 5.2. *Advantages and disadvantages of the H-IPS/N-IPS*

5.3.4. *Examples of IDS/IPS*

In this section, we will present some examples of the IDS or IPS that can offer open-source or commercial solutions. Each of its advantages, weaknesses and classifications will be explained.

5.3.4.1. *Snort*

Snort is a free IDS/IPS[1]. It offers several modes:

– *Listening mode*: this starts Snort in a sniffer mode so the packets identified by an IDS can be observed.

– *Packet log mode*: this archives packets in circulation on the IDS network. It essentially provides an address book thanks to its log of actions of interest, which allows logs to be limited to certain criteria.

– *Intrusion detection mode*: the IDS mode allows Snort to take a particular action when a succession of character frames is detected in intercepted packets, following the rules defined in the configuration files.

Alerts provided by Snort can be of different kinds. For example, we can ask Snort to redirect all alarms regarding standard outgoing traffic in order to observe the evolution of attacks.

1 https://www.snort.org

Snort is also capable of adopting behavior with the aim of denying access to a certain IP address, for example, when this IP address tries to enter a network. In such cases, the IDS can interact with the firewall to update its access rules and prevent any contact with a potential hacker.

5.3.4.2. *Cisco IOS IPS*

Setting up this kind of IPS requires the following steps:

– Find the appropriate IPS on the Cisco website. Download IOS IPS files.

– Create an IOS IPS configuration directory in Flash.

```
R#mkdir IPSDIR
```

– Configure an IOS IPS encryption key.

```
R#crypto key pubkey-chain rsa
```

– Activate IOS IPS.

```
R(config)#ip ips name IOSIPS
R(config)#ip ips name IOSIPS list <ACL-relevant-flux>
R(config)#ip ips config location flash:IPSDIR
R(config)#ip ips notify sdee
R(config)#ip ips notify log
R(config)#ip ips signature-category
R(config-ips-category)#category all
R(config-ips-category-action)#retired true
R(config-ips-category-action)#exit
R(config-ips-category)#category ios_ips basic | advanced
R(config-ips-category-action)#retired false
R(config-ips-category-action)#end
R(config)#interface g0/0
R(config-if)#ip ips IOSIPS in | out
```

– Load the IOS IPS signature package onto the router. Download and copy the signature package from the appropriate FTP server.

```
R#copy ftp://ftp-user:password@server_IP_adress/sign
_package idconf
```

5.4. Conclusion

Antiviruses, though they are a specific solution, are essential tools for securing computer systems. Currently, modern security solutions allow many security tools to be grouped within one application, for example, antivirus, antispam, antispyware, antimalware, etc. as well as firewall functions, namely filtering and tracing.

IDSs, however complex, are a necessary solution for ensuring the security of a computer or network system. Combined with other tools, they reduce the effect of attacks.

IDSs are generally neglected by users who confuse them with antiviruses, even though the two actually offer complementary services. IDSs are more evolved and more complex and the need to use them in critical systems is inevitable, although they remain optional for domestic use.

6

Techniques and Tools for Encryption, IPSec and VPN

6.1. Introduction

Encryption is a very old technique used to ensure the confidentiality of information both locally and through a network. Since it appeared, it has undergone many evolutions. Nevertheless, this technique does not escape every attempt at infiltration by hackers. Its importance is measured by the time needed to crack the code.

Encryption, a branch of mathematics, is not linked to computer science. It has been used for several centuries, either strictly in the context of war or during the transmission of critical information. Historically, many events have occurred and many plans have been foiled simply because a third party decoded an encryption, as was the case in World War II, when defeats were caused by the interception of the encryption mechanism.

Currently, encryption has become a necessity and continues to play a role in all areas and fields of activity. It has become a necessary and inevitable measure in the process of securing any activity.

Numerous encryption techniques have been developed and affect the TCP/IP inference model in many ways; for example, SSL in application layers and IPSec in network and transport layers. These techniques provide authentication and/or the confidentiality of IP packets, or their content only.

Computer Science Security: Concepts and Tools,
First Edition. Ameur Salem Zaidoun.

VPNs are a very widespread communication solution because they offer two advantages: limited cost and strong security. The use of VPNs continues to spread as an alternative solution for specialized connections, and especially for communications that do not require a large bandwidth. An example of this would be the interconnection of different agencies within one company with multiple sites in different locations, working from home or any other company access via the Internet.

VPNs are a tunnel mode application of IPSec or any other tunneling protocol, which allows encrypted information that cannot be decrypted by a third party to be sent over the infrastructure of a public network.

6.2. Encryption techniques

With the appearance of computing and the proliferation of network use, notable for the easy access to information it provides, it is indeed possible to access information saved on computers, as well as information that gets sent over the network without any hindrance. Cryptography has been used for centuries to send critical information between leaders and individuals, especially in circumstances of war.

Encryption is a security mechanism of capital importance. It ensures confidentiality and prevents passive attacks.

It is summarized in Figure 6.1.

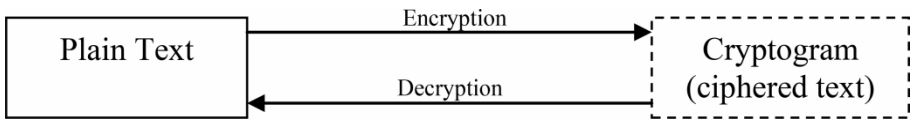

Figure 6.1. *Principles of encryption/decryption*

– Encryption is the act of obtaining a ciphered text from a plain text by using the necessary tools. It is the modification of a plain text, making it incomprehensible to a third party.

– Decryption is the inverse operation, which is performed using the necessary tools.

6.2.1. *Basic principles of encryption*

Encryption consists of finding ways to obtain a cryptogram based on a text that is to be secured. It is a reversible action so that the corresponding plain text can be obtained during the decryption process.

Encryption is based on two, basic, complementary techniques; they can be applied numerous times one after another to make the encryption stronger.

6.2.1.1. *Substitution*

This is a primitive technique of manual mono-alphabetic substitution. It associates each character with a different character. The two parties agree on this correspondence, which needs to be formulated and memorized by both parties.

EXAMPLE.–

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Q	A	W	X	S	Z	E	D	C	V	F	R	T	G	B	N	H	Y	U	J	K	I	O	L	M	P

“COMPUTERSECURITY” → “WBTNKJSYUSWKYCJM”

6.2.1.2. *Shifts*

This involves associating each character with a corresponding character according to an interval of a chosen number, which is called the step. We call this Caesar’s cipher, an ancient encryption technique using mono-alphabetic substitution. The step is a sort of key, since the knowledge of this number allows the cryptogram to be entirely determined during encryption, and inversely, the corresponding plain text during decryption.

EXAMPLE.– Interval of 2

“COMPUTERSECURITY” → “EQORWVGTUGEWTKVA”

In addition to these two basic techniques, other multi-alphabetic substitution techniques make encryption stronger. The plain text can undergo different transformations in advance or throughout the encryption process.

6.2.2. *Cryptoanalysis*

Encryption, however strong it may be, is not above all criticism insofar as it can be compromised and the hacker can obtain the plain text using an encrypted text, without knowing anything about the encryption technique used. Turing's efforts during World War II are a good illustration of this.

Cryptoanalysis is a technique used by hackers to break encryption. It consists of analyzing an encrypted text with the purpose of obtaining the corresponding plain text.

6.2.2.1. *Mechanisms of cryptoanalysis*

Cryptoanalysis uses many mechanisms based on linguistic statistics (e.g. "ing" is repeated in 4% of an English text, "tion" is repeated in 2% of a French text, etc.), or on sentences and expressions that are commonly used in messages. Currently, given the power of computers, they can be used in systematic trial and error.

A hacker can proceed in numerous ways:

– extract hypotheses about an encrypted text, for which the corresponding plain text is at hand to decode the encryption of another targeted text encrypted in the same way;

– encrypt a particular text in order to determine certain characteristics of the encryption technique in question;

– examine the repetitions of letters in the cryptogram in order to find the key. This is frequential analysis, an attack that targets the encryption via mono-alphabetic substitution;

– calculate the probability of repetitions of letters in the cryptogram. This is analysis by coincidence;

– test all of the words in a list as a key. This is dictionary analysis or brute force hacking;

– make a linear approximation of the internal structure of the encryption method. This is linear cryptoanalysis;

– slightly modify the plain text of a text for encryption and statistically analyze the structure of the encryption method. This is differential cryptoanalysis;

– combine the last two methods for differential-linear cryptoanalysis.

6.2.2.2. *Measuring encryption robustness*

Cryptoanalysis theoretically always manages to find the plain text of an encrypted text, given sufficient time and material. Any information has a limited life span of confidentiality. The robustness of an encryption algorithm is measured by the time needed for cryptoanalysis, which must be well above the duration of validity of the information (the time during which the information must remain confidential).

Political or even military information has a theoretical duration of validity, and because of this, a need arose for an encryption technique for which the theoretically necessary duration of cryptoanalysis surpasses the duration of validity for the information in question, making the exposure of any encrypted message pointless.

6.2.3. *Evolution of cryptography*

The evolution of computing and transmission techniques created a new challenge: confidentiality. Encryption appears to be the solution to this challenge. The latter has always undergone evolutions throughout its history. In the following section, we will examine some of the most important generations.

6.2.3.1. *Shared secret algorithm*

This generation was among the first encryption solutions to have been developed. A good illustration of this is to make a table of correspondence between characters and their replacement. This table must remain hidden from others to ensure confidentiality.

Principle

It consists of choosing an algorithm, as presented in Figure 6.2, shared between two or more people and which remains a secret among the relevant parties. This algorithm is used for both decryption and encryption.

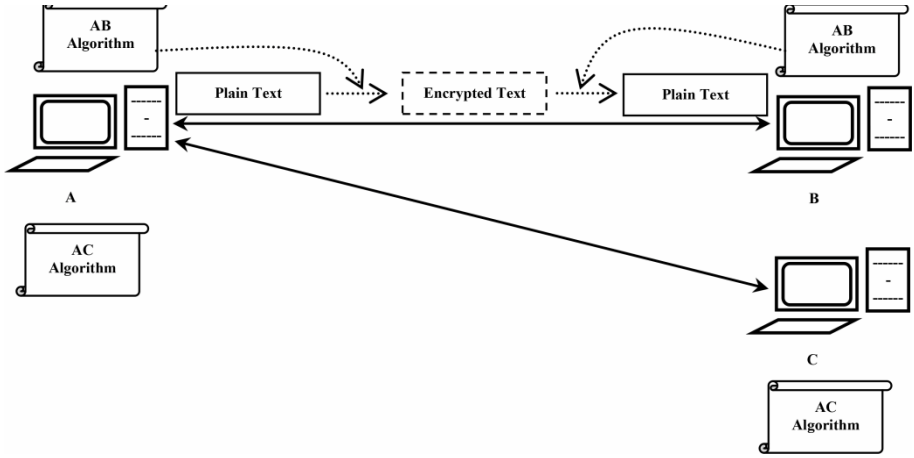

Figure 6.2. *Encryption by shared secret algorithm*

Limits

This encryption solution contains a certain number of oversights:

– the difficulty of choosing an algorithm and evaluating its strength;

– the problem of sharing the algorithm, something which is complex and thus difficult to memorize, and that has a long lifetime, which facilitates its disclosure over time.

6.2.3.2. *Shared symmetrical key algorithm*

Given the limits identified regarding the previous generation, this generation has appeared in response. The idea is to have the key be a hash code associated with a password chosen by the user. This approach is a revolution in the field of cryptography, and numerous algorithms of this type have now been developed.

Principle

The problems inherent to the use of the preceding method favored the appearance of this technique, which consists of using known and carefully selected algorithms and connecting each use of this algorithm to a key that is secret and shared by both users (e.g. the step in the Caesar cipher). It is used for both the encryption and decryption (symmetry), as presented in

Figure 6.3. Examples of this kind of algorithm include DES, 3DES and Blowfish.

