Chapter 1

Security Fundamentals

THE AWS CERTIFIED SECURITY SPECIALTY EXAM OBJECTIVES THAT LEVERAGE CONCEPTS EXPLAINED IN THIS CHAPTER INCLUDE THE FOLLOWING:

- ✓ **Domain 1: Incident Response**
  - 1.2. Verify that the Incident Response plan includes relevant AWS services
- ✓ **Domain 2: Logging and Monitoring**
  - 2.1. Design and implement security monitoring and alerting
- ✓ **Domain 3: Infrastructure Security**
  - 3.1. Design edge security on AWS
  - 3.2. Design and implement a secure network infrastructure
- ✓ **Domain 4: Identity and Access Management**
  - 4.1. Design and implement a scalable authorization and authentication system to access AWS resources
- ✓ **Domain 5: Data Protection**
  - 5.3. Design and implement a data encryption solution for data at rest and data in transit

Introduction

An understanding of the concepts explained in this chapter will be critical in your journey to pass the AWS Certified Security Specialty exam. We will introduce the following topics:

- Basic security definitions
- Foundational networking concepts
- Main classes of attacks
- Important security solutions and services
- Well-known security frameworks and models

In this chapter, you will learn about basic security concepts and some foundational terminology that comes from the information technology (IT) infrastructure knowledge domain. Even if your sole objective is to conquer the AWS Certified Security Specialty certification, this chapter is relevant for any professional, particularly for the officially accredited ones, to demonstrate a good level of general education on the security subject matter (be it related to cloud-based or to traditional on-premises environments).

If you are already an experienced information security expert, you can still use this chapter for concept review purposes.

Understanding Security

The world of data communications has evolved considerably over the years, irrevocably impacting learning methods, business models, human interaction possibilities, and even the dynamics of most day-to-day activity. The networks of today are powerful, enabling individuals and companies to quickly transport data, voice, and video in an integrated fashion, thus providing access from multiple types of devices to all kinds of applications, which may reside anywhere in the globe.

On one hand, virtually limitless use cases are brought to existence by the omnipresent *network of networks*. On the other hand, this almighty global entity, which came to be known as *the Internet,* turned out to be a platform that embeds dangerous characteristics such as user anonymity, the ability to simultaneously control multiple remote computing

devices, and the possibility to automate execution of tasks. Unfortunately, from a technical perspective, this all-encompassing network may be used for both good and evil.

Being aware of the adverse results that may be derived from widespread connectivity, it is natural to look for ways to ensure that only the legitimate or noble usages of the networked systems are allowed. Effective resources that compensate for the absence of natural boundaries in the Internet must be implemented. There should be structured means of defining what the acceptable activities are, from either a productivity or a protection standpoint. Conditional access to networked resources should be put in place, instead of simply providing unrestricted access and naively relying on inherent humankind's goodwill. Dealing with this variety of challenges is what the security practice lends itself to.

But where to start your security learning journey? Well, the first step in solving a problem is recognizing that there is one. The second most effective step is ensuring that you understand what needs to be solved or, in other words, *what is the problem?* And if you are presented with questions for which there may be multiple answers (or multiple choices, as in your certification exam), a good starting point is to eliminate all those options that do not apply. In an attempt to summarize what the practice of security could signify, it is probably easier to begin by defining *what it is not*:

- **Security is neither a product nor a service.** First of all, there is no single product that can act as a "magic black box" that will automatically solve every problem. Moreover, the available capabilities of a given product will be helpful only when they are properly enabled for actual use.
- **Security is not a technology.** Technologies, including those that provide visibility and the ability to block traffic as well as respond to attack situations, may be grouped to form an important *defensive system*. However, the threat matrix is an ever-changing object, meaning that several techniques and tools that have been largely employed on well-known attack scenarios may prove ineffective when facing the newest challenges.
- **Security is not static.** It is not something that you do once and quickly forget. Processes must exist for dealing with planning, implementation, testing, and updating tasks. And all of these items must involve people and discipline.
- **Security is not a check box.** You should know what you are protecting against and, once you determine that, look for resources that can demonstrate true *security effectiveness*.
- **Security is not made only by nominal security elements.** In spite of the existence of dedicated security hardware and software products, security is not limited to them. For example, there are countless contributions that can be given to the overall security process by well-configured network infrastructure devices such as routers.
- **Security is not a beautiful graphical user interface (GUI).** You should always understand what is going on behind the scenes—what is in the brain of the system and not relying blindly, for instance, on reports that state "you are protected."

Now that you've learned what security is not about, it is time to start getting acquainted with what it can be. One general principle that has proved valuable in many fields is to move from global concepts to specifics, and not in the opposite direction. In that sense, if the assigned duty is to protect the relevant digital assets of a particular organization, it is

highly advisable that you understand its vision, mission, objectives, and also the possible competitors. All of these items will be considered in a high-level document known as the *organizational security policy*, which establishes the foundation for all initiatives and tasks pertaining to security.

Among the typical pieces of information that are used to guide policy creation, some deserve special mention:

Business Objectives The main references for policy definition, these are related to the classic "*Why we are here*?" and "*What are we trying to achieve?*" questions that are answered in mission statements or company strategies for a period.

Regulatory Requirements These are specific to the industry sector to which the organization belongs and must be always considered. These requirements are normally able to give a clue to what type of data is valuable in that particular industry.

Risk The acceptable level of risk, from the point of view of senior leadership, should be included in the policy. There can be various categories of risks, such as direct financial loss, improper disclosure of intellectual property, strategic information theft, or damages to the public image of the organization.

Cost/Benefit Analysis This analysis should always be evaluated for the mitigation of the identified risks. The cost/benefit ratio of implementing a certain control must always be taken into consideration, and this calculation involves not only investment in products but also the cost of specialized personnel to make it possible.

A security policy is related to an organization's business strategy and, as such, is normally written using broader terms. To have practical applicability, the general rules and principles it states need to be carefully described in a set of companion documents, which are tactical in nature. The most common of these elements are as follows:

Standards These specify *mandatory* rules, regulations, or activities.

Guidelines These encompass sets of recommendations, reference actions, and operational guides to be considered under circumstances in which standards are not applicable.

Baselines These documents are meant to define the minimum level of security that is required for a given system type.

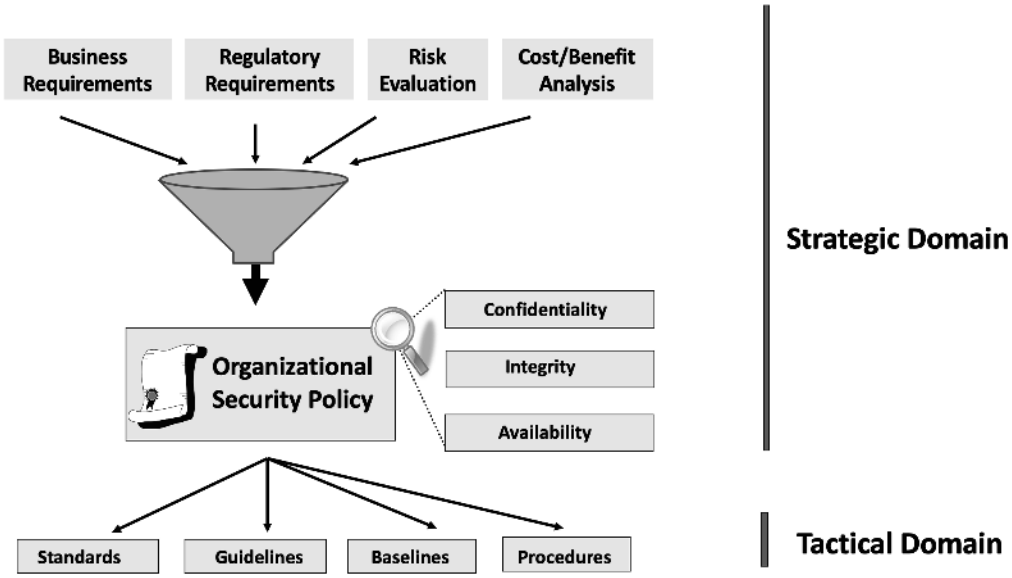
Procedures These include step-by-step instructions for performing specific tasks. They define how policies, standards, and guidelines are implemented within the operating environment.

Figure 1.1 depicts the relationship of the security policy with its companion documents and main sources of information. It also displays some important attributes that must be present in the policy.

You should be aware of several important principles, especially if you are in charge of defending important digital assets. First, you should be aware that *attacks happen*. It does not matter whether or not you detect them. It is not even important whether those attacks have already been successful (even if they haven't, they might be someday—it's just a matter

of time). In dealing with security, it is critical to have an attack-and-defense culture in place so that you are always reflecting on potential exposures and how to mitigate the associated risk.

FIGURE 1.1 Positioning the security policy



You should also notice that every networked element is a potential attack target. This is the case with servers (web, application, database servers, and so on), client devices of any kind, and even infrastructure devices, such as routers, switches, and wireless access points.

Hope is not a strategy. You should make sure your security strategy directly states the access policies and clarifies what types of traffic are permitted and under what conditions. There should be precisely documented network topologies that provide easy understanding of allowed connections, from sources to destinations. You should deploy elements acting as established *policy enforcement points*, instead of assuming that users and devices will behave properly.

Much like onions, security is built in layers. By considering the hypothesis that a certain defense may be circumvented, you should build additional protection layers along the path that leads to your valuable hosts.

At this point of the discussion, some questions may arise, such as: How can you link the macro statements from the overarching security policy to those down-to- earth requirements of configuring a certain access control rule? Or, for instance: What does a particular traffic flow permission have to do with a given business objective of an organization?

To respond to such inquiries, begin by identifying the critical business systems of your organization. What communication protocols are involved in connecting to those systems? What are the inherent risks of having these protocols running in your network? Are there reported vulnerabilities that could be exploited? What are the suitable security measures for risk mitigation?

Basic Security Concepts

Imagine that you have been assigned a mission and that you are truly committed to accomplish it. Before you begin executing the specific tasks that compose the major objective of your journey, you must understand, at a minimum, the following:

- What rules are involved?
- What are the restrictions?
- What is available in your toolkit?
- What kind of help can you count on?
- What are the parameters that indicate that you have succeeded?

