Chapter 1

Mastering Security Basics

CompTIA Security+ objectives covered in this chapter:

2.2 Given a scenario, use appropriate software tools to assess the security posture of an organization.

- Command line tools (ping, netstat, tracert, arp, ipconfig/ip/ifconfig)

3.2 Given a scenario, implement secure network architecture concepts.

- Segregation/segmentation/isolation (Virtualization)

3.7 Summarize cloud and virtualization concepts.

- Hypervisor (Type I, Type II, Application cells/containers), VM sprawl avoidance, VM escape protection, VDI/VDE

3.8 Explain how resiliency and automation strategies reduce risk.

- Non-persistence (Snapshots, Revert to known state, Rollback to known configuration), Elasticity, Scalability

5.7 Compare and contrast various types of controls.

- Deterrent, Preventive, Detective, Corrective, Compensating, Technical, Administrative, Physical

6.1 Compare and contrast basic concepts of cryptography.

- Common use cases (Supporting confidentiality, Supporting integrity, Supporting obfuscation, Supporting non-repudiation, Resource vs. security constraints)

**

Before you dig into some of the details of security, you should have a solid understanding of core security goals. This chapter introduces many of these core goals to provide a big picture of the concepts, and introduces basic risk concepts. Security controls reduce risks and you'll learn about different security control categories in this chapter. You'll also learn about virtualization and have a chance to do some labs to set up a virtual environment. Finally, this chapter includes some relevant commands that you can run on your primary computer or within a virtual machine.

Understanding Core Security Goals

Security starts with several principles that organizations include as core security goals. These principles drive many security-related decisions at multiple levels. Understanding these basic concepts will help you create a solid foundation in security. Confidentiality, integrity, and availability together form the CIA security triad, a model used to guide security principles within an organization. Each element is important to address in any security program.

What Is a Use Case?

CompTIA includes the term use case in multiple objectives. A use case describes a goal that an organization wants to achieve. Engineers use it in systems analysis and software development to identify and clarify requirements to achieve the goal. A common naming strategy for a use case is in the verb-noun format. As an example, consider a use case named “Place Order.” Different departments within an organization might use it differently, but it can still retain the same name.

In Chapter 7, “Protecting Against Advanced Attacks,” you’ll read about development life- cycle models such as agile. The agile model uses a set of principles that can be shared by cross- functional teams—employees in different departments. Developers can use the steps in the use case to create software to support the goal. The use case can help marketing personnel understand where they need to focus their efforts to motivate the buyer to start the process of placing an order. Billing and Shipping departments use it to understand their responsibilities after the customer places the order.

Imagine that Lisa wants to place an order via an online e-commerce system. The Place Order use case for this might include the following elements:

- **Actors.** Lisa is one of the actors. She might have an account and be a registered user with her shipping and billing information in an existing database. Or, she might be a brand-new customer and her information needs to be collected. Other actors include the billing system that bills her for the order and a fulfillment system that

processes and ships the order.

- **Precondition.** A precondition must occur before the process can start. For example, Lisa needs to select an item to purchase before she can place the order.
- **Trigger.** A trigger starts the use case. In this case, it could be when Lisa clicks on the shopping cart to begin the purchase process.
- **Postcondition.** Postconditions occur after the actor triggers the process. In this case, Lisa's order will be put into the system after she completes the purchase. She'll receive an acknowledgment for her order, the Billing department may take additional steps to bill her (if she wasn't billed during the purchase process), and the Shipping department will take steps to ship the product.
- **Normal flow.** A use case will typically list each of the steps in a specific order. In this example, you might see a dozen steps that start when Lisa picks an item to order and end when she completes the order and exits the purchase system.
- **Alternate flow.** All purchases won't be the same. For example, instead of using existing billing and shipping information, Lisa might want to use a different credit card or a different shipping address. It's also possible for Lisa to change her mind and abandon the process before completing the purchase or even cancel the purchase after she completes the process.

Note that these are not the only possible elements in a use case. There are many more. However, you don't need to be an expert in agile to understand the overall concept of a use case. If you want to be an expert in agile, you can pursue the Project Management Institute Agile Certified Practitioner (PMI-ACP) certification. It requires at least 2,000 hours of general project experience working on teams, 1,500 hours working on agile project teams or with agile methodologies, and at least 21 hours of in-classroom training. The point is that passing the CompTIA Security+ exam doesn't require you to have the PMI-ACP or to know all the elements of use cases. It does require you to understand the basic concept of a use case.