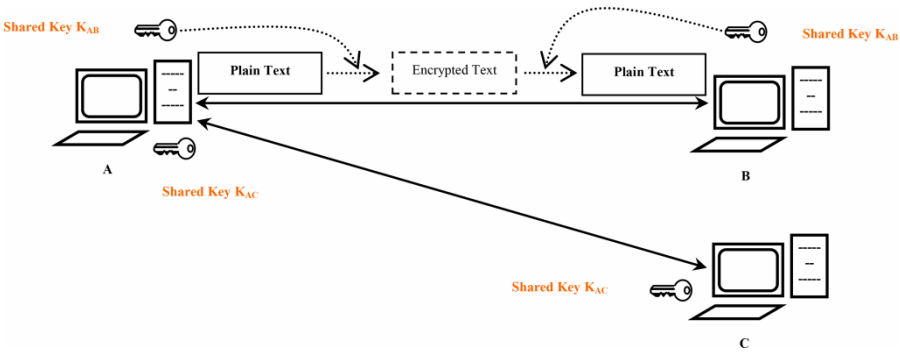

Figure 6.3. *Encryption by shared symmetrical key. For a color version of this figure, see www.iste.co.uk/zaidoun/computer.zip*

Examples

– DES: Data Encryption Standard, since 1977, uses a key with 56 bits.

– 3DES: a variant of DES with three repetitions of three different keys (A-B-C 168 bits) or with two different keys (A-B-A 112 bits).

– RC2, RC4, RC5: these use keys up to 1,024 bits.

– Blowfish.

– IDEA: International Data Encryption Algorithm.

– AES: Advanced Encryption Standard, since 2001.

Limits

This technique also has limits, but these are less extreme than the first:

– the problem of sharing the key, which facilitates its discovery and demands it be periodically modified;

– there are as many keys as there are communications for a given entity, which creates the supplementary task of managing keys.

6.2.3.3. *Asymmetrical key algorithms*

In the two previous generations, the major problem was that of sharing (the algorithm or the key). Indeed, sharing something secret creates a paradoxical scenario, given that sharing facilitates disclosure or sows doubt between two parties with respect to one another. It became necessary to find a solution that completely eliminated the need to share; this was done with algorithms whose use was based on the choice of a pair of keys, rather than just one key.

Principle

This technique was developed to resolve the problem of sharing, identified for previously used techniques, as a follow-up to the new generation of algorithms. A pair of keys (k1 and k2) is associated with each use, such that if one key is used to encrypt, only the second key can decrypt, and vice versa: ($D_{k1}(E_{k2}(X)) = X$ and $D_{k2}(E_{k1}(X)) = X$).

The idea consists of giving each user a pair of keys, one of which is designated as the secret, private key and the other is designated as the public key for everyone, as shown in Figure 6.4.

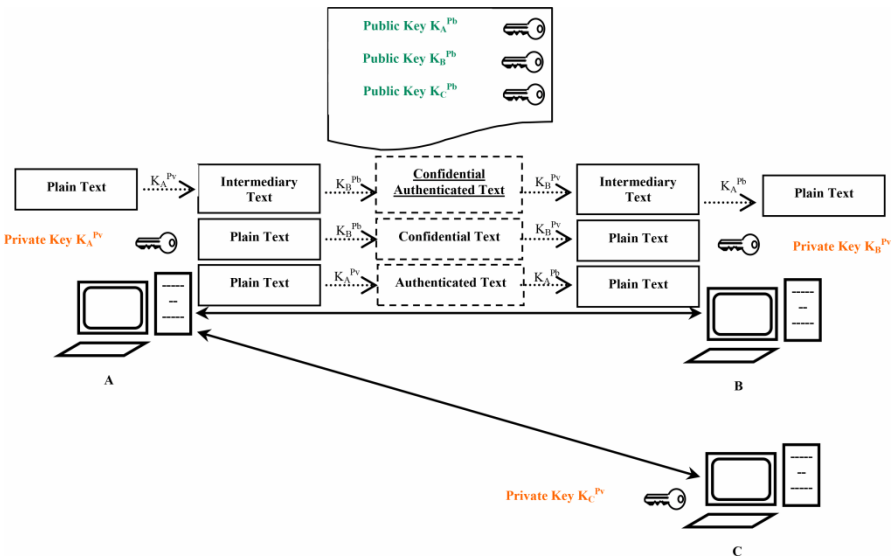

Figure 6.4. *Encryption by asymmetrical keys. For a color version of this figure, see www.iste.co.uk/zaidoun/computer.zip*

Examples

– RSA: Rivest–Shamir–Adleman. This algorithm was described in 1977 by Ronald Rivest, Adi Shamir and Leonard Adleman. It is based on the use of a pair of keys made up of one public key for encrypting (respectively, verified) and one private key for decrypting (respectively, signed) confidential data.

– DSA: Digital Signature Algorithm. This is a signature algorithm with a public key. The private key is used to generate a digital value called a signature; the public key is used to verify this signature.

Limits

This solution is complex, even though it avoids the problem of sharing. Moreover, the issue of trust regarding the public key is raised.

6.2.4. *The concept of certificates*

Certificates offer a solution to the problem of key distribution. A certificate is a digital document that provides assurance that a public key is associated with the relevant person or entity. It includes the following information:

– the public key;

– the owner's name;

– the expiration date of the key;

– the name of the certification guarantor;

– the series number of the certificate.

This information is signed by a trusted authority called the certification authority; it provides an imprint (the hash code related to the set of information above) that is encrypted by the private key of the certification authority, as presented in Figure 6.5. This is the case for x509 certificates.

The verification depicted in Figure 6.6 is done by comparing the hash code for the information to the chain obtained by deciphering the imprint using the public key of the certification authority.

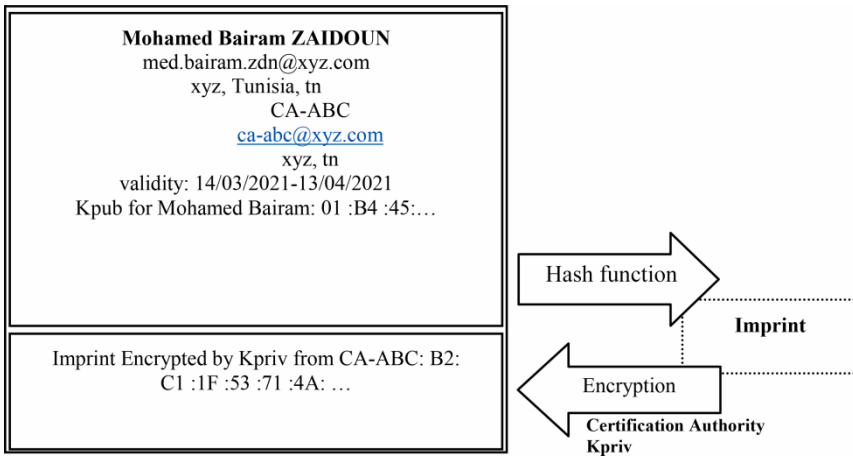

Figure 6.5. *x509 Certificate signed by the certification authority*

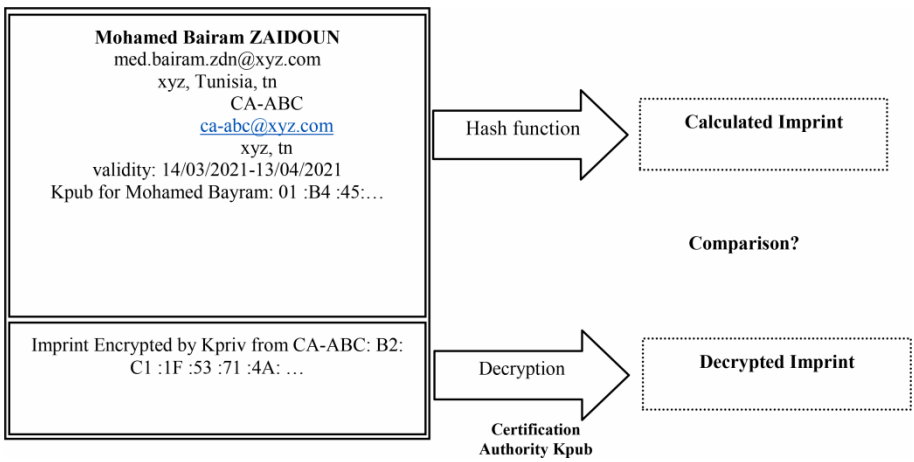

Figure 6.6. *Verification of the x509 certificate validity*

6.2.5. *Comparison of encryption techniques*

The three generations of encryption solutions offer many points of overlap and many differences in terms of principles, usage and fields of application. A comparison of these techniques is presented in Table 6.1.

Encryption technique / Characteristic	Shared algorithm	Symmetrical shared key	Asymmetrical key
Definition of Characteristics	Algorithm per communication	Key per communication	Pair of keys per participant
Problem of sharing	Yes (algorithm)	Yes (key)	No
Coupling of authentication/confidentiality	Yes	Yes	No
Management of keys	No	Difficult (as many keys as communications)	Simple (publication of public keys)
Complexity	Complexity of choosing an algorithm and evaluating it	Moderate	Complex use

Table 6.1. *Comparison of encryption techniques*

6.3. IPSec

IPSec provides authentication of IP packets via the AH protocol and confidentiality of data via the ESP protocol.

The IPSec protocol is applicable in two modes, the transport mode and the tunnel mode. The first is the simpler mode that only concerns the data, while the second secures the entire IP packet (header and data). We speak of IP in IP encapsulation. The tunnel mode is used, among other things, as a basic technique for implementing VPNs.

The family of IPSec protocols includes two complementary techniques, AH and ESP. These manage both the implementation of numerous encryption algorithms (DES, 3DES, Blowfish, RSA, etc.) and hashing (MD5, SHA1, etc.). The configuration of an IPSec communication determines the appropriate choice of algorithms to be used. IPSec was defined by the IETF as a chain of 40 RFCs that detail all aspects of the protocol. Moreover, IPSec is automatically integrated with IPv6.

IPSec uses the notion of a security association (SA) to define usage and the security parameters to apply for different parties.

An IPSec security association is a structure of data whose function is to record all of the security parameters associated with a communication. Since an SA is unidirectional, two SAs are needed to protect a communication in both directions.

6.3.1. *AH*

AH, or Authentication Header, is a protocol that provides authentication and guarantees the integrity of IP packets. It is applicable in transport mode and tunnel mode.

The AH protocol offers the following security services:

– integrity in non-connected mode;

– authentication of data;

– anti-replay (optional).

6.3.2. *ESP*

ESP, or Encapsulating Security Payload, is an IPSec protocol that provides confidentiality and can offer authentication, ensure the integrity of data and the anti-repudiation of sessions. It is applicable in transport mode and tunnel mode.

The ESP protocol offers the following security services:

– confidentiality;

– protection against traffic analysis;

– integrity in non-connected mode (like AH);

– authentication of data (like AH);

– anti-replay (like AH).

ESP indeed covers the services offered by AH.

6.3.3. *Different IPSec modes*

Security can involve only the IP packet data or it can apply to the entirety of the packet (header and data). We can distinguish two application modes for IPSec, transport and tunnel mode.

6.3.3.1. *Transport mode*

This ensures the security of the IP packet data. Transport mode is the default mode of IPSec and is used for end-to-end communications (communications between a client and a server). When transport mode is used, IPSec only encrypts the IP payload. Transport mode provides protection for the IP payload via an AH or ESP header.

6.3.3.2. *Tunnel mode*

This offers security for the IP packet (header and data). When IPSec tunnel mode is used, IPSec encrypts the IP header and payload.

Tunnel mode offers protection for the entirety of the IP packet by considering it an AH or ESP payload. In tunnel mode, a complete IP packet is encapsulated with an AH or ESP header and a supplementary IP header. The IP addresses for the exterior IP header are the endpoints of the tunnel, while those for the encapsulated IP header are the real source and destination addresses.

IPSec tunnel mode is useful for protecting traffic between different networks when the traffic must pass through an unapproved intermediary network. It is used especially for interoperability with gateway and terminal systems that do not accept IPSec connections.

Tunnel mode can be used in the following configurations:

– gateway to gateway;

– server to gateway;

– server to server.

6.3.4. *Different IPSec implementations*

Since it appeared, the family of IPSec protocols has been overhauled many times. There is a huge divergence between versions, which creates a handicap in adapting this solution, given the incompatibility.