Likewise, if your particular mission has something to do with protecting a given computing environment, you must have a solid knowledge not only of the available security building blocks but also of the typical terminology that relates to risk, exposure, threats, and the absence of proper safeguards. The purpose of this section is to provide a reference, within the realm of IT security, which you can revisit while reading the rest of this book.

Vulnerability, Threat, and Security Risk

The concepts of vulnerabilities, threats, and security risks are distinct and yet interrelated:

- A *vulnerability* is a weakness within a computer system that can be exploited to perform unauthorized actions.
- A *threat* is defined by any entity (such as a person or a tool) that can exploit a vulnerability intentionally or by accident. Such an entity is also known as a *threat actor* or *threat agent*.

The concept of *security risk* relates to the probability of a certain vulnerability being exploited by a threat actor. A risk also depends on the value of the digital asset under analysis. For instance, if the same software bug (an example of vulnerability) is present on both a lab virtual machine and a production application server, a higher security risk should be associated with the latter.

Security Countermeasures and Enforcement

Within a computing environment, the mechanisms aimed at risk mitigation are called *security countermeasures* (or *security controls*). They can come in multiple formats, including the following:

- Software patching (to eliminate a previously detected vulnerability).
- Implementation of security capabilities that are specifically designed as defensive resources (thus avoiding vulnerability exploitation). Some examples of such capabilities

will be explored in the "Important Security Solutions and Services" section later in this chapter.

- Verification of user identity before granting access to critical data.

The mere process of defining access policies and their component rules is not sufficient for effective security. You must have a means to ensure that those rules are implemented and obeyed—or, in other words, there must be *enforcement*.

Confidentiality, Integrity, and Availability

The following are foundational attributes that you should consider not only for policy definition but also for evaluation of security effectiveness:

Confidentiality This principle is concerned with preventing unauthorized disclosure of sensitive information and ensuring that a suitable level of privacy is ensured at all stages of data processing. Encryption is a typical example of a technology designed with confidentiality in mind.

Integrity This principle deals with the prevention of unauthorized modification of data and with ensuring information accuracy. Hash message authentication codes, such as HMAC-MD5 and HMAC-SHA (largely employed by the Internet Protocol Security [IPsec] framework), are mathematical functions conceived to provide integrity for the data transmitted in Internet Protocol (IP) packets.

Availability This principle focuses on ensuring reliability and an acceptable level of performance for legitimate users of computing resources. Provisions must be made against eventual failures in the operating environment, which includes the existence of well-designed recovery plans at both the physical and logical levels.

In many publications, the confidentiality, integrity, and availability security principles are also referred as the *CIA triad*.

Accountability and Nonrepudiation

Accountability is an attribute related to a certain individual or organization being held responsible for its actions. The idea is to ensure that all operations performed by systems or processes can be identified and precisely associated with their author.

Nonrepudiation is the property of ensuring that someone cannot deny that they have performed an action in an effort to avoid being held accountable. In the IT security world, repudiation examples are someone denying that a certain system transaction has been carried out or a user denying the authenticity of its own signature.

Authentication, Authorization, and Accounting

Authentication, authorization, and accounting are three security functions that are usually combined to deliver access control services. This interaction inspired the creation of the *AAA architecture*, in which the meaning of each "A" is more easily grasped when associated with the question it was designed to answer:

Authentication Deals with the question "*Who is the user?*" The process to find this answer basically involves extracting user-related information (such as a username and its corresponding password) from an access request to a system and comparing it to a database of previously defined valid users. Certain environments may treat non-registered users as *guests* or *generic users*, thus granting a basic level of access.

Authorization Addresses the question "*What is the user allowed to do?*" This user should have been authenticated before authorization occurs in order to differentiate the access privileges, or *authorization attributes*. The authorization failures that appear on an AAA service report can help characterize improper access attempts.

Accounting Answers the question "*What did the user do?*" Through this process, an accounting client—for instance, a networking device—collects user activity information and sends it to an accounting server (or service in the case of the AWS Cloud). This function serves not only to provide statistics about legitimate use but also to spot unexpected user behavior (in terms of traffic volume or abnormal access hours, for instance).

Visibility and Context

It is certainly much easier to protect your computing systems from the threats that are visible. Fortunately, in today's computing environments, *visibility* is not restricted to what you are able to directly see. Tools and techniques have been specifically developed to provide information about many parameters of packet flows, including the hidden ones.

Another important concept for the current security practice is *context*. Providing context relates to the ability to gather additional pieces of information around the main one so that ambiguity removal is possible before making policy decisions. Here are some examples:

- The same user may be granted different levels of access to corporate resources, depending on the device being used. On a domain-registered personal computer, the user will be provided with full access, whereas on a personal device the same user will have only basic access to applications.
- Access to certain strategic systems may be deemed normal only for a specific time of day or day of the week. Any deviation from what is considered standard may indicate a misuse and should trigger further investigation.
- A certain traffic pattern may be deemed an attack according to the source IP address that it comes from.

Foundational Networking Concepts

Chances are that you may be the security architect in charge of protecting companies that view the AWS Cloud as an interesting disaster recovery option for its critical workloads. You may also be responsible for providing security for companies that are adapting applications so that they can be migrated to the AWS Cloud. Or you may be the security consultant for a cloud-native organization. In any of these scenarios, it is important to keep in mind that, although hosted in a *special network place*, your cloud-based systems will still be reachable through the Internet using standard data communication protocols. Consequently, it is not possible to perform *cloud security* well without a good knowledge of *network security*, which, in turn, is not achievable unless you are familiar with the basics of networking.

This section will visit two network communication models: the *Open Systems Interconnection* (OSI) model as well as what came to be the most prevalent and successful standard for network-based communications, the *TCP/IP protocol stack*. This approach will prove insightful and allow you to quickly locate the layer(s) over which an attack is taking place, thus making it easier to figure out what types of protection mechanisms may prove the most suited.

The OSI Reference Model

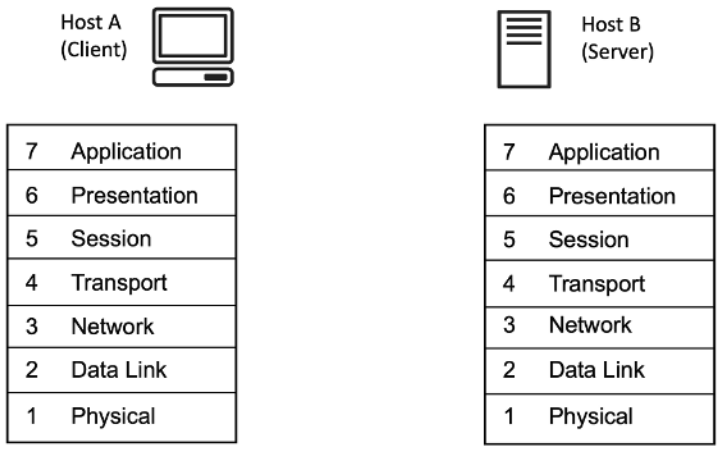
The OSI model was developed by the International Organization for Standardization (ISO, from the Greek word *iso*, which means *equal*) in 1984. This example of a divide-and-conquer approach for explaining network-based communications was aimed at reducing the overall perception of complexity and, undoubtedly, has contributed to generations of professionals and students working in this field. OSI divides the communication system into seven abstract layers, each of which is in charge of a well-defined job, while working in a collaborative way, in order to achieve data transmission between two given systems.

Some of the OSI benefits are listed here:

- It allows the standardization of interfaces among devices, making interoperability possible between diverse systems, even those created by distinct vendors.
- It enables modularity, from engineering and development standpoints, making it possible to design features that belong to a particular layer, without worrying, at least momentarily, about what happens on another.
- It makes it possible to build specialized devices that may act on a specific layer or, eventually, on just some of them.
- It allows more direct isolation of problems, which is useful not only for troubleshooting efforts but also for security planning.

Now that you know the motivation behind the creation of this famous conceptual model, whose hierarchy is illustrated in Figure 1.2, we'll briefly describe the main functions associated with each of the seven layers:

FIGURE 1.2 The OSI model



Physical Layer (Layer 1) The lowest layer is responsible for the physical connection between the devices and is concerned with transmitting raw bits over a communication channel. The main design issues include dealing with mechanical, electrical, optical, and timing interfaces as well as with ensuring that when one side sends a 1 bit, it is accurately received by the other side as a 1 bit, and not as a 0 bit.

Data Link Layer (Layer 2) This layer is in charge of node-to-node delivery of the message in the form of larger and sequential units of data called *frames*. For proper identification of end hosts, it defines a physical addressing scheme, which has only local significance, such as the 48-bit MAC address used by the Ethernet network interface cards (NICs). Some of the issues this layer deals with are error-free delivery, flow control (thus avoiding a fast transmitter from overwhelming a slow receiver), and controlled access to shared media (such as those that allow broadcast transmission).

Network Layer (Layer 3) The main task of this layer concerns *routing* a unit of data (the so-called Layer 3 *packet*) from one given source to a destination that resides on a different network, potentially connected by means of a different Data Link layer technology. This layer introduces the concept of the *logical address*, of which the IP address is the most important example. This addressing paradigm, which treats an individual host as part of a larger logical entity (known as a Layer 3 *subnet*), is what makes global delivery of packets accurate, scalable, and flexible, independently of the Layer 2 media (and the correspondent Layer 2 address of the destination node).

Transport Layer (Layer 4) This layer is aimed at providing reliable message delivery, from the source to the destination host, irrespective of the types, and number, of physical or logical (Layer 3) networks traversed along the path. The Transport layer is also able to confirm (or acknowledge) the successful data transmission and to trigger retransmission if errors are detected. Layer 4 introduces the concepts of source and destination ports, thus allowing multiple service processes to run (and be identified) within the same computing node. The most common transport protocols are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), both of them belonging to the TCP/IP stack, which will be discussed later in the "The TCP/IP Protocol Stack" section.

Session Layer (Layer 5) A session consists of the coordinated exchange of requests and responses between application processes running on endpoint machines. The main functions associated with this layer are session handling (establishment, maintenance, and termination), controlling the dialogue between the two communicating parties (half-duplex and full-duplex transmission), and inserting synchronization control points into the data flow (which makes it possible for a large transfer to be restarted from the point where it was interrupted, rather than retransmitting everything).