The following sections discuss some common use cases related to supporting confidentiality, integrity, availability, authentication, obfuscation, and non-repudiation.

Ensure Confidentiality

A common use case that any organization has is to support confidentiality. ***Confidentiality*** prevents the unauthorized disclosure of data. In other words, authorized personnel can access the data, but unauthorized personnel cannot access the data. You can ensure confidentiality using several different methods discussed in the following sections.

Encryption

Encryption scrambles data to make it unreadable by unauthorized personnel. Authorized personnel can decrypt the data to access it, but encryption techniques make it extremely difficult for unauthorized personnel to access encrypted data. Chapter 10, "Understanding Cryptography and PKI," covers encryption in much more depth, including commonly used encryption algorithms like Advanced Encryption Standard (AES).

As an example, imagine you need to transmit Personally Identifiable Information (PII), such as medical information or credit card data via email. You wouldn't want any unauthorized personnel to access this data, but once you click Send, you're no longer in control of the data. However, if you encrypt the email before you send it, you protect the confidentiality of the data.

Access Controls

Identification, authentication, and authorization combined provide access controls and help ensure that only authorized personnel can access data. Imagine that you want to grant Maggie access to some data, but you don't want Homer to be able to access the same data. You use access controls to grant and restrict access. The following bullets introduce key elements of access controls:

- **Identification.** Users claim an identity with a unique username. For example, both Maggie and Homer have separate user accounts identified with unique usernames. When Maggie uses her account, she is claiming the identity of her account.
- **Authentication.** Users prove their identity with authentication, such as with a password. For example, Maggie knows her password, but no one else should know it. When she logs on to her account with

her username and password, she is claiming the identity of her account and proving her identity with the password.

- **Authorization.** Next, you can grant or restrict access to resources using an authorization method, such as permissions. For example, you can grant Maggie's account full access to some files and folders. Similarly, you can ensure that Homer doesn't have any permissions to access the data.

Chapter 2, "Understanding Identity and Access Management," covers these topics in more depth.

Steganography and Obfuscation

A third method you can use for confidentiality is ***steganography***. Chapter 10 covers steganography in more depth, but as an introduction, it is the practice of hiding data within data. It obscures the data and can be used in a use case to support obfuscation.

Obfuscation methods attempt to make something unclear or difficult to understand. Within the context of information technology (IT) security, it's called security by obscurity or security through obscurity. It's worth noting that most security experts reject security through obscurity as a reliable method of maintaining security.

Many people refer to steganography as hiding data in plain sight. For example, you can embed a hidden message in an image by modifying certain bits within the file. If other people look at the file, they won't notice anything. However, if other people know what to look for, they will be able to retrieve the message.

As a simpler example, you can add a text file to an image file without the use of any special tools other than WinRAR and the Windows command line. If you're interested in seeing how to do this, check out the Steganography Lab in the online exercises for this book at *http://gcgapremium.com/501labs/.*

Remember this

Confidentiality ensures that data is only viewable by authorized users. The best way to protect the confidentiality of data is by encrypting it. This includes any type of data, such as PII, data in databases, and data on mobile devices. Access controls help protect

confidentiality by restricting access. Steganography helps provide confidentiality by hiding data, such as hiding text files within an image file.

Provide Integrity

Another common use case is to support integrity. ***Integrity*** provides assurances that data has not changed. This includes ensuring that no one has modified, tampered with, or corrupted the data. Ideally, only authorized users modify data. However, there are times when unauthorized or unintended changes occur. This can be from unauthorized users, from malicious software (malware), and through system and human errors. When this occurs, the data has lost integrity.

Hashing

You can use hashing techniques to enforce integrity. Chapter 10 discusses the relevant hashing algorithms, such as Message Digest 5 (MD5), Secure Hash Algorithm (SHA), and Hash- based Message Authentication Code (HMAC). Briefly, a ***hash*** is simply a number created by executing a hashing algorithm against data, such as a file or message. If the data never changes, the resulting hash will always be the same. By comparing hashes created at two different times, you can determine if the original data is still the same. If the hashes are the same, the data is the same. If the hashes are different, the data has changed.

For example, imagine Homer is sending a message to Marge and they both want assurances that the message retained integrity. Homer's message is, "The price is $19.99." He creates a hash of this message. For simplicity's sake, imagine the hash is 123. He then sends both the message and the hash to Marge.