From a practical standpoint, IPSec is a relatively difficult protocol to implement, on the one hand due to its intrinsic complexity (multiple subprotocols, etc.), and on the other hand due to its interactions with common network processes. Because of this, there are three variants of the implementations:

– *modification of the kernel IP stack*: this is the most direct method and probably the most complicated. This method is applicable to all types of entities (hosts and gateways);

– *"Bump-In-The-Stack" (BITS)*: this consists of separating routines for IPSec processing from the usual IP stack routines (the code specific to IPSec comes between the network level and the physical connection level, hence the name of the method). Nevertheless, certain elements must be modified in this stack, such as the fragmentation and reassembly of packets. Additionally, the kernel remains intact in this type of implementation. This method is applicable to all kinds of entities (hosts and gateways);

– *"Bump-In-The-Wire" (BITW)*: this consists of setting IPSec processing aside, and thus the code, in a dedicated space placed upstream from the network (hence its name), outside of the computer. This kind of location can either be a specific "box", firewall, router, etc. Depending on its type, this method will not apply to all kinds of hosts.

6.3.5. *Different IPSec encapsulations*

This section will present the different possible transformations of the IP packet during the application of each of these techniques (AH and ESP) in each mode (transport and tunnel).

6.3.5.1. *AH in transport mode*

AH in transport mode consists of authenticating the IP packet data. This task requires the addition of an AH header to the other part of the packet, while keeping the header of the original packet intact.

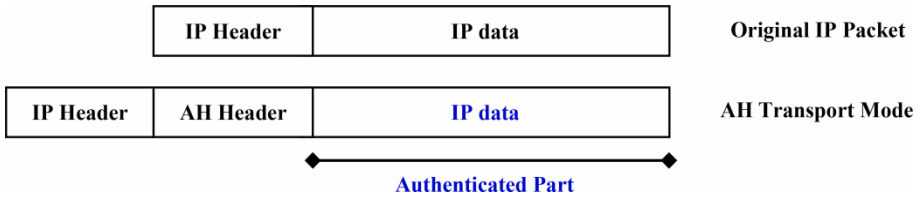

Figure 6.7. *AH encapsulation in transport mode*

6.3.5.2. *ESP in transport mode*

ESP in transport mode consists of encrypting the IP packet data. This task requires the addition of an ESP header and ESP footer to the data, while keeping the original header of the packet intact.

The resulting data, now composed of the added ESP header and the original data, will be authenticated.

The encryption and authentication also concern a part of the footer.

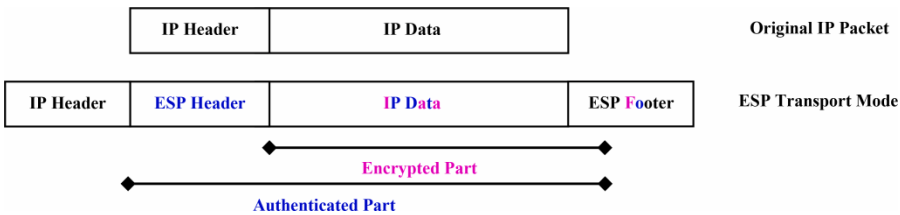

Figure 6.8. *ESP encapsulation in transport mode*

6.3.5.3. *AH-ESP in transport mode*

AH-ESP in tunnel mode consists of first encrypting the IP packet data using the ESP protocol, then applying the AH protocol and finally putting the original header of the IP packet back.

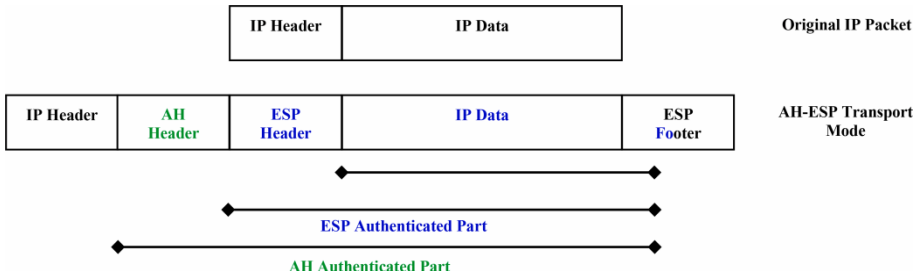

Figure 6.9. *AH-ESP encapsulation in transport mode*

6.3.5.4. *AH in tunnel mode*

AH in tunnel mode consists of authenticating the entire IP packet (header and data). This task requires the addition of an AH header and a new IP header to form a new packet with different characteristics.

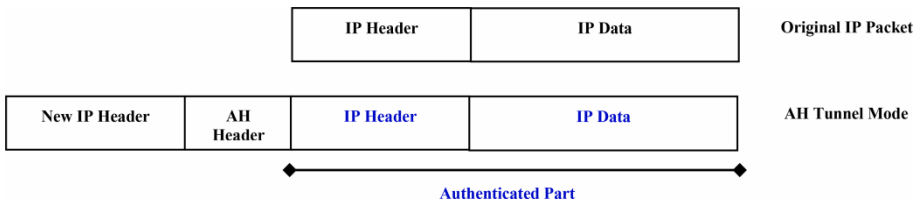

Figure 6.10. *AH encapsulation in tunnel mode*

6.3.5.5. *ESP in tunnel mode*

ESP in tunnel mode consists of encrypting the entire IP packet (header and data). This task requires the addition of an ESP header and a new IP header to create a new packet with different characteristics.

The resulting packet data made up of the added ESP header and the entire original packet gets authenticated.

The encryption and authentication also concern a part of the footer.

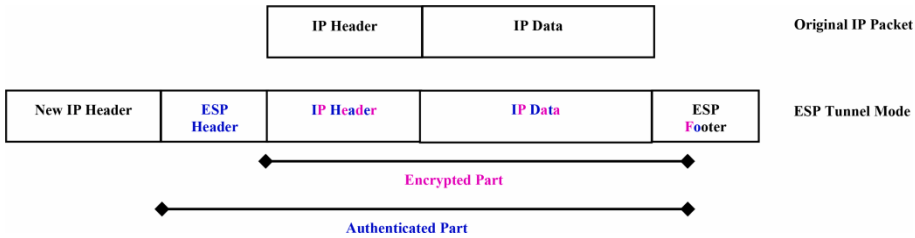

Figure 6.11. *ESP encapsulation in tunnel mode*

6.3.5.6. *AH-ESP in tunnel mode*

AH-ESP in tunnel mode consists of first encrypting the entire IP packet (header and data) using the ESP protocol and then applying the AH protocol to generate a new IP packet at the end.

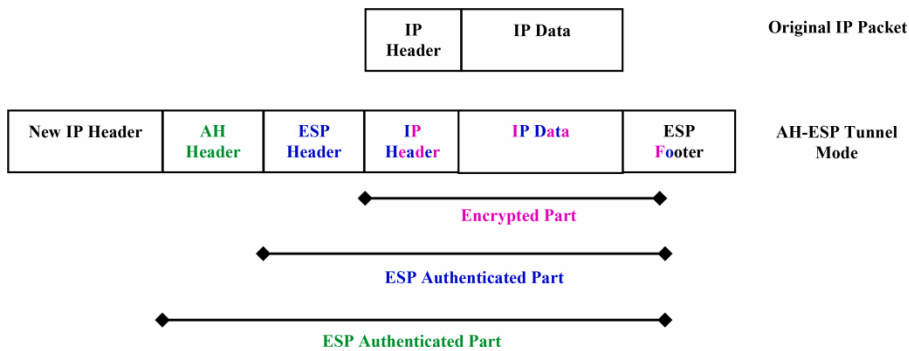

Figure 6.12. *AH-ESP encapsulation in tunnel mode*

6.3.6. *IKE protocol*

IKE is the acronym for Internet Key Exchange, a protocol for exchanging keys and negotiating security parameters. It establishes SAs (Security Associations) between partners, in this case the extremities of the tunnel.

SA is a group of unidirectional security parameters that englobe algorithms for authentication, encryption and integrity control.

SAs can be altered manually or via the ISAKMP protocol. ISAKMP (Internet Security Association and Key Management Protocol) is used for the negotiation, establishment, modification and deletion of SAs and associated parameters. It defines the procedures and format of packets for creating and managing authentication by SA pairs and the techniques for generating keys.

IKE includes two phases:

– phase 1: the ISAKMP rules for creating a tunnel and exchanging keys are established using the DH (Diffie–Hellman) protocol before verifying identities;

– phase 2: the IPSec rules for transmitting traffic in a secure fashion through the tunnel are established.

6.4. VPNs

VPN is the acronym for Virtual Private Network. This is when public resources are used to work in a secure fashion, using encryption as if it were a private network.

6.4.1. *Issues and justifications*

If someone wants to connect sites from one company (bank, insurance, etc.) in different geographical locations, two solutions are possible: a public or a private option.

The first solution consists of using the Internet to communicate. This is an inexpensive solution, but also un-secured since the network is shared with a myriad of other actors.

The second solution consists of using specialized connections. This is a completely private and therefore secure infrastructure, but it is very expensive.

Faced with this choice, VPN appears to be the optimal connection solution in terms of security and cost.

6.4.2. *VPN principles*

The establishment of a VPN consists of using tunnels to communicate via the Internet instead of using specialized connections.

A tunnel provides security for data, as well as the identity of users. We speak of IP in IP encapsulation. The new IP packet generated has the WAN addresses of local network routers communicating on the Internet via the tunnel as the source and destination.

VPN is based on a protocol, called the tunneling protocol, that is, a protocol allowing data to go from one end of the VPN to the other to be secured by cryptographic algorithms.

The term tunnel is used to symbolize the fact that between the entry and exit of the VPN, data is encrypted and thus normally incomprehensible for any person situated between the two extremities of the VPN, as if the data was passing through a tunnel. Moreover, to create a tunnel also means encapsulating a protocol within a protocol at the same OSI model level (e.g. IP in IPSec). In the case of a VPN established between two computers, we call the element encrypting the entering data the VPN client and the element decrypting the data at the exit the VPN server (or more generally, the remote server).

This is the case when a system outside of a private network (nomad client, agency, or someone working from home) wishes to connect to the company's network and:

– the packets (which contain data) are encrypted by the VPN client (using the algorithm selected by the two interlocutors when establishing a VPN tunnel) and eventually signed;

– they are transmitted via a transporter network (usually the Internet);

– they are received by the VPN server, which decrypts them, processes them and checks that the required verifications are correct.

6.4.3. *Different types of VPNs*

Based on the extremities using a VPN, there are two types: site-to-site VPN and remote access VPN.

6.4.3.1. *Site-to-site VPN*

This is a setup that connects two networks from two geographically distinct sites via the configuration of a VPN between the routers, the interface with which the Internet connection is made at both sites.

A good application of this is the connection of two or more agencies from any company, either among themselves or to the central site. This kind of VPN tunnel can also be used to connect a supplier with a client.

This kind of VPN can be an optimal solution, both secure and inexpensive, to ensure the total interconnectivity of a company with other site locations.

6.4.3.2. *Remote access VPN*

This is a secure connection via the Internet from a nomad user or home workstation to a geographically distant network. This technique offers an alternative for facilitating remote work or allowing isolated users to connect to a company without compromising security and remaining safe from attacks by hackers or network pirates.

This kind of VPN allows a company to connect its remote collaborators, who will in turn share services with the company for lower costs and with full security.

6.4.4. *Different tunneling protocols*

VPN tunnels can be based on numerous protocols:

– *PPTP (Point-to-Point Tunneling Protocol)*: a level 2 protocol developed by Microsoft, 3Com, Acend, US Robotics and ECI Telematics;

– *L2F (Layer Two Forwarding)*: a level 2 protocol developed by Cisco Systems, Nortel and Shiva;

– *L2TP (Layer Two Tunneling Protocol)*: the product of work done by the IETF (RFC 3931) to combine the functions of PPTP and L2F. This is a level 2 protocol dependent on PPP;

– *IPSec (IP Secured)*: a level 3 protocol, based on work by the IETF, transmitting encrypted data for IP networks;

– *SSL/TLS (Secure Sockets Layer/Transport Layer Security)*: it provides a very good tunneling solution. The advantage of this solution is that it uses a Web browser as the VPN client;

– *SSH (Secure Shell)*: initially considered the secure replacement for telnet, it offers the possibility of tunneling TCP connections, thus providing secure access to services offered on a protected network, without creating a virtual private network.