Presentation Layer (Layer 6) This layer is sometimes referred to the as the *translation layer*, because it deals with the syntax and semantics of the data being transmitted. This is what makes it possible for devices that employ different data representations to communicate. The data structures being exchanged can be defined in an abstract way, along with a standard encoding to be used over the transmission media.

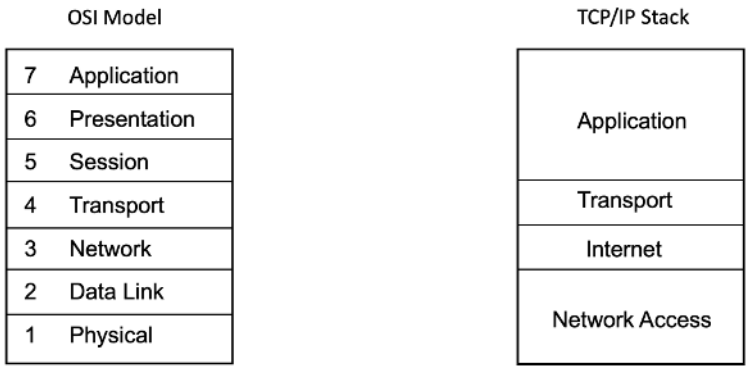
Application Layer (Layer 7) This is the top layer of the OSI reference model and the closest to the end user. Many examples of application protocols are very well known for the typical end user; the most common are the Hypertext Transfer Protocol (HTTP), the File Transfer Protocol (FTP), and the auxiliary Domain Name System (DNS), which acts as the mapping agent between site names and IP addresses before the actual application connection takes place.

The TCP/IP Protocol Stack

Sponsored by the U.S. Department of Defense, the academic network known as the ARPANET (Advanced Research Projects Agency Network) is considered the ancestor of today's Internet. It already employed packet switching (instead of circuit switching) and was the first network to implement the TCP/IP protocol suite.

Even though it is always instructive during the learning process to contrast a certain protocol stack with the OSI model, you should not forget that other suites of protocols were already in use before OSI was established. This was the case of the TCP/IP stack, which survived the test of time and, despite any eventual criticism, became the de facto standard for internetworking. Due to the practical importance of protocols such as IP, TCP, and UDP, we'll provide a dedicated analysis that, although brief, may be useful to you later. Figure 1.3 compares TCP/IP and the OSI layers.

FIGURE 1.3 Comparison between the OSI model and the TCP/IP stack



The *Internet Protocol* (IP) is almost a synonym of the OSI network layer; packet routing is its most relevant task. IP routing deals with the choice of a path over which the IP packets (or datagrams), destined to a particular host, will be sent. Even though some techniques employ additional attributes (the *source IP address*, for example), the classic definition of routing considers the *destination IP address* as the only criterion for path selection. The IP routing function can be divided into four basic activities:

1. **Gathering routing information:** This can be achieved by manual definition of static routes or by using dynamic routing protocols, such as *Open Shortest Path First* (OSPF), *Routing Information Protocol* (RIP), or *Border Gateway Protocol* (BGP).
2. **Building the routing table:** Before installing a path in this table, a router sequentially performs two comparisons: (1) If more than one *equal-length* network prefix is available to a destination, the router will prefer the one with the lowest *administrative distance* (a measure of the trustworthiness among static routes, dynamic routes originated from routing protocols, or a mixture of both), and (2) for two equal-length prefixes that have the same value for the administrative distance parameter, a router will choose the one with the *lowest cost* under the perspective of the particular routing protocol.
3. **Searching for the longest prefix match:** When a packet arrives at the incoming interface, its destination IP address is extracted and compared with the available entries in the routing table. The comparison that results in the longest *bitwise* match for the network mask is selected. The last possibility of finding such a match is to use a *default route*, if one is configured.
4. **Forwarding the packet on the outgoing interface:** When a match happens in step 3, it will point to an entry in the routing table that has a corresponding *outgoing interface*. This last step involves building the appropriate Layer 2 header for this interface.

The TCP/IP model defines two end-to-end transport layer protocols: TCP and UDP. The choice will depend on the requirements of the application protocol being used. TCP is connection-oriented, is reliable, and includes flow control, while UDP is a much simpler option that provides *best effort* delivery of individual packets. UDP is connectionless and unreliable, but nevertheless well suited for real-time traffic (such as voice and video) and other applications that use a client-server communication model with simple request-reply queries.

The types of field included in the header of a communication protocol can tell a lot about its operational capabilities and flexibility. The way a packet with such a header is processed by end hosts or routers along the path can also reveal insightful information about potential protocol vulnerabilities that, if exploited, may lead to security issues. You should therefore have a good understanding of the IP, TCP, and UDP headers. We also recommend that you pay attention to header elements whenever a new protocol is introduced.

Figure 1.4 shows the IPv4 header and Figure 1.5 shows the UDP and TCP headers, with a special reference to the TCP flags field.

FIGURE 1.4 The IPv4 header

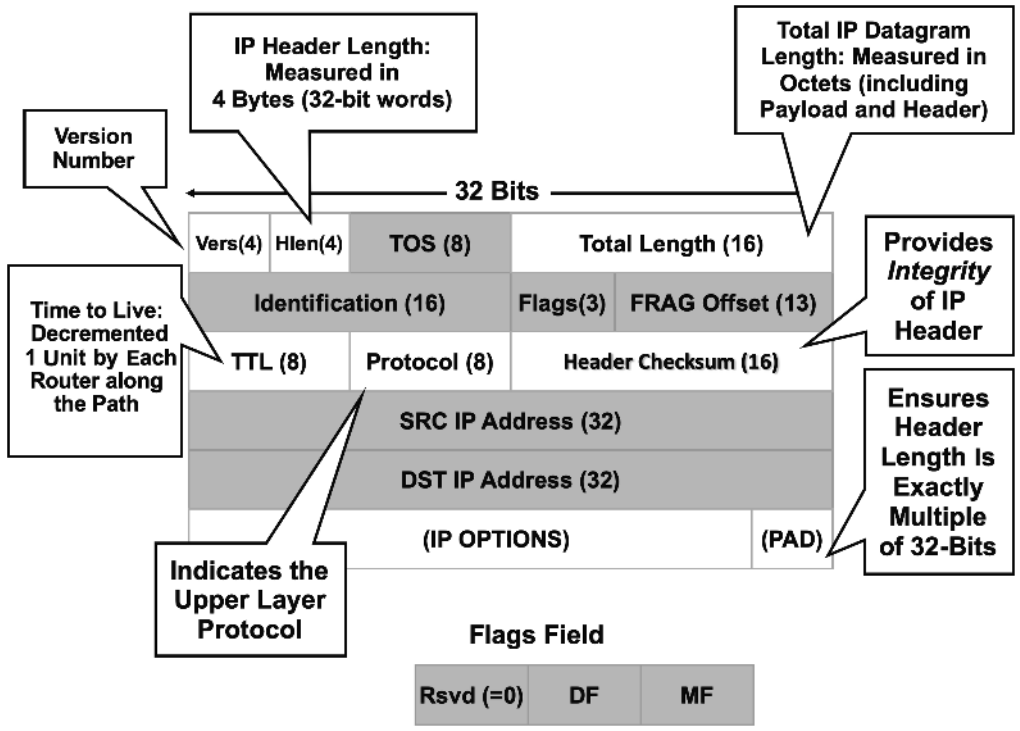
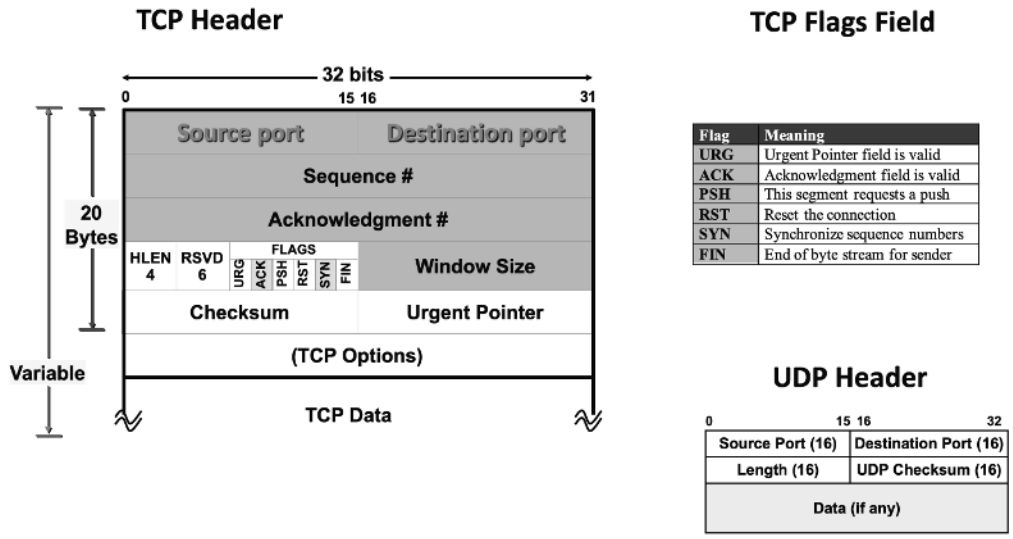


FIGURE 1.5 UDP and TCP headers



Main Classes of Attacks

Even though cyberattacks have always been around, when you compare current threats with those of the Internet's humble beginnings, the difference is essentially related to the intent of the individuals (or groups) carrying out the exploitation attempts. If in the beginning notoriety was the main motivation, the possibility of quick personal profit is what typically attracts more on today's threat landscape. Stealing intellectual property, having access to digital products without paying for it, illegally transferring money from someone else's bank account, and even using the Internet as a weapon for warfare between nations (or specific political groups within a country) are just a few examples of what could be done.

Another relevant challenge faced by the security professionals of today is that new exploit tools are made available daily, whereas the technical knowledge required to operate them is constantly decreasing. Most of them come with examples of use, scripts for attack automation, and sometimes even a GUI, which can make the cyberattack an even simpler task.

This section provides an overall view of the main classes of threats. Before starting the actual discussion, though, a word of warning: the descriptions that follow by no means should be deemed complete, mainly because this is a dynamic subject matter. Furthermore, it is common to find some types of attacks falling within more than one class.