Acronyms (Sidebar)

Don't you just love these acronyms? MD5, SHA, HMAC. There are actually three different meanings of MAC within the context of CompTIA Security+:

1. Media access control (MAC) addresses are the physical addresses assigned to network interface cards

(NICs).

2. The mandatory access control (MAC) model is one of several access control models discussed in Chapter 2.
3. Message authentication code (MAC) provides integrity similar to how a hash is used.

If you're having trouble keeping them all straight, don't feel alone. Appendix A, "Glossary," spells out—and lists brief descriptions—for relevant acronyms used in this book.

Marge receives both the message and the hash. She can calculate the hash on the received message and compare her hash with the hash that Homer sent. If the hash of the received message is 123 (the same as the hash of the sent message), she knows the message hasn't lost data integrity. However, if the hash of the received message is something different, such as 456, then she knows that the message she received is not the same as the message that Homer sent. Data integrity has been lost.

Hashing doesn't tell you what modified the message. It only tells you that the message has been modified. This implies that the information should not be trusted as valid.

You can use hashes with messages, such as email, and any other type of data files. Some email programs use a message authentication code (MAC) instead of a hash to verify integrity, but the underlying concept works the same way.

You can also use hashing techniques to verify that integrity is maintained when files are downloaded or transferred. Some programs can automatically check hashes and determine if a file loses even a single bit during the download process. The program performing the download will detect it by comparing the source hash with the destination hash. If a program detects that the hashes are different, it knows that integrity has been lost and reports the problem to the user.

As another example, a web site administrator can calculate and post the hash of a file on a web site. Users can manually calculate the hash of the file after downloading it and compare the calculated hash with the posted hash. If a virus infects a file on the web server, the hash of the infected file would be different from the hash of the original file (and the hash posted on the web

site). You can use freeware such as *md5sum.exe* to calculate MD5 hashes. If you want to see this in action, check out the Creating and Comparing Hashes Lab in the online exercises for this book at *http://gcgapremium.com/501labs/.*

It's also possible to lose data integrity through human error. For example, if a database administrator needs to modify a significant amount of data in a database, the administrator can write a script to perform a bulk update. However, if the script is faulty, it can corrupt the database, resulting in a loss of integrity.

Two key concepts related to integrity are:

- **Integrity provides assurances that data has not been modified, tampered with, or corrupted.** Loss of integrity indicates the data is different. Unauthorized users can change data, or the changes can occur through system or human errors.
- **Hashing verifies integrity.** A hash is simply a numeric value created by executing a hashing algorithm against a message or file. Hashes are created at the source and destination or at two different times (such as on the first and fifteenth of the month). If the hashes are the same, integrity is maintained. If the two hashes are different, data integrity has been lost.

Remember this

Integrity verifies that data has not been modified. Loss of integrity can occur through unauthorized or unintended changes. Hashing algorithms, such as MD5, SHA-1, and HMAC, calculate hashes to verify integrity. A hash is simply a number created by applying the algorithm to a file or message at different times. By comparing the hashes, you can verify integrity has been maintained.

Digital Signatures, Certificates, and Non-Repudiation

You can also use digital signatures for integrity. Chapter 10 covers digital signatures in more depth, but as an introduction, a ***digital signature*** is similar in concept to a handwritten signature. Imagine you sign a one-page contract. Anyone can look at the contract later, see your signature, and know it is the same contract. It isn't possible for other people to modify the words in the contract unless they can reproduce your signature, which isn't easy to do.

It's common to use digital signatures with email. For example, imagine

that Lisa wants to send an email to Bart. She can attach a digital signature to the email and when Bart receives it, the digital signature provides assurances to him that the email has not been modified.

A digital signature also provides authentication. In other words, if the digital signature arrives intact, it authenticates the sender. Bart knows that Lisa sent it.

Authentication from the digital signature prevents attackers from impersonating others and sending malicious emails. For example, an attacker could make an email look like it came from Lisa and include a link to a malicious web site urging Bart to click it. Without a digital signature, Bart might be fooled into thinking that Lisa sent it and click the link. This might result in Bart inadvertently downloading malware onto his system.

Remember this

Integrity verifies that data has not been modified. Loss of integrity can occur through unauthorized or unintended changes. Hashing algorithms, such as MD5, SHA-1, and HMAC, calculate hashes to verify integrity. A hash is simply a number created by applying the algorithm to a file or message at different times. By comparing the hashes, you can verify integrity has been maintained.