6.4.5. *Site-to-site IPSec VPN configuration*

Let us configure, for example, a site-to-site IPSec VPN in CLI mode (Command Line Interface) to the network configuration depicted in Figure 6.13.

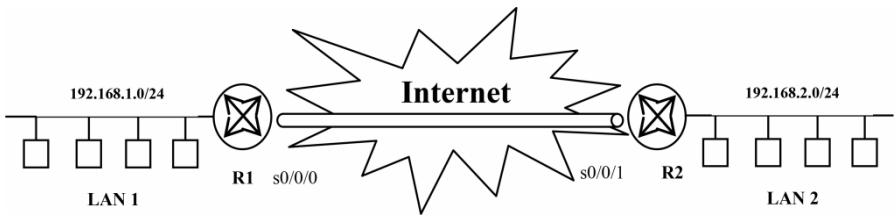

Figure 6.13. *Example of a site-to-site IPSec VPN*

The configuration of the tunnel is defined in Table 6.2.

Parameters	R1	R2
Name of transform set	VPN-SET	VPN-SET
ESP encryption configuration	esp-aes	esp-aes
ESP authentication configuration	esp-sha-hmac	esp-sha-hmac
IP addresses of tunnel extremities	10.2.2.2	10.1.1.2
Encrypted traffic	access-list 110 (source 192.168.1.0 dest 192.168.2.0)	access-list 110 (source 192.168.2.0 dest 192.168.1.0)
Crypto map name	VPN-MAP	VPN-MAP
Way of generating SA (Security Association)	ipsec-isakmp	ipsec-isakmp

Table 6.2. *Configuration parameters for an IPSec tunnel*

The configuration follows these steps:

– Update the IOS license of the different routers to make sure that the new security functions are supported:

```
R1(config)# license boot module c1900 technology-package
securityk9
R2(config)# license boot module c1900 technology-package
securityk9
```

– Accept the end user license.

– Save the configuration during execution and reset the router to activate the security license.

– Verify that the security package was activated using the command: `show version`.

– Configure the ACLs to identify traffic going through the tunnel on both routers:

```
R1(config)# access-list 110 permit ip 192.168.1.0 0.0.0.255
192.168.2.0 0.0.0.255
R2(config)# access-list 110 permit ip 192.168.2.0 0.0.0.255
192.168.1.0 0.0.0.255
```

– Configure the ISAKMP encryption rules on R1 and R2 (routers 1 and 2) using the shared encryption key, vpnpa55:

```
R1(config)# crypto isakmp policy 10
R1(config-isakmp)# encryption aes 256
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 5
R1(config-isakmp)# exit
R1(config)# crypto isakmp key vpnpa55 address 10.2.2.2
R2(config)# crypto isakmp policy 10
R2(config-isakmp)# encryption aes 256
R2(config-isakmp)# authentication pre-share
R2(config-isakmp)# group 5
R2(config-isakmp)# exit
R2(config)# crypto isakmp key vpnpa55 address 10.1.1.2
```

– Configure the IPSec IKE rules on R1 and R2 by setting up the transform-set VPN-SET to use esp-aes and esp-sha-hmac; set up the

encryption card VNP-MAP, which will connect all of the parameters in phase 1 of the IKE protocol:

```
R1(config)# crypto ipsec transform-set VPN-SET esp-aes
esp-sha-hmac
R1(config)# crypto map VPN-MAP 10 ipsec-isakmp
R1(config-crypto-map)# description VPN connection to R2
R1(config-crypto-map)# set peer 10.2.2.2
R1(config-crypto-map)# set transform-set VPN-SET
R1(config-crypto-map)# match address 110
R1(config-crypto-map)# exit
R2(config)# crypto ipsec transform-set VPN-SET esp-aes
esp-sha-hmac
R2(config)# crypto map VPN-MAP 10 ipsec-isakmp
R2(config-crypto-map)# description VPN connection to R1
R2(config-crypto-map)# set peer 10.1.1.2
R2(config-crypto-map)# set transform-set VPN-SET
R2(config-crypto-map)# match address 110
R2(config-crypto-map)# exit
```

– Link the encryption mapping VPN-MAP to the series interface at each end of the tunnel for each router:

```
R1(config)# interface s0/0/0
R1(config-if)# crypto map VPN-MAP
R2(config)# interface s0/0/1
R2(config-if)# crypto map VPN-MAP
```

– Verify the tunnel on each end for both routers:

```
R1#show crypto ipsec sa
R2#show crypto ipsec sa
```

6.5. Conclusion

Encryption is a very efficient technique that ensures confidentiality and/or authentication of local information or messages exchanged by network applications. Such a solution is insufficient without low-level encryption. That is the purpose of the IPSec protocol, which is a version of the TCP/IP that offers confidentiality and/or authentication at the packet level.

The appearance of IPSec was without a doubt a radical solution for mitigating the faults and vulnerabilities of TCP/IP. Its use remains timid, given the diversity and incompatibility of implementations.

IPSec applies encryption techniques at the level of the packet to create a secure protocol stack that can be used at high levels to create VPNs.

VPNs use the simple principle of the tunnel, which is a secure encapsulation by encryption. It is based on a large panoply of open and proprietary technologies. The richness of this diversification offers a greater panoply of choices for establishing VPN tunnels. It offers flexible access to company data and extends security to the supplier, client, home worker, etc. This technique, coupled with virtualization techniques such as “cloud computing”, makes access commonplace, all the while ensuring security.

7

New Challenges and Trends in Security, SDN and IoT

7.1. Introduction

Nowadays, the classical architecture of networks is being left behind, to the detriment of new technologies, among which we can mention the SDN (software-defined network). The new technology has different characteristics and architectures that can cause different attacks compared to traditional networks. This scenario presents specific challenges that require appropriate security solutions.

The SDN architecture was developed with the aim of making a network virtual. It makes the control panel virtual by moving it from each peripheral device to a central network entity for control and rule generation, called the SDN controller.

Indeed, smart objects are beginning to dominate all areas, connected to one another via a network and evolving increasingly towards intelligence and automation. This considerable evolution presents limitations with respect to security challenges at every level. These specific challenges demand appropriate security solutions. This kind of network is a potential target for intrusion insofar as the connected smart objects and the automation they offer provide pirates with key control and spy points. These vulnerabilities make security especially important in the IoT/IoE.

Computer Science Security: Concepts and Tools,
First Edition. Ameur Salem Zaidoun.

7.2. SDN security

Networks are progressively migrating towards automation and virtualization, which opens the door to a new trend in security that is specific to SDNs.

7.2.1. *General description of an SDN*

SDN technology is a network architecture approach that allows data (useful traffic) to be separated from control (network management traffic). The SDN architecture is made up of three layers, separated by interfaces through which control and management information is sent.

SDNs have specific functions compared to traditional networks. They offer a centralized architecture in the sense that network traffic is managed by a central device called the SDN controller. It communicates with the infrastructure layer containing different network components via a communication interface called Southbound. Moreover, it communicates with the management layer containing administrative and supervisory devices and applications via a communication interface called Northbound.

The SDN can be managed by scripts and programs written in advanced programming languages such as Java, Python, C and P4.

Two functions, namely centralization and programmability, make the SDN a flexible network that is easy to manage using virtualization mechanisms and tools.

These functions mean that the SDN incites different attacks, presents different vulnerabilities and offers different security solutions. Security challenges are very different from those in traditional networks.

The SDN can be managed in collaboration with a cloud computing solution, which is used to store rules and policies. In cloud-based security, the SDN controller uses the security actions of cloud services to create traffic rules in order to apply security actions.

7.2.2. *SDN architecture*

The architecture of an SDN is made up of three hierarchical layers, as shown in Figure 7.1.

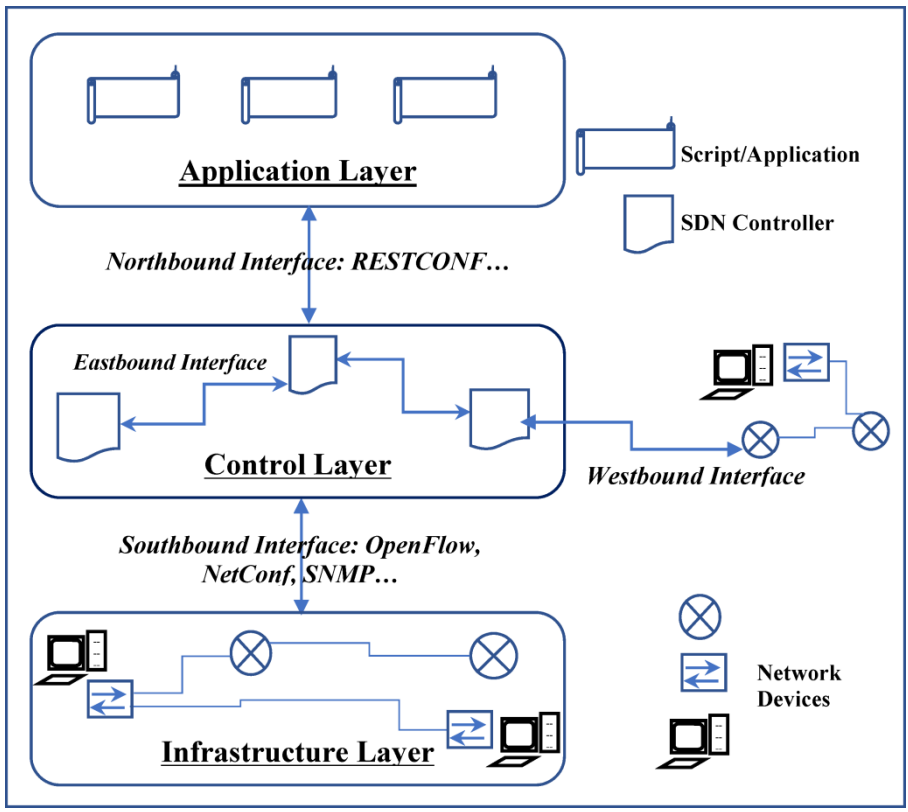

Figure 7.1. *SDN architecture*

The functions and components of the layers that define the architecture of an SDN are as follows.

– *Infrastructure or data level*: this is the lower layer, which is made up of equipment and network nodes such as the switch/router, workstations and sensors. This is also called the transmission or data plane, and this layer refers to the switching that connects different network ports on a peripheral. The data plane for each peripheral device transmits and advances traffic.

– *Control level*: this is the middle layer, which contains the most important component, the SDN controller. This is also called the control plane, which is considered the brain of the system. The control plane is the location where transmission decisions are made. Information sent to the control plane is processed by network nodes.

– *Application level*: this is the top layer, which is optional. It includes advanced equipment that control applications along with management and administrative scripts.

This architecture includes four different types of interfaces.

– *Southbound*: this is used by the controller to communicate with the data layer.

– *Northbound*: this is used by the controller to communicate with the application layer.

– *Eastbound*: this is used by the controller to communicate with other controllers in the same SDN.

– *Westbound*: this is used by the controller to communicate with equipment in another traditional network.

In order to program networks and make them virtual, various options are available:

– *protocols*: OpenFlow, NETCONF, RESTCONF, etc.;

– *programming languages*: C, C++, Java, etc.;

– *script languages*: Python, Lua, etc.;

– *data modeling language*: Yang;

– *markup language*: XML.

7.2.3. *SDN components*

The components of an SDN are divided into three layers, making up its architecture.

The most important component is the SDN controller, which is part of the control layer. Other components are further divided into two layers, namely

the nodes in the data layer and the advanced components in the application layer, which is the management layer.

7.2.3.1. *Network nodes*

These components only provide the function of transmitting data, regardless of the routing function or the virtual control provided by the SDN controller.

Packets are transmitted directly into the data layer based on the information contained in the FIB (forwarding information base) and the adjacency table, without having to resort to the control layer, which is in charge of updating the FIB in deferred time.