Reconnaissance

Reconnaissance is normally defined as an attack preparation phase rather than an attack class on its own. The underlying goal is to obtain as much information as possible about potential targets without actually interacting with their main application. Internet Control Message Protocol (ICMP) ping sweeps, port scanning (both on UDP and TCP), the observation of a host's behavior under particular conditions (such as exposure to fragmented IP packets), and even *social engineering* are mechanisms that belong to the class of reconnaissance attacks.

Frequently considered harmless (and, as such, overlooked), this practice may be an indicator that attacks are about to happen.

Password Attacks

It is very natural for inside users to have more access rights to the systems of an interconnected organization. Being aware of this characteristic, many attackers leverage techniques that allow them to be authenticated as regular privileged users in such environments. Two possible ways of accomplishing such a goal are

- Creating a new privileged account
- Compromising an existing account and elevating its privileges

Brute-force attacks are those in which all possible combinations of letters are sequentially tried by a program. A *dictionary* attack assumes that users tend to select a common word (typically small) to build their passwords. If the attacker gets access to an encrypted file that contains all the passwords, it will be possible to apply the same encryption to a dictionary of frequently used passwords and compare the results.

Eavesdropping Attacks

Network eavesdropping, also called *sniffing*, is an attack targeted at the *confidentiality* attribute of data. One typical goal here is to obtain valid username and password combinations. In passive eavesdropping, the attacker listens to the message exchange that takes place over the network. This is achievable in many ways: by installing a wiretap; by connecting to shared media, such as an Ethernet hub; or by configuring switch port mirroring (using the attacker's machine as the destination). In active eavesdropping, the attacker tries to produce the mirroring effect but does not need to configure it. Some examples are the exploitation of weaknesses in auxiliary local area network (LAN) protocols such as Dynamic Host Configuration Protocol (DHCP) or Address Resolution Protocol (ARP).

IP Spoofing Attacks

IP spoofing is the act of copying or falsifying a trusted source IP address. It is frequently used as an accessory resource for performing innumerable types of attacks. Typical motivations behind IP spoofing are

- Impersonating a trusted user (or host) and taking advantage of the privileges associated with this trust relationship
- Diverting attention away from the actual attack originator in an attempt to remain undetected
- Casting suspicion on a legitimate host

Man-in-the-Middle Attacks

Man-in-the-middle (MitM) is a broad class of attacks that involve a hacker maliciously inserting a third system into a two-part network conversation or transaction. To achieve this, the attacker establishes independent connections with the victim machines and relays the exchanged messages, thus tricking both victim machines into believing they are directly communicating.

Denial-of-Service Attacks

Since the early days of computer networking, attackers have employed many different techniques to take advantage of system vulnerabilities. Whereas many of the attack categories are focused on compromising the confidentiality or integrity attributes, there are also attempts to affect the availability of services. This last form of doing harm to networked organizations is the practice of *denial of service* (DoS), which induces exhaustion of processing resources (on either connectivity devices or computing hosts), thus keeping legitimate users from accessing the intended applications. DoS attacks can occur in many layers of the OSI reference model, as illustrated in the following examples:

Layer 4 DoS The *TCP SYN flood* is a classic attack that exploits the *three-way handshake* that TCP uses for connection setup. Normally, a TCP three-way handshake consists of a client sending a SYN message to the server, which acknowledges it by sending a SYN-ACK back to the client, causing the client to establish the connection via an ACK message. In this DoS attack, the client never sends the ACK message, creating a substantial number of half-open connections on the server that may exhaust its computing resources.

Layer 3 DoS Earlier examples of Layer 3 DoS attacks were the *Ping of Death* (where a large ICMP Echo message is maliciously sent to a host to cause buffer overflow when it attempts to reassemble the malformed packet), the *Smurf* attack (where an attacker broadcasts an ICMP Echo message using a target host IP as its source and causing

all other hosts to flood this host with ICMP Echo Reply messages), and the *teardrop* attack (where an attacker sends fragmented IP packets to a target host, which may crash when trying to reassemble them).

Layer 2 DoS The Spanning Tree Protocol (STP) was created to remove looped connections in an Ethernet LAN. Switches deploying the same STP version exchange communication during a time interval to decide which links must be blocked to avoid such loops. An attacker may send false messages to initiate STP recalculations that can lead to a LAN environment becoming unavailable.

When a DoS attack is performed in a coordinated fashion, with simultaneous use of multiple source hosts, the term *distributed denial-of-service* (DDoS) is used to describe it.

Malware Attacks

Broadly speaking, *malware* is a software program designed to perform unauthorized actions on computer systems, sometimes reaching the limit of causing irreversible damage to them. Malware enters the network through vulnerabilities and can perform multiple types of malicious actions, including blocking access to network elements; installing additional hostile software on the initial target; propagating to neighboring hosts; creating communication channels with remote control machines in order to perpetuate illegal access; and, eventually, rendering systems unusable. Reflecting their main purpose or the way they act, malware is known under various names:

Virus A specific type of malware that depends on some kind of human action to start its job. A virus replicates itself by inserting its code into other programs and files, or even into a computer's boot sector. Viruses may be distributed through peripheral devices (such as flash drives), email attachments, or infected websites.

Worm This malware type does not require a program to trigger its execution, self-replication, and propagation. Once installed in a victim system, it can create multiple copies of itself and spread through the network, infecting any devices that do not have suitable protection in place.

Trojan Horse This is a destructive program that deceives the user by posing as a genuine application. It is very common for Trojans to create backdoors, thus providing attackers with continuous access to the infected system and allowing, for instance, the theft of information.

Adware This malware class is focused on presenting unwanted advertising to users and is typically bundled with free software or browser toolbars.

Launcher This accessory malware is used to download other malicious software. It is normally used for the initial compromise of a target.

Keylogger This malware is designed to stealthily record everything that is typed on a computer keyboard and transmit the data to a remote agent.

Ransomware This type of malware encrypts all user files on the target machine. After the initial compromise, the victim receives a message offering to restore access in return for the payment of a ransom.

Phishing Attacks

Phishing is the practice of sending fraudulent emails that appear to have come from trusted sources with the objective of obtaining personal information or inducing the victim to perform some action, such as clicking on a hyperlink that will install malware.

Spear phishing is a more advanced technique in which the attackers include information that looks personal and is meaningful for the victims. With this investment in time and special preparation, the received message is more likely to be considered genuine.

Risk Management

Now that you have an understanding of common attack categories and how they operate, it is time to start working on the defense-related activities. Such practices involve, but are not limited to, the following:

- Understanding what each of the security technologies in your protection toolbox can bring to the game
- Knowing your key security personnel and determining the level of security education in your organization
- Designing the security processes to be implemented
- Spreading the security culture inside your team and organization

Important Security Solutions and Services

This section provides a quick review of the main security solutions and services available in the market. You can use this section as a reference that you can return to while you read the remaining chapters of this study guide.

Firewalls

In the context of networking, a *firewall* is a security system aimed at isolating specific areas of the network and delimiting domains of trust. The firewall acts as a sort of *conditional gateway*, specifying the traffic types allowed to go from one domain to another by

means of access control policies. It is important to keep in mind that a firewall is capable of controlling only the traffic that *passes through* it. Therefore, you must have a clear knowledge of the location of clients (*connection initiators*) and servers in the network before defining your policy.

Firewalls are the classic example of a specialized (and dedicated) security device and are pivotal elements in any defense system. Their evolution, through decades of service, has a lot to do with the OSI layers in which they act. Here's a brief review of the various generations of firewalls:

Packet Filters Packet filters focus their access control efforts on static parameters related to the network and transport layers. They are *stateless* in essence, acting only over individual packets instead of connections.

Circuit-Level Proxies (or Generic Proxies) These proxies establish *sessions*, as defined in Layer 5 of the OSI model, to the intended destinations on behalf of requesting source hosts. The classic implementation of this category is the SOCKS5 software.

Application-Level Proxies (or Dedicated Proxies) These proxies understand and interpret the commands within the application protocol they are providing services for. Given their application awareness, they are able to provide functionality such as caching, detailed logging, and user authentication. The trade-offs are the need to develop specific client software for each protected application and their CPU-intensive nature.

Stateful Firewalls These firewalls incorporate the concept of connections and *state* to the original packet filters. Their access control rules act on *groups of packets* that belong to the same connection (or *flow*), rather than on individual packets. This class of firewalls has been widely deployed, not only because their capabilities are much more advanced than those of packet filters but also because they provide much higher performance than dedicated proxies.

Next-Generation Firewalls (NGFWs) NGFWs have been developed using stateful inspection as a departure point. They include the critical capability of identifying applications, regardless of the TCP or UDP service port they use for transport. This is quite relevant because, with the advent of Web 2.0, many applications try to disguise themselves inside HTTP flows (which are always allowed through stateful firewalls), thus avoiding the use of their originally assigned service ports. This modern class of firewalls also includes easy ways of creating user-based rules and integration with auxiliary tools that dynamically analyze the content inside the IP packets, thus helping overall malware detection and prevention efforts. They are also capable of categorizing and filtering uniform resource locators (URLs) and decrypting Secure Sockets Layer (SSL) channels to inspect the content in the communication data payload. Many NGFWs also deploy intrusion-mitigation techniques, which will be explained in the "Intrusion Detection and Intrusion Prevention" section later in this chapter.

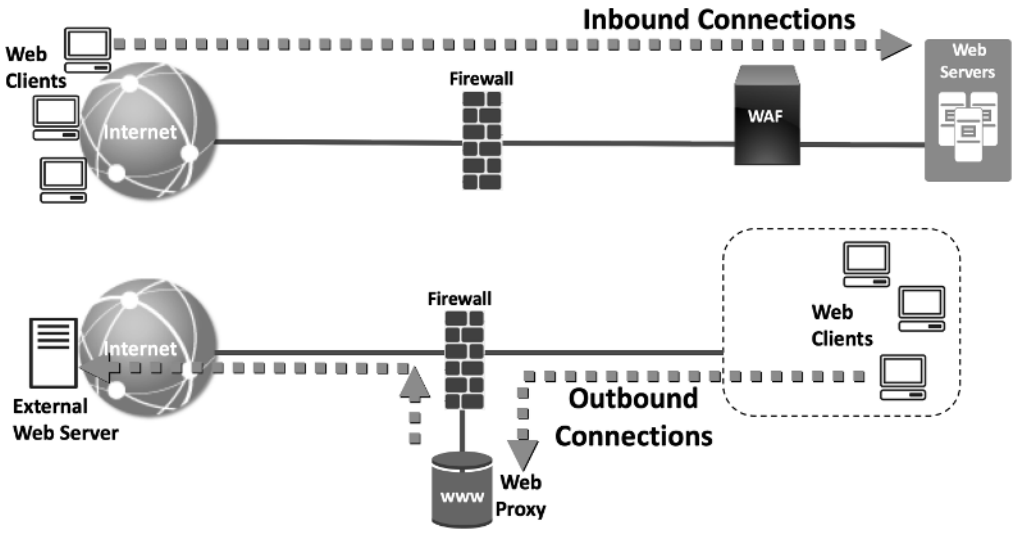
Web Proxies

A *web proxy* (also known as a web gateway) is an important example of an application-level firewall, typically used to control the access of internal corporate users to outside web servers (*outbound* access control). Among many other features, this class of device is capable of blocking malware, enforcing acceptable-use policies, categorizing and filtering URLs, and controlling content based on the reputation of the sites hosting it. The main objective of web proxies is to keep external content requested by internal clients from harming the organization. Nevertheless, given the evolution of NGFWs, this well-known security element is falling into obsolescence.