7.2.3.2. *SDN controller*

This is a device in the SDN control layer that serves as a programmable automation point for managing, configuring and repairing physical and virtual network infrastructures. Thus, the centralized controller manages data flow, reinforces security and provides other services so that each node can be in charge of only transmitting data.

The SDN controller, in which the intelligence of the network is centralized, is the brain of the SDN. One or more controllers can be configured. With just one controller, we speak of control being physically centralized, whereas with numerous controllers, control is physically distributed and, in this case, the controllers can function together (centralized software) or independently of each other and those with distributed software. It is also possible to have hybrid SDN controllers.

An SDN controller can be configured differently, for example by using the web mode, CLI (command line interface) mode or an advanced programming language such as Python.

7.2.3.3. *High-level components*

These components are necessary for executing scripts and programs that control the SDN controller so that the configuration can be updated and new security and optimization functions can be deployed.

7.2.4. *Security issues in SDNs*

Because of its distinct architecture, the SDN faces different security challenges. It is affected by a variety of specific attacks.

7.2.4.1. *Security attacks in SDNs*

The SDN can be the target of sniffing by hackers who exploit gathered information to launch access attacks such as DoS, called black hole attacks, or to use malicious codes, called wormhole attacks. The Sybil attack is triggered by adding entries to affect network traffic.

Other attacks mainly exist on IoT platforms, such as jamming attacks caused by frequency interference and resource exhaustion attacks caused by excessive energy consumption.

In another situation, an attacker can appear as a fraudulent neighbor who can trigger a malicious hello flood attack.

7.2.4.2. *Security challenges in SDNs*

Given that the security issues identified for an SDN are different from those of the traditional network, the security challenges are not the same.

First, data flow in the SDN is greater than in a traditional network. The control data is sent by SDN controllers in order to manage communication. If this data is sent in plain text via the network, it can be captured by hackers. This vulnerability is a big threat and poses a serious challenge.

Second, it is necessary to provide an access control solution to filter the flow of data passing through different interfaces.

Third, DoS and DDoS attacks must be attenuated to ensure SDN availability.

Fourth, access must be limited by using appropriate security rules and access control systems.

Finally, SDN controllers require security measures to provide adaptability, coherence and dependency. The placement and synchronization of the latter constitute a very important challenge.

7.2.5. *Security solutions for SDNs*

Since this type of network is defined by software, security solutions generally take the form of software, which we call software-defined security (SDSec). Security solutions must be mainly based on the SDN controller, the brain of the setup, which must be highly secured via firewalls and IDS/IPS to attenuate the risks created by DoS and DDoS attacks. Access control systems, authentication systems, encryption and virtualization systems are distinguished from network functions and especially NFV (network function virtualization)

In the SDN, numerous defense strategies can be used to apply security solutions and attenuate attacks:

– a defense strategy based on rules is defined in a dynamic manner and uses system properties and network statistics;

– an automatic learning defense strategy detects attacks and generates security rules;

– a moving target defense strategy is defined by analogy with the honey pot technique or the DMZ in the security of traditional networks, making the surface of the attack unrecognizable to hackers;

– a collaborative/distributed defense strategy shares cyber threats and plans a defense policy collaboratively.

7.2.5.1. *SDN security requirements*

In SDN environments, security of the SDN must be ever-present. SDN security must be integrated in the architecture as well as provided as a service to protect the availability, integrity and confidentiality of all connected resources and information.

Within the architecture, it is necessary to do the following.

– Protect the controller: if the SDN controller is out of service (e.g. due to a DDoS attack), the network also disappears, which means the availability of the SDN controller must be maintained.

– Investigate and remedy: when an incident occurs, it is important to determine what happened, fix it, report it, and build protection against it for the future.

– Establish confidence: it is essential to protect communications throughout the network. This means that the SDN controller, the applications downloaded on it and the peripheral devices it manages must all be trusted entities that should work as they are meant to.

– Secure access to the controller: given it is the centralized location for decision-making, access to the SDN controller must be strictly controlled.

SDN security solutions will address at least one of the above challenges.

7.2.5.2. *Blockchain: an SDN security solution*

Blockchain is a constantly growing list of records in the form of blocks. These blocks are connected and secured using cryptography. A blockchain uses the following elements:

– digital signatures;

– decentralized register;

– an algorithm to reach a consensus;

– each block contains the hash of the previous block, making a chain of blocks called the blockchain.

This technique is an efficient and reliable solution for encrypting communication, principally via the Southbound interface, which is the principal artery of the SDN insofar as it carries control information generated by SDN controllers and applies it via the nodes of the infrastructure layer. In this way, it becomes impossible for hackers to capture control information and then either alter it in a fraudulent manner or attack the source, in this case the SDN controller, which is the brain of this technology.

7.2.5.3. *Cisco infrastructure based on applications (ACI)*

Security in an SDN must be integrated from its conception and deployed for the automation of different services.

Insofar as the SDN can offer reliable network services that are faster and easier to set up, manage and fix, the objective is to reduce the scale of time for threat, detection and response from days or weeks to minutes or hours, all the while maximizing the ability of the operator’s tools to react and attenuate.

The solution offered by Cisco ACI (application centric infrastructure) is a framework in which the applications guide the behavior of the network[1].

Cisco ACI is an SDN solution that provides automation based on rules; the automation of the rules is extended to all workstations, including virtual computers, physical servers and containers.

Cisco adopted the opposite approach to that of VMware: instead of taking intelligence from the equipment, this approach makes the equipment more intelligent.

Cisco ACI allows a network administrator to create quality service models and to apply them to individual applications. If a certain application requires a superior bandwidth priority at any moment, as in a vocal system, a model can be created to assure this priority. The same model can then be used for all other applications with the same needs.

As a result of this process of creating automated models, Cisco ACI creates an environment that has the same advantages as a network defined by traditional software, but with a completely different strategy.

7.3. IoT/IoE security

Given that the IoT/IoE are continually evolving and represent the greater part of any computer system, they need to be secured with the objective of preventing any unauthorized use and neutralizing any attempt to infiltrate and attack services, applications and infrastructures.

All of these characteristics present several challenges at every level, which makes it essential to have security via various techniques, both generic and specific.

7.3.1. *Sensor networks*

In a world where everyone is connected, the number of connected objects is far greater – five or six times as many – than the number of people. Sensor networks are composed of objects and once they are appropriately programmed, they are capable of automating the evaluation of data (even in

1 www.cisco.com, developer.cisco.com.

great quantities) and modifying processing processes. A good illustration of this is smart cities, which make use of the intelligence of sensors to automate a wide array of municipal services tied to traffic, parking, control of water, electric and gas consumption, and various other installations and infrastructures. We can also point to self-driving cars with different specific sensors, cameras, GPS and computer-assisted services.

7.3.1.1. *Characteristics and requirements of sensor networks*

The Internet of Things (IoT) is the connection of millions of devices and smart sensors connected to the Internet. Sensor networks generate data, automate processing and need to be secured.

The data generated by the IoT is very high in volume. For example, a self-driving car can generate 4,000 gigabytes (Go) of data per day, and a smart home can produce up to 1 Go of data per week.

This data, stored and managed in the form of Big Data, poses problems in terms of management, security, redundancy, analytics and access.

The IoT opens an entirely new world in which tasks that previously required human intervention can now be automated, notably tasks that must be carried out in dangerous conditions or repetitive tasks.

7.3.1.2. *Application domains for the IoT/IoE*

The use of the IoT/IoE covers many areas of application, for example:

– smart homes;

– smart buildings;

– smart factories;

– smart cities;

– smart networks (roads, electricity, water supply, sanitation, communication);

– smart cars;

– management of stores and services;

– medical and surgical diagnostics;

– autopiloting and monitoring of air and naval control.

7.3.2. *Security issues in the IoT*

With the IoT/IoE spreading throughout the world, new and diverse security challenges have arisen that result in specific attacks which can be launched in these kinds of environments.

7.3.2.1. *Security challenges for the IoT*

Sensor networks present specific challenges concerning the protection of devices connected to the IoT due to:

– *the increase in the number of devices*: the number of interconnected sensors and smart devices increases in an exponential manner, which in turn increases the risk of attack;

– *the diversification of device locations*: certain devices connected to the IoT can interact with the physical world;

– *the difficulties of updates*: IoT devices with sensors can be found in faraway and/or inaccessible places, which makes interventions or configurations complicated;

– *the use of wireless communication systems*: this technology facilitates access and thus makes attacks on sensors and IoT services more likely.

The vulnerabilities of IoT networks vary from weak or non-existent authentication to integrated non-secured servers, and to applications with little security that allow hackers to break through access limitations with relative ease.

A user can download web clients functioning on remote devices and systems or on the cloud in order to access and read email on IoT services and systems, which makes them vulnerable to the same attacks as any other computer.

The use of many IoT devices over a long period of time – with old or outdated operating systems or misconfigurations having not been created following material and software security norms – make them vulnerable and the target of attacks, which constitutes a major challenge in security matters.

7.3.2.2. *Security attacks in the IoT*

IoT attacks can be classified into various types as follows:

– *peripheral-level attacks*: material vulnerabilities, physical vulnerabilities of constrained devices, vulnerabilities of micro-software;

– *communication-level attacks*: IP, TCP, UDP and ICMP vulnerabilities;

– *application-level attacks*: vulnerability of local applications, as well as web and cloud applications.

The constrained devices are often placed in faraway places where it becomes hard to ensure physical security.

The potential material vulnerabilities can include:

– theft of the device;

– physical damage to the device;

– deactivation of the device and suppression of the energy source;

– deactivation of the communication, disconnection of cables or other forms of disruption.

The vulnerabilities of micro-software are:

– default connection logins that are not updated. It is important that user names and passwords be modified to respond to strict criteria before connecting an IoT device to the Internet;

– attacks by distributed denial-of-service (DDoS);

– outdated micro-software that has not been fixed with a patch;

– buffer overflow attacks;

– installation of a backdoor.

The most common TCP/IP attacks are:

– *DoS attacks*: perpetrators try to prevent legitimate users from accessing information or services;

– *DDoS attacks*: this is similar to a DoS attack, but includes a simultaneous and coordinated attack from numerous source computers;

– *ICMP attacks*: perpetrators use echo ICMP (internet control message protocol) packets to discover sub-networks and hosts on a protected network to generate DoS overflow attacks and to modify the routing tables of hosts;

– *address takeover attacks*: perpetrators put the source IP address in a packet to appear as a different source, tricking the destination by making it believe that the packet came from a legitimate source;

– *man-in-the-middle attack (MITM)*: perpetrators place themselves between a source and a destination in order to watch, capture and control communication in a transparent manner. They can spy by only inspecting captured packets or modify the packets before passing them to their original destination;

– *session diversion*: perpetrators only have access to the physical network so they use an MITM attack to command a valid token to help them access a web server.

7.3.3. *Blockchain: an IoT security solution*

The blockchain technique, already presented in this chapter, is generally used to secure transactions and communications between sensors and IoT network nodes. Because the sensors are critical entities and communicate with the smallest of resources in terms of energy and configuration, they need to be secured with a distributed and autonomous solution.

This technique can be used to resolve numerous security and confidence challenges for the IoT by:

– monitoring sensor data and preventing malicious data;

– providing IoT device identification, authentication and the secure transfer of data;

– allowing IoT sensors to securely and directly exchange data between themselves, without an intermediary;

– eliminating a unique source of failure in the IoT ecosystem with a distributed register;

– simplifying deployment of the IoT with lower operating costs because there is no intermediary;

– IoT devices are directly addressable with the blockchain, providing a permanent history.

7.4. Conclusion

The SDN, as a network solution based on virtualization, makes its management, administration and even its security more flexible through the use of software solutions. With the evolution of both virtual technologies and cloud computing, the SDN will increasingly promote the automation of control services and tasks for managing networks that encompass surveillance and filtering functions, providing better security.

Access control is ensured via scripts and programs using a large panoply of programming languages. Encryption and integrity verification can be guaranteed using the blockchain.

Given that the world is becoming increasingly connected, a large number of new people and about five times more objects connect to each other every day and participate in generating enormous quantities of data.

Sensors present limitations and challenges in terms of security, at both the physical and energy levels, as well as micro-software and applications.