Web Application Firewalls

The web application firewall (WAF) is a specialized security element that acts as a full-reverse proxy, protecting applications that are accessed through the HTTP protocol. Whereas web proxies protect the client side, WAF devices protect the server side of the connection from application layer attacks. A typical WAF analyzes each HTTP command, thus ensuring that only those actions specified in the security policy can be performed. A reference for WAF action is compensating for the top vulnerabilities identified by OWASP (*Open Web Application Security Project*). Among the most common vulnerabilities are code injection and cross-site scripting. Figure 1.6 contrasts the insertion of web proxies and WAF devices in the network topology.

FIGURE 1.6 Contrasting WAF and a web proxy



You should not confuse the *Application Visibility and Control* (AVC) capabilities of NGFWs, which are focused on controlling outbound user access, with the services provided by WAFs. Instead of replacing one with another, they can actually work in tandem.

Intrusion Detection and Intrusion Prevention

Intrusion-detection and intrusion-prevention technologies provide in-depth inspection capabilities so that the occurrence of malicious traffic can be discovered inside network packets, at either their header or data portion. While *intrusion-detection system* (IDS) devices handle only copies of the packets and are mainly concerned with monitoring and alerting tasks, *intrusion-prevention system* (IPS) solutions are deployed inline in the traffic flow and have the inherent design goal of avoiding actual damage to systems.

IDSs and IPSs can look for well-known attack patterns within packet streams and take an action according to the configured policy. Some typical actions are packet drop, connection block, denying further access to the address that sourced the attack, and sending an alert when an attack indication (*signature*) is spotted.

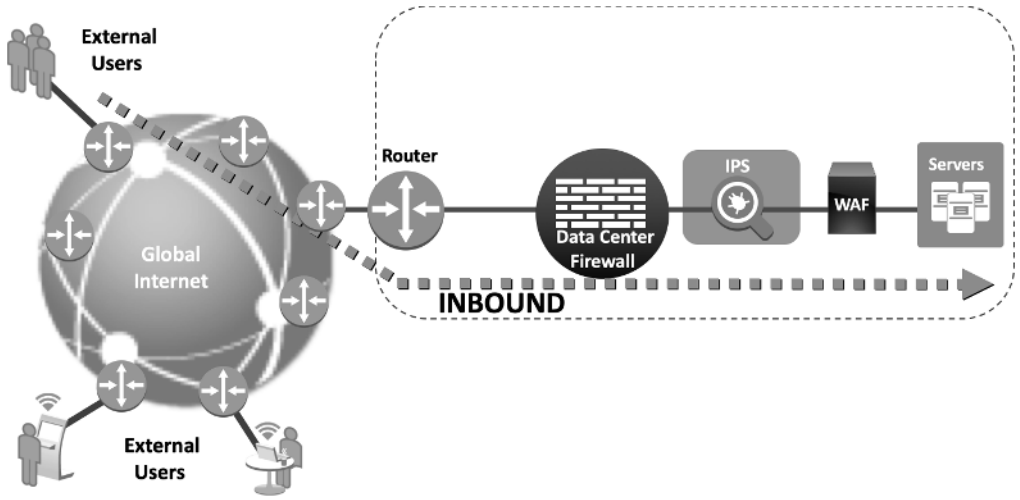
IPSs act as a normal companion to stateful firewalls, mainly for data center protection (inbound traffic). They provide detailed analysis for the connections permitted by firewalls, thus complementing their work. An IPS concentrates most of its analysis at the Network, Transport and Application layers, possibly maintaining state (*stateful pattern matching*) and executing traffic anomaly detection.

There are many possible formats for practical IPS deployment:

- As a dedicated appliance
- As a dedicated hardware (or software) module inside a stateful firewall
- As a resource that is enabled on a per-firewall rule basis on NGFWs

Figure 1.7 displays a typical coordination of protection devices that are well suited for controlling *inbound* access.

FIGURE 1.7 Sample inbound topology



Virtual Private Networks

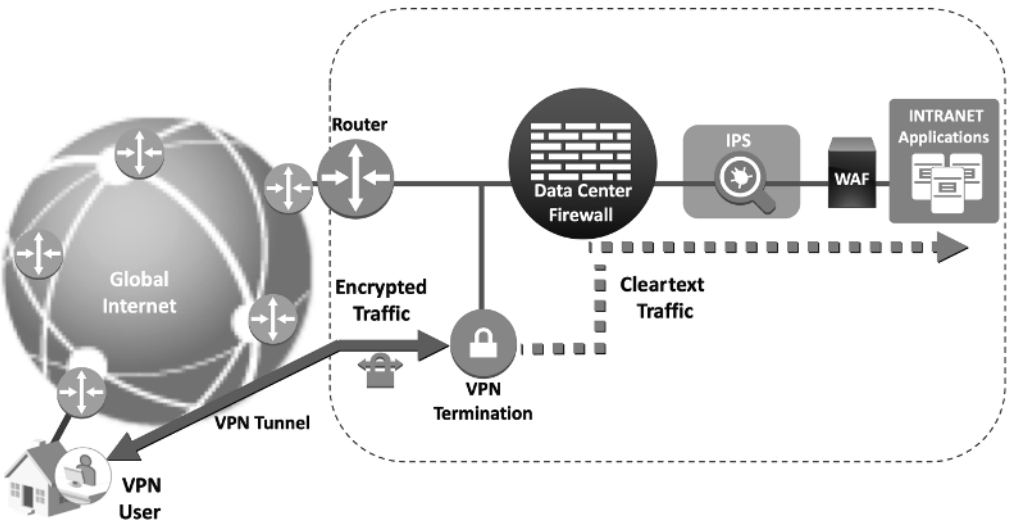
The term *virtual private network* (VPN) is used to refer to technologies that reproduce the characteristics of a private corporate network, even when traffic is being transported over a shared network infrastructure. VPNs were created to provide a secure extension of corporate networks, without requiring a dedicated infrastructure based on expensive WAN circuits.

But security has a broad meaning and may represent very different resources when considered from the standpoint of a particular VPN technology. To further your understanding, we'll summarize the main categories.

The IPsec framework provides answers for questions such as confidentiality, integrity, and authentication of VPN participants and management of cryptographic keys. All of these tasks are accomplished by using standardized protocols and algorithms, a fact that contributed to render IPsec ubiquitous. IPsec supports both client-to-site (or *remote access VPN*) and site-to-site (also known as *LAN-to-LAN*) deployment models. IPsec operates at the Network layer (Layer 3) and, as such, is able to protect native IP protocols (ICMP, for instance) or any application carried over TCP or UDP.

SSL VPN is another remote access VPN technology that does not require a dedicated client. When it was created, SSL VPN was even considered a synonym of *clientless access* (because web browsers are considered a sort of universal client that is available on any networked machine). And this possibility of providing secure remote access, even for those stations that were not managed by corporate IT, sounded appealing for administrators. Not long after the inception of SSL, customers started to request a *client-based* SSL-VPN option that could eventually replace IPsec as the standard remote-access VPN solution. One benefit is the fact that SSL is used everywhere (mainly for HTTP protection) and, as a result, is permitted through firewalls and routers along the path. A classic topology for VPN termination (either IPsec or SSL-based) is shown in Figure 1.8.

FIGURE 1.8 Classic topology for VPN termination



A third common use of the term VPN relates to the *Multiprotocol Label Switching* (MPLS) space, where the service known as *MPLS VPN* was designed to achieve logical network segmentation and optimized routing (in an *any-to-any* or *full-mesh* topology). Although segmentation means providing native traffic isolation among tenants that are transported through the MPLS backbone, there is no provision for the security mechanisms that IPsec deals with (integrity, confidentiality, and authentication). IPsec can be used to add security services for the virtual circuits that characterize MPLS VPN. Although MPLS VPN is primarily used by telecommunications providers to isolate tenant traffic in a shared backbone, we included it here to show you how flexible the term *VPN* is.

Protecting DNS and Using Insights from DNS

The Domain Name System (DNS) is a hierarchical and decentralized directory that maps the assigned hostnames of resources connected to the Internet or other IP networks to their corresponding IP addresses. The motivation is to simplify access to applications by using (ideally) easy-to-remember names, thus avoiding the need for the end user to know the machine-readable IP addresses. This name-to-IP association process, which precedes the actual application request, lies at the heart of the Internet's operation. This official distributed database is constructed in a collaborative way, and the exchange of information pertaining to this process relies on the DNS protocol.

Given the foundational role played by DNS in the Internet architecture, research on two mutually complementary fields has increased:

- Understanding how the protocol operates and its inherent vulnerabilities so that not only message exchange (client-to-server and server-to-server) can be secured but also access can be provided to a directory whose data integrity can be ensured
- Leveraging the information on such a large and dynamic database to keep users from connecting to well-known malicious domains in an attempt to minimize the likelihood that a legitimate user brings harmful content inside the organization

Tools for Vulnerability Analysis and Management

As defined in the earlier section "Basic Security Concepts," a vulnerability is a weakness that can be exploited. It may be seen as a sort of *door that should not be open*. In the context of IT, the *vulnerability assessment* process involves identifying and classifying vulnerabilities in the digital entities that compose the computing environment under analysis (such as end-user machines, operating systems, applications, and key networking elements). After this information-gathering phase, the findings should be presented in a structured fashion so that the corrective actions can be prioritized according to the risk level. Although the grouping of topics may vary among vendors, the typical tasks included in the process may be summarized as follows:

Inventory Compiling *It is very difficult to protect a resource you are not aware of.* Building a hardware and software inventory of the environment is a basic step toward the goal of minimizing the attack surface. The assets that are mission-critical must be identified and grouped according to their business value.

Identification of Vulnerabilities Running scanning tools throughout the inventory will help you create a list of current vulnerabilities (mainly software bugs and misconfigurations) and where they reside.

Definition of Priorities Not all vulnerabilities have the same criticality level and, as such, must not be deemed equally urgent. Considering the business relevance of each potential target will guide the IT staff on prioritizing efforts. If your tool of choice is able to automate the categorization of vulnerabilities and assign some sort of *risk rating*, instead of just presenting a long list of affected entities, it will be much more useful.

Remediation By using the risk-oriented list of the previous step, you can build a road map of actions to be taken so that you reduce the exposure level. If the chosen tool automatically identifies the corrections that must be made, it will be even more useful.

Effectiveness Measurement Knowing that you took the right path, and having some perception of progress, positively affects the morale of the team. "*How much did we reduce the exposure level?*" and "*How does our company compare with our competitors in the same market segment?*" are two simple evaluations that should follow remediation. Again, this can be done manually or, ideally, by selecting a tool that automates it.

Security is not just about technology. It is a continuous process. It involves people. Irrespective of the particular tool you eventually choose and the way it structures and presents the vulnerability data, going through the typical steps we've just described must be a part of your security team's routine. Furthermore, there should be a suitable periodicity.

Correlation of Security Information and Events

Security information and event management (SIEM) solutions are designed to collect security-related logs as well as flow information generated by systems (at the host or application level), networking devices, and dedicated defense elements such as firewalls, IPSs, IDSs, and antivirus software. They work by aggregating, normalizing, and categorizing the received information and then applying intelligence algorithms that allow them to correlate events that refer to particular sessions. Here are some reasons you might employ SIEM solutions:

- To lower the volume of data that must be dealt with by removing ambiguous session information and avoiding the generation of events for legitimate resource use
- To provide real-time insights on security alerts, clearly separating what is meaningful in an attempt to minimize the occurrence of false positives
- To prioritize response actions so that the most critical issues are treated first (according to associated risk level)

Modern SIEM solutions include *artificial intelligence* (AI) and *user behavior analytics* (UBA) so that they can quickly spot deviations from the normal network activity profile for a particular user, which may indicate intentional misuse or, eventually, derive from system

compromise. In a sense, SIEM solutions tend to complement the vulnerability management practice. Whereas the latter deals with the somehow *static* measures of closing the doors that do not need to remain open, the former is more dynamic in nature, providing real-time visibility of what is flowing through the doors.

One critical thing you can do to make your SIEM project successful is to devote time for event filtering so that you can limit the amount of data that will be collected. This approach not only improves performance but also impacts on cost, because most products (and services) are charged based on storage amount or *events per second* (EPS). Another factor in your SIEM selection is to understand *how ready it is* for the correlation tasks pertaining to the systems within your environment. To determine that, ask the following questions:

- Are there native integration agents so that my systems start sending logs to the SIEM?
- Do I need to send every log to the system and then select the security events? Is it possible for the systems of interest to send only security-related information?
- How *plug-and-play* is the solution? What is the average customization time for putting it to work?
- How does the system scale?
- Does the SIEM integrate with incident response tools?

SIEM is a classic example of a solution that has a tremendous potential for alleviating the operational burden of security monitoring. But, to really benefit from it, you will need to invest in training, thus making sure that the team gets acquainted with the capabilities of the tool. It is also advisable to ensure that a detailed documentation of the environment is available and what types of logs and flow data are generated by each of the monitored sources. *This process is not only about acquiring a product or service.*

TLS/SSL Offload and Visibility

The *Secure Sockets Layer* (SSL) protocol was developed to provide services such as data integrity, confidentiality, and peer authentication for application protocols that are carried over TCP. The motivation behind the construction of such a generic layer of security was to avoid the need for embedding security for every application. SSL was updated up to version 3.0, which was deprecated in 2015 and replaced by the standards-based *Transport Layer Security* (TLS) protocol.

Some studies show that TLS/SSL usage is growing continuously. What should be deemed an evolution of the overall security practice brings the collateral effect of lack of visibility, thus allowing malicious traffic to hide inside the encrypted channels. To cope with this new challenge, many security solutions, such as NGFWs, WAFs, and IPSs, started supporting decryption of the TLS streams before going through the analysis activities they were designed for.

By deploying TLS-offload operations, these solutions can provide the following benefits:

- Web servers can be offloaded from any encryption duties (or leverage less-intensive encryption algorithms), which allow them to serve more concurrent clients with a better response time.

- The security solutions can now process and analyze Layers 5–7 information that was originally encrypted (and therefore, reaching application services without any protection).
- The security solutions can centralize public certificates and allow the use of private certificates (which are less expensive and easier to maintain) in the web servers.

TLS Orchestration

Because TLS decryption can be relatively CPU-intensive, it can negatively impact the performance of specialized on-premises appliances that deploy TLS offload (such as NGFWs, WAFs, and IPSs) and introduce latency in their filtering operations. To deal with such a scenario, a solution known as *TLS orchestration* was devised. The idea is to provide a centralized TLS decryption service whose operations include the following:

1. TLS traffic arriving on a centralized device called *TLS orchestrator,* where such traffic is decrypted.
2. The cleartext traffic is dynamically steered to the inspection elements, according to their individual capabilities.
3. After going through the chain of inspection services, the legitimate traffic is sent back to the TLS orchestrator.
4. Traffic is re-encrypted by the orchestrator and delivered to the original destination.

Handling Security Incidents

In the context of information security, an *incident* is defined as a violation (or a threat of violation) of security policies, acceptable-use policies, or standard security practices. Among the actions that may follow from the occurrence of an incident are a negative impact on its reputation, a loss of intellectual property, or unauthorized access to data.

To deal with these sorts of events, you should establish an *incident handling program*. For instance, you should define the meaning of "incident" within your organization. Another important step is to assign an incident response team, with clearly defined responsibilities, among which creating an incident response plan deserves special mention.

As in other domains, incident response may also leverage some tools in order to make the daily tasks of security administrators a bit easier. For example, a new class of products, grouped under the term *security orchestration, automation, and response* (SOAR), has been developed. SIEM solutions are a natural source of information to SOAR systems, which focus on coordinating and automating response actions among multiple protection elements using the concept of playbooks. The ability to create response templates for those incidents that happen often may free a significant amount of time for administrators, thus allowing them to focus on what is different or new.

Structured Malware Protection

As discussed before, there are various categories of computer programs that can be grouped under the name *malware*. There is no absolute strategy for providing protection against malware attacks. Therefore, the next section will introduce an approach for *structured malware protection* as a practical usage example of a security model.

Well-Known Security Frameworks and Models

The concept of security risk is based on the likelihood of a certain vulnerability being exploited, and its respective impact potential, which depends on the value of the digital asset under analysis.

The underlying goal of reducing risk inspired the creation of *security frameworks*, which are published materials that typically include standards, guidelines, sample policies, recommended security safeguards and tools, risk management approaches, relevant technologies, and recognized best practices for protection of certain computing environments.

At a tactical level, the contents inside the framework will translate into *security controls*, which must map to the specific threats a company may be exposed to. A basic design principle that will help in selecting the appropriate controls is to add more layers of defense according to the criticality of the asset being protected.

Many frameworks were developed with a particular goal in mind, such as providing guidance for a given industry segment (according, for instance, to the type of data that is more relevant for that sector, the most valuable systems, and the associated communication protocols). On the other hand, there are examples of frameworks that are meant for more general use. Among the various examples that are in use, some deserve special mention:

Payment Card Industry Data Security Standard (PCI DSS) Created with the goal of increasing the level of protection for issuers of credit cards by requiring that merchants meet minimum levels of security when they process, store, and transmit card holder data.

Health Insurance Portability and Accountability Act (HIPAA) A set of security standards for protecting certain health information that is transferred or held in electronic form.

National Institute for Standards and Technology Cybersecurity Framework (NIST CSF) A publication that results from a collaborative work among industry, academia, and the U.S. government. The framework recognizes that the cybersecurity activities inside an organization should be guided by its business drivers. It also establishes that the overall risk management process should include the risks pertaining to the cybersecurity domain. The CSF assembles standards, guidelines, and practices that have proved effective and may be used by entities belonging to any market segment.

General Data Protection Regulation (GDPR) A set of rules created by the European Union (EU) that requires businesses to protect the personal data and privacy of EU citizens. GDPR not only applies to transactions that occur within the EU members but also governs the transfer of personal data outside the EU. This regulation states that foreign entities willing to conduct business with EU companies also need to demonstrate their compliance with GDPR. This fact motivated the creation of GDPR-like standards outside Europe.

Instead of focusing on security standards that apply to the type of organization you are in charge of protecting, you should become familiar with those principles that can be employed in a broader sense.

Sample Practical Models for Guiding Security Design and Operations

Security controls are implemented to reduce the level of risk to which an organization is exposed. Generically speaking, they are divided into three main categories:

Physical Controls Designed to protect facility, personnel, and material resources. Some examples are locks, fencing, monitoring cameras, and security agents. This type of control is inherently present in the data centers belonging to the large cloud service providers such as AWS.

Logical Controls Many examples of this category of controls are provided in the section "Important Security Solutions and Services" earlier in this chapter.

Administrative Controls Some examples of this class are risk management process, security documentation, and training (not only specific to operations but also to promote overall security awareness within the organization).