Security can be ensured via a range of measures, including physical security, energy autonomy and access control using the blockchain, in addition to filtering techniques and traditional security.

8

Security Management

8.1. Introduction

Computer security has become a very important subject for companies and institutions, as well as individuals. Because of this, numerous legal texts have been enacted in most of the world's countries to structure this area and define solutions and measures to adopt when securing computer systems, which constitute the backbone and a valuable asset for companies, as well as individuals.

A security audit is a necessary legal and economic legal requirement for the survival and existence of the company, as well as for its reputation and influence. It is a periodic task led by security experts to identify security vulnerabilities and faults, along with the appropriate solutions and recommendations. A new discipline has thus been developed, that of the security consultant or auditor.

An audit goes through three necessary stages. The first step concerns the organizational and physical aspect. It identifies structural and physical vulnerabilities. Then, the second step is devoted to the technical aspect. It consists of uncovering security faults at various levels and providing the necessary solutions for such problems. Finally, the intrusive test must be passed, which consists of bombarding our own system with a series of attacks to evaluate how robust it is.

In the domestic sphere, as well as at the company level, it is necessary to plan a security policy with the objective of limiting risks, attenuating attacks

Computer Science Security: Concepts and Tools,
First Edition. Ameur Salem Zaidoun.

and increasing the efficiency of security solutions. A security policy covers several axes, intervenes at several levels and calls on different actors to apply it appropriately.

In order to develop a security policy, it is necessary to first test the state of things and evaluate the existing level of security.

The development and follow-through of a security policy is governed by a piloting committee led by a security manager. Once the policy is developed, depending on the existing norms and standards and the identified interlocutors, its application will require the definition of directives and procedures that will facilitate its implementation and efficiency.

8.2. Security audits

A security audit, required by law in many countries, is necessary for a company to overcome the challenges of attacks and security failures.

8.2.1. *Objectives*

Security presents a very specific and specialized problem that can under no circumstances be resolved internally or by company personnel. Their expertise may be limited, which could prevent them from properly identifying and fixing the problem. An audit must warn users first, and after the identification of faults, must then provide solutions and measures to face them, in the form of recommendations.

8.2.1.1. *Creating a security culture*

Security is a culture above all, and the personnel of a company must have a minimum of this culture if they are to remain aware, secure and behave appropriately when faced with potential attacks.

An audit must focus on security and make the management, agents and clients aware of the issue of security by identifying bad habits. This culture can be presented through organizations, posters, charters and control measures limiting physical and computer access.

8.2.1.2. *Establishing technical security solutions*

After identifying security faults, the audit team must try to find solutions for such problems. The solutions must cover several aspects:

– topology and the interconnections of the local network;

– Internet connection;

– operating systems;

– applications in use;

– current means for authentication.

8.2.2. *Audit action diagram*

The action diagram during an audit is summarized in the following figure.

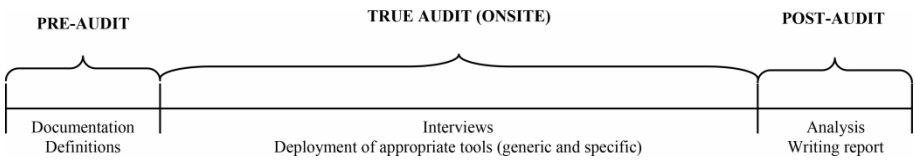

Figure 8.1. *Action diagram in an audit*

The first phase is devoted to the documentation and preparation of the task, with a precise and clear definition of the actors, interlocutors, circumstances, limits and obligations.

The second phase is the audit in the strictest sense, the on-site intervention using the appropriate generic and specific tools, both technical and otherwise.

Finally, during the third phase, the audit team focuses on the analysis of the results and the writing of a report.

Companies should not limit themselves to getting audits and then leaving the report to one side. It is important to apply the recommendations made, especially when the report is validated by authorized authorities.

Structure of computing and/or security services

Most companies neglect computing and security insofar as it serves as a tool facilitating the work and is not in and of itself a productive unit. The measures taken for computing are the product of numerous users who are not specialists in the field. For this reason, following every consultation, it is necessary create, restructure, enrich and ensure the independence of computing services and/or the security unit.

Training

For personnel development, training activities are necessary. The audit must provide an indicator regarding security knowledge within a company, and the investment in training should be based on this indicator.

Establishing security solutions

This is the technical aspect to implement after a security consultation. It covers several areas:

– Internet connection;

– antivirus;

– firewall;

– encryption;

– authentication.

8.2.3. *Organizational and physical audit*

Also called a high-level audit, this estimates risk by analyzing organizational and physical vulnerabilities.

8.2.3.1. *Objectives*

This is a preliminary stage that allows us to critique organizations and uncover bad structures or a bad division of labor and responsibilities among the participants.

The second stage is devoted to physical security through a critique of physical access to computing system sites.

8.2.3.2. *Utilities and implementation*

The discovery of organizational and physical vulnerabilities is done through appropriate questionnaires for management, agents and clients, covering various aspects.

We can also use known models that offer a database of questionnaires related to many topics, and that can be adapted depending on the context of the audited company. This produces statistics and security indicators.

There are two possible approaches: ascending and descending.

– *Descending approach*: this is a systematic, complete approach, but takes a long time to implement.

– *Ascending approach*: this is an intuitive, incomplete approach whose implementation is fast, but has certain oversights in terms of precision.

8.2.4. *Technical audit*

Also called a low-level audit, this identifies technical vulnerabilities (in systems and networks). This part requires expertise from the auditors that covers multiple areas. There are several aspects that require auditing:

– network topology;

– system resistance;

– servers;

– connection equipment;

– network applications;

– SGBDs and databases;

– messaging systems;

– specific applications.

8.2.4.1. *Objectives*

This is the most important part, providing an audit of systems and computer applications by finding technical vulnerabilities and faults, on the

one hand, and formulating appropriate technical recommendations, on the other hand.

A technical audit requires auditors with a minimum of expertise who will discuss issues with administrators and security managers by way of questionnaires and discussions, as well as the use of appropriate tools for systems, services and applications.

The tools used vary depending on the context and importance of the computer service and its content in terms of equipment and deployed solutions, data traffic, services and functions provided.

8.2.4.2. *Implementation tools*

Many computing tools can be used for a technical audit. The most appropriate tools are free software, the very same tools used by hackers and attackers.

Examples of these tools are as follows:

– *Nessus*: this is a tool that discovers security faults in a network or segment of network.

– *NMAP*: this is a tool that identifies open doors in a network, sub-network or computer.

– *LANguard*: this is a tool that detects patches, shares, open doors, unused user accounts and missing service packs.

8.2.5. *Intrusive test*

This is a simulation of attacks that tests the strength of a computer system and its response to attacks. It consists of using attack tools and observing the way the system reacts. The intrusive test must be programmed carefully to avoid perturbing proper functioning.

8.2.6. *Audit methodologies*

There are several audit methodologies that can be adapted during an audit.

8.2.6.1. *ISO 17799*

Descended from the British BS 7799 norm, the ISO 17799 norm provides guidelines and recommendations for managing security[1].

The ISO 17799 norm thus offers a model for identifying and implementing solutions for the following risks:

– *Security policy*: this writes and disseminates the company security policy.

– *Security organization*: this defines roles and responsibilities, and also takes control of partners and outside activity.

– *Asset classification and control*: this takes inventory of company assets and defines how critical they are and their associated risk.

– *Personnel security*: this consists of hiring, training and security sensitivity training.

– *Physical and environmental security*: this consists of security perimeters and an inventory of security equipment.

– *Communication and operation management*: this consists of procedures in case of an accident, recovery plans, definition of service levels and recovery times.

– *Access control*: this consists of establishing access controls at different levels (systems, networks, buildings, etc.).

– *System development and maintenance*: this takes into account security notions in systems from conception to maintenance.

– *Business continuity planning*: this consists of definitions of availability needs, recovery times and emergency exercises.

– *Compliance*: this consists of respecting author rights, legislation and the rules of the company.

8.2.6.2. *MARION*

The MARION method (acronym for the French equivalent of Methodology of Analysis of Computing Risks Oriented by Level) is an audit methodology that, as its name suggests, evaluates the level of security in a

1 www.iso.org.

company (the risks) by way of moderated questionnaires that provide indicators in the form of notes on different topics related to security.

The level of security is evaluated following 27 indicators divided into six themes, each given a score of 0 to 4, with level 3 being the level to reach if we are to have what is considered adequate security. The method is based on questionnaires focusing on precise areas. The questionnaires must allow vulnerabilities specific to the company in all areas of security to be evaluated.

The group of indicators is evaluated using several hundreds of questions, whose responses are moderated (the moderations evolving along with the updates to the method).

The themes are as follows:

– organizational security;

– physical security;

– continuity;

– computer organization;

– computer security and use;

– application security.

8.2.6.3. *MEHARI*

MEHARI (acronym for the French equivalent of Harmonized Method of Risk Analysis) is derived from two other risk analysis methods, MARION and MELISA. This method was developed and kept in France by CLUSIF, the French Security Club for Information Systems. MEHARI is one of the most commonly used methods of risk analysis today. This method appears to be a veritable tool kit for computer system security, identifying risks within an organization in various ways. This tool kit is made up of several modules which, independently of the selected security method, provide in particular:

– an analysis of security stakes (by describing the types of feared malfunctions), and a classification of resources and information following the three basic tenets of security: confidentiality, integrity and availability;

– an audit of security services to evaluate the efficiency of each, its control and to summarize vulnerabilities;

– an analysis of risky situations, providing an evaluation of possibilities and intrinsic impacts, as well as risk attenuation factors, and finally, an indicator of risk gravity.

8.3. Security policy demonstration

A security policy, written and published, must be established in any organization making use of a large- or medium-sized computer system. It requires a test to validate and measure the requirements and the existing levels of security.

8.3.1. *Security test and evaluation*

Tests and evaluations are a necessary first step before proceeding to the formulation of a security policy. Security tests take many forms and use diverse and targeted tools.

8.3.1.1. *Security test types*

Security tests are used to verify the strength of a computer system and identify weaknesses and limits. These tests fall into the following categories:

– penetration test;

– network analysis;

– vulnerability analysis;

– password cracking;

– examination of records;

– integrity control;

– virus detection.

These types of tests cover the different areas of security and their results make up an inventory of the computer system, which will be the starting point for defining and formulating a security policy.

8.3.1.2. *Security test tools*

To provide an efficient evaluation, we should use different types of free software, the same ones used by hackers and attackers.

The most commonly used tools are:

– Nmap/Zenmap;

– SuperScan;

– SIEM;

– GFI LANguard;

– Tripwire;

– Nessus;

– L0phtCrack;

– Metaspoilt.

Nmap/Zenmap

Nmap, short for Network Mapper, is a free and open-source tool used for discovering networks and security audits[2]. Many system and network administrators also find it useful for tasks, such as network inventories, calendar management for service updates and monitoring the availability of the host or service. Nmap uses raw IP packets in an innovative way to determine which hosts are available on the network, which services (name and application version) these hosts offer, which operating systems (and versions of the operating systems) they use, what kind of packet filters or firewalls are being used and dozens of other characteristics. It was conceived to analyze large networks quickly, but works very well for unique hosts. Nmap works on all the principal computer operating systems, and official binary packages are available for Linux, Windows and Mac OS X. In addition to the traditional executable line command Nmap, the later version includes an advanced graphic interface and a Results Viewer (Zenmap), a flexible tool for data transfer, redirection and debugging (Ncat), a tool for comparing analysis results (Ndiff) and a tool for generating packets and response analysis (Nping).

Nmap has the following characteristics:

– *Flexible*: it takes charge of dozens of advanced techniques to map out networks filled with IP filters, firewalls, routers and other obstacles. This

2 nmap.org.

includes numerous mechanisms for analyzing ports (TCP and UDP), detecting operating systems, versions, ping sweeps and so on.

– *Powerful*: Nmap has been used to analyze enormous networks with literally hundreds of thousands of computers.

– *Portable*: most operating systems are supported, namely, Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS, Amiga and so on.

– *Easy*: traditional versions of line command are available in addition to graphic versions (GUI) using binaries for those who do not wish to compile Nmap from sources.