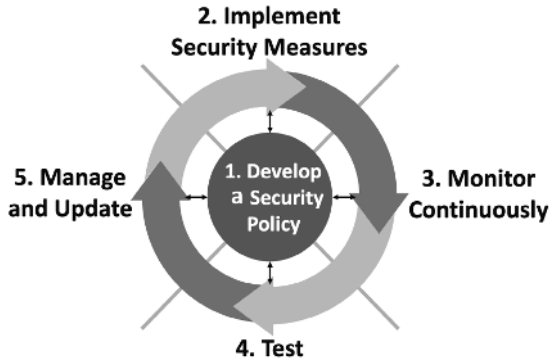
The Security Wheel

Figure 1.9 portrays the *security wheel*, a closed-loop model for security operations that is centered on the foundational concept of security policy, discussed earlier in the "Understanding Security" section. This model recognizes that the security practice has a continuous and cyclical nature and is structured in five basic stages:

1. **Develop a Security Policy:** Start with a high-level policy that clearly establishes and documents the strategic goals that relate to the business drivers or mission of the organization. The policy should refer to the appropriate standards, guidelines, procedures, and baselines that will guide the actual implementation.
2. **Implement Security Measures:** Having defined the assets that need to be protected and the appropriate protection level (according to business relevance), you next deploy the

security controls that will contribute to risk mitigation. Many layers of defense, such as those described in the "Important Security Solutions and Services" section, may be coordinated to provide better security.

FIGURE 1.9 The security wheel



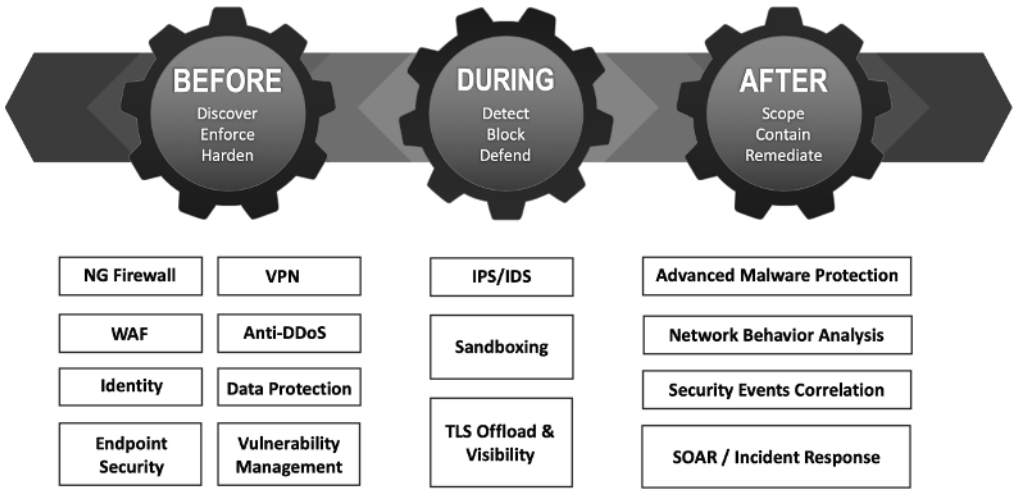
3. **Monitor Continuously:** At this stage, resources such as logging, intrusion detection, and SIEM are employed to spot violations of the access policies.
4. **Test:** Even though you may have invested a lot in protection and monitoring capabilities, *you should not take them for granted*. Systems evolve rapidly and new ways to exploit their weaknesses will be readily available. Vulnerability analysis and management apply well to this stage.
5. **Manage and Update:** Taking as a departure point the feedback that comes from stages 3 and 4, improvements can be applied to stage 2 in the form of new or updated controls. Depending on the specific findings, you may need to review the security policy (for instance, in case it was too permissive or too simple in its original definitions and that reference led to exposure and high risk to the organization's assets).

The Attack Continuum Model

Security is a moving target. No matter the investment of time, effort, and financial resources to protect the organization, new challenges will be faced every day. And attack attempts keep happening all around. Despite the security structure you might have in place, falling victim to a cyberattack is just a matter of time. Of course, the more solid the construction of your layered defense system, the lower the probability of the attack effects being spread throughout the company.

Building on the axiom that *attacks happen*, the attack continuum model associates the security controls with the phase to which they relate most: *before*, *during*, or *after* the attack. Figure 1.10 represents the attack continuum model and suggests some security solutions that may fit each of the model phases:

FIGURE 1.10 The attack continuum model



- **Before:** The tools and methods used here refer mainly to attack *prevention* tasks. The motivation is to have the controls that allow you to minimize the attack surface.
- **During:** *Detection* and monitoring tools, which provide visibility and awareness of what is going on during live packet flow, are the key types of resources for this stage.
- **After:** It is increasingly common that modern attack tools are programmed to have an initial *dormant phase*, with the underlying goal of remaining undetected for a certain period. A good defense implementation needs retrospective security capabilities so that these intentionally delayed threats can be detected (and stopped) before any damage. The protection mechanisms for this phase should identify the point of entry, understand its reach, contain the propagation, and remediate any eventual damage or disruption.

Of course, once a threat is found in the *after the attack* stage, the implicit feedback loop of this model must be used so that the prevention resources of the *before the attack* stage are updated to avoid reinfection.

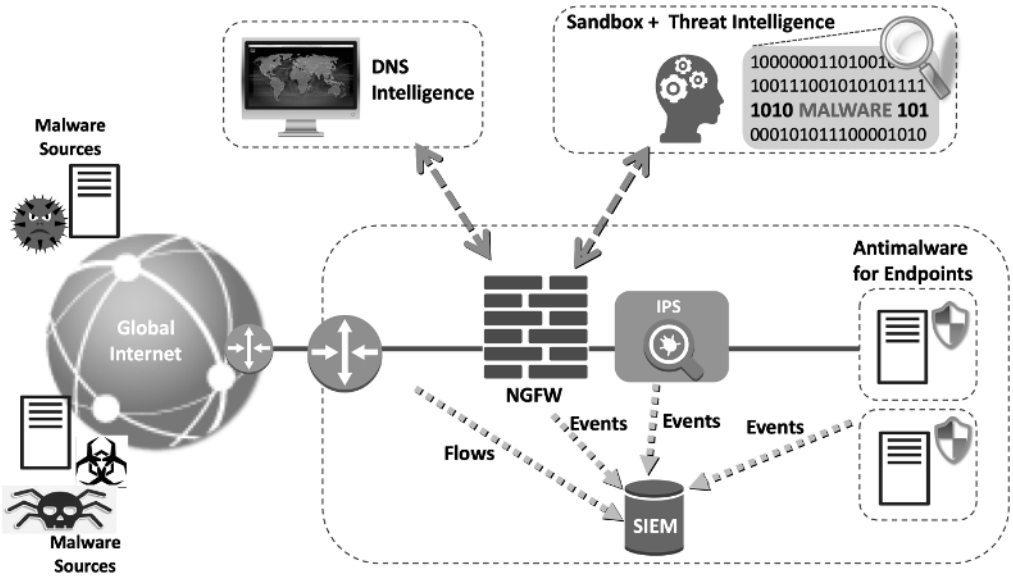
Applying the Attack Continuum Model for Structured Malware Protection

As a practical illustration, a range of protection mechanisms that can be effective in fighting against malware are positioned within the attack continuum model, as shown in Figure 1.11.

- **Before the Attack:** Avoiding the connection of internal users to well-known malicious domains, and to IP addresses that have been used to host domains with bad reputations, proves to be an effective antimalware measure. To achieve that, tools that provide insights about the current DNS infrastructure and help with identifying domain

generation algorithms are very useful. NGFWs and modern IPSs that can block traffic after evaluating the content being transported can also contribute to this effort. For instance, they can block files that are deemed harmful, based on hash information that comes from cyberintelligence sources, thus avoiding their entry into the organization. TLS orchestration systems can play the important auxiliary role of providing visibility for all the inspection devices.

FIGURE 1.11 The attack continuum model applied to malware protection



- **During the Attack:** While the allowed packets are flowing through the network, monitoring tools should be in place. For instance, IPSs should be searching for attack patterns, and files that come through email or web vectors (and whose reputation are not known) can be dynamically evaluated by interaction with *sandbox* solutions, which execute untrusted applications in a separate environment without risking production systems.
- **After the Attack:** Even after all the efforts carried out in the previous phases, some attacks may still be successful. Instead of getting depressed about such an occurrence, consider this an invitation to invest even more in an organized work method. A good start here relates to answering questions such as: Was the attack a result of misconfiguration of one of my protection elements? Was it a result of an unpatched vulnerability? Was the attack totally unknown? In the two first cases, it is a matter of hard and structured work to apply the appropriate corrections. In the last case, a good option is to have antimalware software running on the endpoints, which should be considered

the last line of defense. By looking for abnormal behavior, this kind of element can block malicious activities taking place on end hosts, without need of specific previous knowledge. An access attempt to an external *command and control* source, from a dormant threat on the inside, can also be an indicator of a new threat; DNS insight tools are useful for that purpose. SIEM solutions can also be handy to identify that certain actions, when combined, represent an attack.

The Zero-Trust Model

Once upon a time, in an ideal land, there existed clearly defined perimeters for corporate networks. Branch offices connected through leased lines or IPsec VPNs to the company headquarters, and there was a central point of access to the Internet. The classic firewalls were able to delimit the borders of the organization and establish the access control conditions for traffic to flow between two domains of trust. The simplest logical division was to consider the inside networks as the *trusted domain* and all those IP addresses falling outside the organization as *untrusted*.

Nonetheless, times have changed significantly and computer communication has not only become ubiquitous but also much more complex. The options of access technologies have been multiplied, and each user now possibly owns several devices. Branch offices connect directly to the Internet, thus greatly expanding the frontiers that need supervision and protection. Most companies are moving at least a portion of their workloads to cloud computing environments and, as a result, need to coordinate the security efforts so that they have compatible levels of defense, independently of the data being located on-premises or on the cloud. The extranet connections that, on one hand, simplify business transactions among partners, on the other hand may create shared vulnerabilities, especially if one of the companies is less experienced about security. These are only a few examples of the common challenges faced, on a daily basis, by network administrators.

To adapt to this context, a relatively recent framework, *zero trust,* has established itself as a reference for security design. The model is based on the *principle of least privilege*, which states that organizations should grant the minimal amount of permissions that are strictly necessary for each user or application to work. Therefore, they should not implicitly trust any entity, at any time, no matter if it resides outside or inside the organization domains.

The great philosophy change brought about by zero trust is concerned with avoiding a blind trust of inside hosts and networks. The typical assumption that internal systems are reliable is flawed, because if a single inside host is compromised, lateral movement and associated threat spread will become easy tasks. From a network traffic standpoint, this new security paradigm advocates heavy use of network segmentation, encryption technologies, and identity-based access controls, both at the user and the device levels.