– *Free*: the principal objective of the Nmap project is to contribute to making the Internet a bit more secure and offer administrators, auditors and hackers an advanced tool for exploring networks. Nmap is available for free download, and is also delivered with a complete source code that can be modified and redistributed according to the license terms.

Zenmap is the official graphic interface of the security scanner, Nmap. It is a free and open-source multi-platform application (Linux, Windows, Mac OS X, BSD, etc.) that makes Nmap easy to use for beginners, all while providing advanced functions for experienced Nmap users. Frequently used analyses can be recorded as profiles to facilitate repeated execution. The results of the saved analyses can be compared to each other to see how they differ.

GFI LANguard

GFI LANguard is the award-winning network analysis and security software used by more than 20,000 clients[3]. GFI LANguard analyzes networks and ports to detect and repair vulnerabilities with minimal management.

Any network administrator must individually manage vulnerability issues, patches and audits, sometimes using several products. With GFI LANguard, these three keystones of vulnerability can be managed with just one software program.

3 www.zdnet.fr.

GFI LANguard provides a complete image of an installation network and helps maintain the network secure, easily and efficiently.

Tripwire

Tripwire is integrity control software that ensures that sensitive files on a computer are not modified without it creating an alert[4]. To do this, the software creates a database (or a reference table for simpler cases) containing the digital signature (hash) of files that the administrator wishes to keep an eye on. During the integrity control phase, Tripwire recalculates the digital signature of each file to be monitored and verifies that this signature corresponds to the one calculated at the time the database was created. If the two signatures do not correspond, Tripwire sends an alert.

Monitored files can be classified according to different degrees of criticality. Tripwire can be rather complex to configure because files and configurations are encrypted. The alerts sent after the files are modified can be sent via email.

Nessus

Nessus is a computer security tool[5]. It signals potential or clear weaknesses on tested computers. This includes, among others, services vulnerable to attacks that would allow control to be taken of the computer, sensitive information to be accessed, and service to be denied.

Nessus works with Unix and Windows; its last version was 8.4.0, published on May 14, 2019.

It is widely used by users and computer system administrators. It is generally used by security managers and auditors.

Nessus detects live computers on a network, sweeps open doors, identifies active services and their versions, and then attempts various attacks.

Nessus is divided in two parts: nessusd, which is a daemon (service) executing requests and communication with the target, and nessus, which is a client application that recovers data and shows the result.

4 www.tripwire.com.

5 https://www.nessus.org.

This division is classic, with the daemon running with advanced privileges (root), while the graphic interface, more complex and thus vulnerable, runs under the identity of an unprivileged user. The tests are played out using plugins; some are compiled in C, but most are written in the script language NASL (Nessus Attack Scripting Language).

Nessus is a network security scanner capable of detecting weaknesses that can be exploited locally; as well as remotely, by either:

– identifying a version number in a banner, but this process is limited to one particular class of weaknesses: network service weaknesses that can only be exploited locally; or

– acquiring the list of software or packets installed on the computer being tested and comparing it to the patches published by editors.

8.3.2. *Security policy development*

A security policy that appears as a well-defined action plan composed of a group of measures guarantees a minimal level of security. This policy covers all aspects and calls on many interlocutors at various levels. It must foresee measures to take and people to alert in case an organizational failure or a technical intrusion is detected.

The principal objective, namely maintaining a level of security, comes through the establishment of a culture and technical security solutions.

The first sub-objective is ensured through training sessions, charters, posters, sports, alerts and so on.

The second sub-objective covers policies, archives, monitoring, analysis, filtering, updates, alerts and so on.

Once defined, a security policy must be formulated and written before being distributed and applied.

8.3.2.1. *Security policy interlocutors*

For the development and establishment of a security policy, numerous actors can be called upon:

– *Security manager*: this is an individual charged with developing and updating the security policy.

– *Committee of collaborators*: this is a team of diverse specialists in the company that can help the security manager develop, apply, monitor and update the security policy.

– *Users*: these are the personnel making use of and accessing computer services.

– *External users*: these are composed of consultants and security auditors participating in audits within the company or assisting with the resolution of a precise security problem, or any other external collaborator in the field of computing.

8.3.2.2. *Security policy steps*

The establishment and application of a security policy goes through several steps, which are defined as follows:

– designate a computer security manager who will be in charge of creating, applying and updating the security policy;

– define the perimeter and objectives of the computer security policy, in order to limit the field of application of this policy and be able to evaluate its impacts and influences for better efficiency;

– analyze the existing equipment and software, and maintain an updated register of all of the elements making up the computer system. This register is important during modifications of components of the computer configuration. In the case of an incident, it can help IT teams find the origin of the problem and identify responsibilities;

– analyze computing risks in terms of possible damage and the probability that an incident will occur;

– determine the necessary means for reducing risks and taking charge of incidents, or for managing continuous activity;

– write a computing charter for all collaborators;

– communicate the computer security policy, with all of its details and procedures, to all users within the company.

8.3.3. *Elements of a security policy*

For a security policy to be efficient and applicable, it must be composed of several complementary elements that cover different hierarchical levels of the company.

8.3.3.1. *Governance policy*

The importance of computer security demands a high-level intervention in a company with the purpose of increasing the efficiency of decision-making and action-taking.

To satisfy this requirement, a security manager and security committee are necessary.

The security manager reports to the management of computer services, fulfilling the following tasks and abilities:

– they can send alerts to general management;

– they have the logistical and financial means and the necessary authority to complete tasks;

– they can define and apply the security policy in direct cooperation with the members of the security committee;

– they must periodically prepare a report with a security inventory.

The security committee includes and brings together the pertinent actors of a company, that is, all of the directors. It is led by the security manager and ensures technical vigilance and creates a culture of cyber security.

8.3.3.2. *Technical policy*

The technical section of the security policy covers technical domains. It varies depending on the material and software assets installed, and the degree of criticality of the information that is dealt with.

A technical security policy is composed of a group of technical measures and solutions for saving, filtering, monitoring and follow-up. The distinct areas are as follows:

– establishment of anti-malware software such as antivirus, antispam and antispyware, with the necessary updates and patches;

– deployment of filtering solutions using routers and firewalls;

– development and application of a saving policy;

– development of specific solutions to apply depending on installed services.

8.3.3.3. *End-user policy*

The section on security culture is of capital importance. It must be addressed to all personnel without neglecting any of the end-users. The end-user is anyone with access to the computer system and represents an important link in the chain of computer security and the policy to be established.

The user must be sensitized and informed via targeted training sessions, posters and charters, which must be adhered to.

8.4. Norms, directives and procedures

Several norms and standards can be used when conducting an audit or following up on its results and impacts, as well as in establishing a computer security policy:

– *family of ISO 20000 norms*: ISO 20000-1 and ISO 20000-2 norms are standards describing the management processes for efficient delivery of computer services to a company and its clients. These respect ITIL requirements[6];

– *family of ISO 27000/ISMS norms*: establishment, use, updating and management of a computer security policy, or information security management systems (ISMS);

– *ISO/FDIS 31000 norm*: risk management;

– *ISO/IEC 38500 norm*: computer security governance;

– *British Standards Institution BS 25999-1 norms*: BCM, practice codes;

– *British Standards Institution BS 25999-2 norms*: BCM, specifications.

6 www.iso.org.

8.4.1. *ISO 27000 norm*

The ISO 27000 series of norms was specifically reserved by the ISO for questions of information security. The 27000 series includes a range of individual norms and documents. A certain number of them have already been published.

– *ISO 27001*: this is the specification of an information security management system (ISMS) that replaced the former BS7799-2 norm.

– *ISO 27002*: this is the standard number of the 27000 series that was at the origin of the ISO 17799 norm (formerly known as BS7799-1).

– *ISO 27003*: this will be the official number of a norm meant to offer suggestions for establishing an ISMS (IS management system).

– *ISO 27004*: this norm covers measures and metrics for managing information security systems, including the controls suggested in ISO 27002.

– *ISO 27005*: this ISO norm is independent of the methodology for managing risks connected with information security.

– *ISO 27006*: this norm offers guidelines for accrediting organizations that offer an ISMS certification.

8.4.2. *ISO/FDIS 31000 norm*

ISO 31000 designates a family of norms for managing risks, codified by the international organization for normalization. The objective of the ISO 31000 norm is to offer the principles and guidelines for risk management, as well as the processes for establishing it strategically and operationally. It does not seek to promote the uniformity of risk management in organizations, but rather to harmonize the multitude of existing approaches, standards and methodologies for risk management.

Currently, the ISO 31000 family includes:

– ISO 31000:2018, Risk management, Principles and Guidelines;

– ISO/CEI 31010:2009, Risk Management, Risk Evaluation Techniques;

– ISO Guide 73:2009, Risk Management, Vocabulary.

8.4.3. *ISO/IEC 38500 norm*

ISO/IEC 38500 is the international norm for the governance of information technology by companies. It is the first official norm for computer governance.

This norm concerns the governance of management processes related to information and communication services used by an organization. These processes can be controlled by computer specialists in an organization or by external service providers.

8.5. Conclusion

Security audits are necessary for securing computer systems, but they remain insufficient; they must be completed and heeded by technological monitoring throughout a company, in order to follow the daily state of security through audit and surveillance tools, and through a strictly cooperative relationship with appropriate organizations.

The security audit constitutes an important step for the survival of a company. It provides a constructive external critique by experts in the domain. This task, required by law, is beneficial for a company seeking to protect its computing assets.

A security policy, well-defined and correctly applied, is the finishing touch to the security measures and activities taken by users. It covers organizational and technical aspects and creates a culture that protects the company from risks and threats.

Computer security continuously presents itself as an urgent issue, to the detriment of other factors; it takes its importance and relevance from both circumstances and conditions that remind us of the volume of destruction, material and otherwise, for individuals, companies and states.

References

Boutherin, B. and Delaunay, B. (2003). *Sécuriser un réseau Linux*. Editions Eyolles, Paris.

Cisco (2021). Cisco platform [Online]. Available at: cisco.netacad.net and developer.cisco.com.

Huawei (2021). Huawei platform [Online]. Available at: e.huawei.com and support.huawei.com.

Webography

CompTIA (n.d.). https://www.comptia.org.

ISO (2021). www.iso.org.

Jakusic, M. (2021). www.securite.teamlog.com.

Nessus (2022). https://www.nessus.org.

Sécurité Info (2021). www.securiteinfo.com.

Snort (2022). https://www.snort.org.

The Virus Encyclopedia (n.d.). http://virus.wikidot.com.

Tripwire (n.d.). https://www.tripwire.com.

Computer Science Security: Concepts and Tools,
First Edition. Ameur Salem Zaidoun.

Index

Computer Science Security: Concepts and Tools,
First Edition. Ameur Salem Zaidoun.

G, I, K

L, M, N

P, R, S

T, V, W

Other titles from

in

Computer Engineering

2021

DELHAYE Jean-Loic
Inside the World of Computing: Technologies, Uses, Challenges

DUVAUT Patrick, DALLOZ Xavier, MENGA David, KOEHL François, CHRIQUI Vidal, BRILL Joerg
Internet of Augmented Me, I.AM: Empowering Innovation for a New Sustainable Future

HARDIN Thérèse, JAUME Mathieu, PESSAUX François, VIGUIÉ DONZEAU-GOUGE Véronique
Concepts and Semantics of Programming Languages 1: A Semantical Approach with OCaml and Python
Concepts and Semantics of Programming Languages 2: Modular and Object-oriented Constructs with OCaml, Python, C++, Ada and Java

MKADMI Abderrazak
Archives in The Digital Age: Preservation and the Right to be Forgotten (Digital Tools and Uses Set – Volume 8)

TOKLU Yusuf Cengiz, BEKDAS Gebrail, NIGDELI Sinan Melih
Metaheuristics for Structural Design and Analysis (Optimization Heuristics Set – Volume 3)

2020

DARCHE Philippe
Microprocessor 1: Prolegomena – Calculation and Storage Functions – Models of Computation and Computer Architecture
Microprocessor 2: Core Concepts – Communication in a Digital System
Microprocessor 3: Core Concepts – Hardware Aspects
Microprocessor 4: Core Concepts – Software Aspects
Microprocessor 5: Software and Hardware Aspects of Development, Debugging and Testing – The Microcomputer

LAFFLY Dominique
TORUS 1 – Toward an Open Resource Using Services: Cloud Computing for Environmental Data
TORUS 2 – Toward an Open Resource Using Services: Cloud Computing for Environmental Data
TORUS 3 – Toward an Open Resource Using Services: Cloud Computing for Environmental Data

LAURENT Anne, LAURENT Dominique, MADERA Cédrine
Data Lakes
(Databases and Big Data Set – Volume 2)

OULHADJ Hamouche, DAACHI Boubaker, MENASRI Riad
Metaheuristics for Robotics
(Optimization Heuristics Set – Volume 2)

SADIQUI Ali
Computer Network Security

VENTRE Daniel
Artificial Intelligence, Cybersecurity and Cyber Defense

2019

BESBES Walid, DHOUIB Diala, WASSAN Niaz, MARREKCHI Emna
Solving Transport Problems: Towards Green Logistics

CLERC Maurice
Iterative Optimizers: Difficulty Measures and Benchmarks

GHLALA Riadh
Analytic SQL in SQL Server 2014/2016

TOUNSI Wiem
Cyber-Vigilance and Digital Trust: Cyber Security in the Era of Cloud Computing and IoT

2018

ANDRO Mathieu
Digital Libraries and Crowdsourcing
(Digital Tools and Uses Set – Volume 5)

ARNALDI Bruno, GUITTON Pascal, MOREAU Guillaume
Virtual Reality and Augmented Reality: Myths and Realities

BERTHIER Thierry, TEBOUL Bruno
From Digital Traces to Algorithmic Projections

CARDON Alain
Beyond Artificial Intelligence: From Human Consciousness to Artificial Consciousness

HOMAYOUNI S. Mahdi, FONTES Dalila B.M.M.
Metaheuristics for Maritime Operations
(Optimization Heuristics Set – Volume 1)

JEANSOULIN Robert
JavaScript and Open Data

PIVERT Olivier
NoSQL Data Models: Trends and Challenges
(Databases and Big Data Set – Volume 1)

SEDKAOUI Soraya
Data Analytics and Big Data

SALEH Imad, AMMI Mehdi, SZONIECKY Samuel
Challenges of the Internet of Things: Technology, Use, Ethics
(Digital Tools and Uses Set – Volume 7)

SZONIECKY Samuel
Ecosystems Knowledge: Modeling and Analysis Method for Information and Communication
(Digital Tools and Uses Set – Volume 6)

2017

BENMAMMAR Badr
Concurrent, Real-Time and Distributed Programming in Java

HÉLIODORE Frédéric, NAKIB Amir, ISMAIL Boussaad, OUCHRAA Salma, SCHMITT Laurent
Metaheuristics for Intelligent Electrical Networks
(Metaheuristics Set – Volume 10)

MA Haiping, SIMON Dan
Evolutionary Computation with Biogeography-based Optimization
(Metaheuristics Set – Volume 8)

PÉTROWSKI Alain, BEN-HAMIDA Sana
Evolutionary Algorithms
(Metaheuristics Set – Volume 9)

PAI G A Vijayalakshmi
Metaheuristics for Portfolio Optimization
(Metaheuristics Set – Volume 11)

2016

BLUM Christian, FESTA Paola
Metaheuristics for String Problems in Bio-informatics
(Metaheuristics Set – Volume 6)

DEROUSSI Laurent
Metaheuristics for Logistics
(Metaheuristics Set – Volume 4)

DHAENENS Clarisse and JOURDAN Laetitia
Metaheuristics for Big Data
(Metaheuristics Set – Volume 5)

LABADIE Nacima, PRINS Christian, PRODHON Caroline
Metaheuristics for Vehicle Routing Problems
(Metaheuristics Set – Volume 3)

LEROY Laure
Eyestrain Reduction in Stereoscopy

LUTTON Evelyne, PERROT Nathalie, TONDA Albert
Evolutionary Algorithms for Food Science and Technology
(Metaheuristics Set – Volume 7)

MAGOULÈS Frédéric, ZHAO Hai-Xiang
Data Mining and Machine Learning in Building Energy Analysis

RIGO Michel
Advanced Graph Theory and Combinatorics

2015

BARBIER Franck, RECOUSSINE Jean-Luc
COBOL Software Modernization: From Principles to Implementation with the BLU AGE® Method

CHEN Ken
Performance Evaluation by Simulation and Analysis with Applications to Computer Networks

CLERC Maurice
Guided Randomness in Optimization
(Metaheuristics Set – Volume 1)

DURAND Nicolas, GIANAZZA David, GOTTELAND Jean-Baptiste, ALLIOT Jean-Marc
Metaheuristics for Air Traffic Management
(Metaheuristics Set – Volume 2)

MAGOULÈS Frédéric, ROUX François-Xavier, HOUZEAUX Guillaume
Parallel Scientific Computing

MUNEESAWANG Paisarn, YAMMEN Suchart
Visual Inspection Technology in the Hard Disk Drive Industry

2014

BOULANGER Jean-Louis
Formal Methods Applied to Industrial Complex Systems

BOULANGER Jean-Louis
Formal Methods Applied to Complex Systems:Implementation of the B Method

GARDI Frédéric, BENOIST Thierry, DARLAY Julien, ESTELLON Bertrand, MEGEL Romain
Mathematical Programming Solver based on Local Search

KRICHEN Saoussen, CHAOUACHI Jouhaina
Graph-related Optimization and Decision Support Systems

LARRIEU Nicolas, VARET Antoine
Rapid Prototyping of Software for Avionics Systems: Model-oriented Approaches for Complex Systems Certification

OUSSALAH Mourad Chabane
Software Architecture 1
Software Architecture 2

PASCHOS Vangelis Th
Combinatorial Optimization – 3-volume series, 2nd Edition
Concepts of Combinatorial Optimization – Volume 1, 2nd Edition
Problems and New Approaches – Volume 2, 2nd Edition
Applications of Combinatorial Optimization – Volume 3, 2nd Edition

QUESNEL Flavien
Scheduling of Large-scale Virtualized Infrastructures: Toward Cooperative Management

RIGO Michel
Formal Languages, Automata and Numeration Systems 1: Introduction to Combinatorics on Words
Formal Languages, Automata and Numeration Systems 2: Applications to Recognizability and Decidability

SAINT-DIZIER Patrick
Musical Rhetoric: Foundations and Annotation Schemes

TOUATI Sid, DE DINECHIN Benoit
Advanced Backend Optimization

2013

ANDRÉ Etienne, SOULAT Romain
The Inverse Method: Parametric Verification of Real-time Embedded Systems

BOULANGER Jean-Louis
Safety Management for Software-based Equipment

DELAHAYE Daniel, PUECHMOREL Stéphane
Modeling and Optimization of Air Traffic

FRANCOPOULO Gil
LMF — Lexical Markup Framework

GHÉDIRA Khaled
Constraint Satisfaction Problems

ROCHANGE Christine, UHRIG Sascha, SAINRAT Pascal
Time-Predictable Architectures

WAHBI Mohamed
Algorithms and Ordering Heuristics for Distributed Constraint Satisfaction Problems

ZELM Martin *et al.*
Enterprise Interoperability

2012

ARBOLEDA Hugo, ROYER Jean-Claude
Model-Driven and Software Product Line Engineering

BLANCHET Gérard, DUPOUY Bertrand
Computer Architecture

BOULANGER Jean-Louis
Industrial Use of Formal Methods: Formal Verification

BOULANGER Jean-Louis
Formal Method: Industrial Use from Model to the Code

CALVARY Gaëlle, DELOT Thierry, SÈDES Florence, TIGLI Jean-Yves
Computer Science and Ambient Intelligence

MAHOUT Vincent
Assembly Language Programming: ARM Cortex-M3 2.0: Organization, Innovation and Territory

MARLET Renaud
Program Specialization

SOTO Maria, SEVAUX Marc, ROSSI André, LAURENT Johann
Memory Allocation Problems in Embedded Systems: Optimization Methods

2011

BICHOT Charles-Edmond, SIARRY Patrick
Graph Partitioning

BOULANGER Jean-Louis
Static Analysis of Software: The Abstract Interpretation

CAFERRA Ricardo
Logic for Computer Science and Artificial Intelligence

HOMES Bernard
Fundamentals of Software Testing

KORDON Fabrice, HADDAD Serge, PAUTET Laurent, PETRUCCI Laure
Distributed Systems: Design and Algorithms

KORDON Fabrice, HADDAD Serge, PAUTET Laurent, PETRUCCI Laure
Models and Analysis in Distributed Systems

LORCA Xavier
Tree-based Graph Partitioning Constraint

TRUCHET Charlotte, ASSAYAG Gerard
Constraint Programming in Music

VICAT-BLANC PRIMET Pascale *et al.*
Computing Networks: From Cluster to Cloud Computing

2010

AUDIBERT Pierre
Mathematics for Informatics and Computer Science

BABAU Jean-Philippe *et al.*
Model Driven Engineering for Distributed Real-Time Embedded Systems

BOULANGER Jean-Louis
Safety of Computer Architectures

MONMARCHE Nicolas *et al.*
Artificial Ants

PANETTO Hervé, BOUDJLIDA Nacer
Interoperability for Enterprise Software and Applications 2010

SIGAUD Olivier *et al*.
Markov Decision Processes in Artificial Intelligence

SOLNON Christine
Ant Colony Optimization and Constraint Programming

AUBRUN Christophe, SIMON Daniel, SONG Ye-Qiong *et al.*
Co-design Approaches for Dependable Networked Control Systems

2009

FOURNIER Jean-Claude
Graph Theory and Applications

GUEDON Jeanpierre
The Mojette Transform / Theory and Applications

JARD Claude, ROUX Olivier
Communicating Embedded Systems / Software and Design

LECOUTRE Christophe
Constraint Networks / Targeting Simplicity for Techniques and Algorithms

2008

BANÂTRE Michel, MARRÓN Pedro José, OLLERO Hannibal, WOLITZ Adam
Cooperating Embedded Systems and Wireless Sensor Networks

MERZ Stephan, NAVET Nicolas
Modeling and Verification of Real-time Systems

PASCHOS Vangelis Th
Combinatorial Optimization and Theoretical Computer Science: Interfaces and Perspectives

WALDNER Jean-Baptiste
Nanocomputers and Swarm Intelligence

2007

BENHAMOU Frédéric, JUSSIEN Narendra, O'SULLIVAN Barry
Trends in Constraint Programming

JUSSIEN Narendra
A TO Z OF SUDOKU

2006

BABAU Jean-Philippe *et al.*
From MDD Concepts to Experiments and Illustrations – DRES 2006

HABRIAS Henri, FRAPPIER Marc
Software Specification Methods

MURAT Cecile, PASCHOS Vangelis Th
Probabilistic Combinatorial Optimization on Graphs

PANETTO Hervé, BOUDJLIDA Nacer
Interoperability for Enterprise Software and Applications 2006 / IFAC-IFIP I-ESA'2006

2005

GÉRARD Sébastien *et al.*
Model Driven Engineering for Distributed Real Time Embedded Systems

PANETTO Hervé
Interoperability of Enterprise Software and Applications 2005

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.

This book serves as a guide to help the reader develop an awareness of security vulnerabilities and attacks, and encourages them to be circumspect when using the various computer resources and tools available today. For experienced users, *Computer Science Security* presents a wide range of tools to secure legacy software and hardware.

Computing has infiltrated all fields nowadays. No one can escape this wave and be immune to security attacks, which continue to evolve, gradually reducing the level of expertise needed by hackers.

It is high time for each and every user to acquire basic knowledge of computer security, which would enable them to mitigate the threats they may face both personally and professionally. It is this combined expertise of individuals and organizations that will guarantee a minimum level of security for families, schools, the workplace and society in general.

Ameur Salem Zaidoun received a National Diploma in Computer Engineering from ENSI, Tunisia, and is a university teacher at ISET of Siliana at the level of Lecturer Technologist. An ex-developer and security consultant, he is a CCNA R&S-, DevNet- and CCNA-Security-certified and a Huawei HCNA-R&S-certified Cisco Instructor.

ISTE
www.iste.co.uk
WILEY