The zero-trust approach is an interesting architecture for cloud environments, which are usually *multitenant* by design and, as such, do not allow you to implicitly trust a shared network. Some relevant controls guided by this model are as follows:

- Using advanced authentication and authorization resources, before granting access, is key for avoiding undesired use of your computing resources or, even worse, unauthorized access to data under your control.
- Promoting granular segmentation of the network, up to the point where you are able to define access control rules down to a single instance level (when necessary), is another key measure associated with the new "shared land" reality. This fine-grained rule-building practice is frequently called *microsegmentation*.
- The use of encryption technologies everywhere is key to bring the confidentiality and integrity attributes to the scene, thus helping you to ensure that data privacy has not been compromised and that data remains unaltered.

Summary

In this chapter, we reviewed general concepts and terminology that are indispensable, both for demonstrating a good level of general education on the security subject matter and for a better understanding of the upcoming chapters.

After exploring basic networking and security definitions, we introduced the main classes of attacks as well as the typical elements that can be used to protect against them. Taking into consideration the critical axiom that *security is not a product*, we reviewed some well-known security frameworks. These particular security models have been selected among the countless options available, not only for being very practical in nature but also because they are complementary to each other.

We discussed the *security wheel*, designed to remind you of the importance of building a security policy and having processes in place to deal with daily security challenges. We next examined the *attack continuum model*, which shows you how to position the available security controls according to their capability of adding value to dealing with a particular attack stage. Finally, we introduced the *zero-trust* paradigm, which is based on the principle of least privilege, thus providing the recommended mindset for creating controls on the shared computing environments that are so characteristic of cloud computing.

Exam Essentials

Understand basic security definitions. A *vulnerability* is a weakness within a computer system that can be exploited to perform unauthorized actions. A *threat* is a potential danger that may derive from the exploitation of a vulnerability. *Security risk* relates to the probability of a certain vulnerability being exploited and the corresponding impact, which

will depend on the value of the digital asset under analysis. *Accountability* is an attribute related to a certain individual or organization being held responsible for its actions. *Nonrepudiation* is ensuring that someone cannot deny an action that has already been performed to avoid attempts of not being accountable.

Understand confidentiality, integrity, and availability. *Confidentiality* is concerned with preventing unauthorized disclosure of sensitive information and with ensuring a suitable level of privacy at all stages of data processing. *Integrity* deals with the prevention of unauthorized modification of data and with ensuring information accuracy. *Availability* focuses on ensuring reliability and an acceptable level of performance for legitimate users of computing resources. Together, these security principles form the *CIA triad*.

Know the components of the AAA architecture. In the AAA architecture, *authentication* deals with the question "Who is the user?"; *authorization* has to do with defining "What is the user allowed to do?"; and *accounting* relates to the question "What did the user do?"

Understand the OSI model. The *OSI model* was developed by the International Organization for Standardization (ISO) and divides data communication into seven layers: Physical, Data Link, Network, Transport, Session, Presentation, and Application.

Understand the TCP/IP stack. The *TCP/IP stack* is a suite of network protocols that was developed to support ARPANET and is largely used on the Internet today. Because it precedes the OSI model, it does not match the OSI's division of seven layers. The main TCP/IP stack protocols are the Internet Protocol (IP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP).

Understand reconnaissance and eavesdropping attacks. *Reconnaissance* attacks aim to obtain as much information as possible about potential targets, without actually interacting with their main application. Such attacks use resources such as ICMP ping sweeps, port scanning, and even social engineering for their objective. *Eavesdropping* attacks analyze network traffic with the objective of obtaining valid username and password combinations or other relevant user information.

Understand common classes of attacks. *Password* attacks attempt to obtain access rights to the systems of a networked organization through brute force (testing all combinations) or dictionary techniques (using common words). *IP spoofing* is the act of copying or falsifying a trusted source IP address. *Man-in-the-middle* attacks involve a hacker inserting itself into a two-part network conversation or transaction as a way of accessing all information. *Denial-of-service* attacks are focused on compromising the availability of services. Finally, *phishing* is the practice of sending fraudulent emails that appear to have come from trusted sources, with the objective of obtaining personal information or inducing the receivers to perform some action, such as clicking on a hyperlink that will install malware.

Be able to define malware. *Malware* is a software program designed to perform unauthorized actions on computer systems, sometimes reaching the limit of causing irreversible damage to them. According to the way malware acts, it may be classified as virus, worm, Trojan horse, adware, launcher, keylogger, or ransomware.

Understand firewalls and their related technologies. A *firewall* is a security system aimed at isolating specific areas of the network and delimiting domains of trust. Throughout their development history, firewalls were classified as packet filters, circuit-level proxies, application-level proxies, stateful firewalls, and next-generation firewalls. A *web proxy* is an example of an application-level firewall that is used to control the access of internal corporate users to outside web servers. A *web application firewall* (WAF) is a specialized security element that protects applications that are accessed through HTTP.

Know the difference between IDS and IPS technologies. Intrusion-detection and intrusion-prevention technologies were conceived to provide in-depth inspection capabilities so that the occurrence of malicious traffic can be determined inside network packets at either their header or data portions. While *intrusion-detection system (IDS)* devices handle only copies of the packets, being mainly concerned with monitoring and alerting tasks, *intrusion-prevention system (IPS)* solutions are deployed inline in the traffic flow and have the inherent design goal of avoiding actual damage to systems.

Be able to define VPNs. *Virtual private networks (VPNs)* refer to technologies that reproduce the characteristics of a private corporate network, even when traffic is being transported over a shared network infrastructure.

Be familiar with SIEM solutions. *Security information and event management (SIEM)* solutions collect security-related logs, as well as flow information, generated by end systems (at the host or the application level), networking devices, and dedicated defense elements, such as firewalls, IPSs, IDSs, and antivirus.

Know the main security frameworks. The *Payment Card Industry Data Security Standard (PCI DSS)* was created with the goal of increasing the level of protection for issuers of credit cards by requiring that merchants meet minimum levels of security when they process, store, and transmit card holder data. The *Health Insurance Portability and Accountability Act (HIPAA)* is a set of security standards for protecting certain health information that is transferred or held in electronic form. The *National Institute for Standards and Technology Cybersecurity Framework (NIST CSF)* is a framework that assembles security standards, guidelines, and practices that have proved effective and may be used by entities belonging to any market segment. The *General Data Protection Regulation (GDPR)* is a set of rules created by the European Union (EU), requiring businesses to protect the personal data and privacy of EU citizens.

Know the main security models. The *security wheel* is a closed-loop practical model that has a continuous and cyclical nature and is structured in five basic stages: develop a security policy, implement security measures, monitor and respond, test, and manage and improve. The *attack continuum model* is based on the axiom that attacks happen and associates the security controls with each phase they relate to the most: before, during, or after the attack. Finally, the *zero-trust* security model is based on the *principle of least privilege*, which states that organizations should grant the minimal amount of permissions that are strictly necessary for each user or application to work. Therefore, they should not implicitly trust any entity, at any time, no matter if it resides outside or inside the organization domains.

Review Questions

1. Read the following statements and choose the correct option:
   - a. A vulnerability is a weakness within a computer system that can be exploited to perform unauthorized actions.
   - b. A security risk is defined as any entity (such as a person or a tool) that can exploit a vulnerability intentionally or by accident.
   - c. A threat relates to the probability of a certain vulnerability being exploited by a threat actor, which will depend on the value of the digital asset under analysis.

   A. Options a, b, and c are correct.
   B. Only option a is correct.
   C. Only option b is correct.
   D. Only option c is correct.

2. Read the following statements and choose the correct option:
   - a. Confidentiality can be addressed through data encryption.
   - b. Integrity can be addressed via hashing algorithms.
   - c. Availability can be addressed with recovery plans.

   A. Options a, b, and c are correct.
   B. Only option a is correct.
   C. Only option b is correct.
   D. Only option c is correct.

3. What better defines "the property of ensuring that someone cannot deny an action that has already been performed so that you can avoid attempts of not being accountable"?
   A. Accountability
   B. Nonrepudiation
   C. Responsibility
   D. Verification
   E. Authentication

4. Which option correctly defines the AAA architecture?
   A. Accountability, authorization, availability
   B. Authentication, authorization, anonymity
   C. Authentication, authorization, accountability
   D. Authentication, authorization, accounting
   E. Authorization, anonymity, accountability

5. Which option represents the seven OSI model layers in the correct order?
   - **A.** Physical, Data Link, Network, Transport, Session, Presentation, and Application
   - **B.** Physical, Data Link, Network, Transport, Session, Application, and Presentation
   - **C.** Physical, Data Link, Routing, Transport, Session, Presentation, and Application
   - **D.** Bit, Frame, Packet, Connection, Session, Coding, and User Interface
   - **E.** Physical, Media Access Control, Network, Transport, Session, Presentation, and Application

6. Which of the following options is not correct?
   - **A.** UDP is part of the TCP/IP stack.
   - **B.** IP can be related to the Network layer of the OSI model.
   - **C.** ICMP, OSPF, and BGP are dynamic routing protocols.
   - **D.** TCP is a connection-oriented and reliable transport protocol.
   - **E.** UDP is a connectionless and unreliable transport protocol.

7. Which well-known class of cyberattacks is focused on affecting the availability of an application, connectivity device, or computing hosts?
   - **A.** Man-in-the-middle
   - **B.** Phishing
   - **C.** Malware
   - **D.** Reconnaissance
   - **E.** Denial of service

8. Which of the following options is not correct?
   - **A.** A firewall is a security system aimed at isolating specific areas of the network and delimiting domains of trust.
   - **B.** A typical WAF analyzes each HTTP command, thus ensuring that only those actions specified on the security policy can be performed.
   - **C.** All VPNs were created to provide a secure extension of corporate networks, without the need of using a dedicated infrastructure but ensuring data confidentiality.
   - **D.** IPsec deals with integrity, confidentiality, and authentication.
   - **E.** SIEM stands for security information and event management.

9. Which security framework was created with the goal of increasing the level of protection for issuers of credit cards?
   - **A.** HIPAA
   - **B.** GDPR
   - **C.** PCI DSS
   - **D.** NIST CSF
   - **E.** CS STAR

10. Which concept guides the zero-trust security model?

- **A.** Develop, implement, monitor continuously, test, and improve
- **B.** Before, during, and after an attack
- **C.** Principle of least privilege
- **D.** In transit and at rest
- **E.** Automation