Digital signatures also provide ***non-repudiation***. In other words, Lisa cannot later deny sending the email because the digital signature proves she did. Another way of thinking about non-repudiation is with credit cards. If you buy something with a credit card and sign the receipt, you can't later deny making the purchase. If you do, the store will use your signature to repudiate your claim. In other words, they use your signature for non-repudiation.

Security systems implement non-repudiation methods in other ways beyond digital signatures. Another example is with audit logs that record details such as who, what, when, and where. Imagine Bart logged on to a computer with his username and password, and then deleted several important files. If the audit log recorded these actions, it provides non-repudiation. Bart cannot believably deny he deleted the files.

Digital signatures require the use of certificates and a Public Key Infrastructure (PKI). Certificates include keys used for encryption and the PKI provides the means to create, manage, and distribute certificates. Obviously, there's much more to certificates and a PKI, but there isn't room in this chapter

for more than an introduction. Feel free to jump ahead to Chapter 10 if you want to learn more right now.

Remember this

Digital signatures can verify the integrity of emails and files and they also provide authentication and non-repudiation. Digital signatures require certificates.

Increase Availability

Availability indicates that data and services are available when needed. For some organizations, this simply means that the data and services must be available between 8:00 a.m. and 5:00 p.m., Monday through Friday. For other organizations, this means they must be available 24 hours a day, 7 days a week, 365 days a year.

Organizations commonly implement redundancy and fault-tolerant methods to ensure high levels of availability for key systems. Additionally, organizations ensure systems stay up to date with current patches to ensure that software bugs don't affect their availability.

Redundancy and Fault Tolerance

Redundancy adds duplication to critical systems and provides fault tolerance. If a critical component has a fault, the duplication provided by the redundancy allows the service to continue without interruption. In other words, a system with fault tolerance can suffer a fault, but it can tolerate it and continue to operate.

A common goal of fault tolerance and redundancy techniques is to remove each single point of failure (SPOF). If an SPOF fails, the entire system can fail. For example, if a server has a single drive, the drive is an SPOF because its failure takes down the server.

Chapter 9, "Implementing Controls to Protect Assets," covers many fault-tolerance and redundancy techniques in more depth. As an introduction, here are some common examples:

- **Disk redundancies.** Fault-tolerant disks, such as RAID-1 (mirroring), RAID-5 (striping with parity), and RAID-10 (striping with a mirror), allow a system to continue to operate even if a disk

fails.

- **Server redundancies.** Failover clusters include redundant servers and ensure a service will continue to operate, even if a server fails. In a failover cluster, the service switches from the failed server in a cluster to an operational server in the same cluster. Virtualization can also increase availability of servers by reducing unplanned downtime. The "Implementing Virtualization" section later in this chapter covers virtualization in more depth.
- **Load balancing.** Load balancing uses multiple servers to support a single service, such as a high-volume web site. It can increase the availability of web sites and web-based applications.
- **Site redundancies.** If a site can no longer function due to a disaster, such as a fire, flood, hurricane, or earthquake, the organization can move critical systems to an alternate site. The alternate site can be a hot site (ready and available 24/7), a cold site (a location where equipment, data, and personnel can be moved to when needed), or a warm site (a compromise between a hot site and cold site).
- **Backups.** If personnel back up important data, they can restore it if the original data is lost. Data can be lost due to corruption, deletion, application errors, human error, and even hungry gremlins that just randomly decide to eat your data. Without data backups, data is lost forever after any one of these incidents.
- **Alternate power.** Uninterruptible power supplies (UPSs) and power generators can provide power to key systems even if commercial power fails.
- **Cooling systems.** Heating, ventilation, and air conditioning (HVAC) systems improve the availability of systems by reducing outages from overheating.

Remember this

Availability ensures that systems are up and operational when needed and often addresses single points of failure. You can increase availability by adding fault tolerance and redundancies, such as RAID, failover clusters, backups, and generators. HVAC systems also increase availability.

Patching

Another method of ensuring systems stay available is with patching. Software bugs cause a wide range of problems, including security issues and even random crashes. When software vendors discover the bugs, they develop and release code that patches or resolves these problems. Organizations commonly implement patch management programs to ensure that systems stay up to date with current patches. Chapter 5, "Securing Hosts and Data," covers patching and patch management in greater depth.

Resource Versus Security Constraints

Organizations frequently need to balance resource availability with security constraints. Consider using encryption to maintain the confidentiality of data. If this is possible, why not just encrypt all the data? The reason is that encryption consumes resources.

As an example, the above paragraph is about 260 characters. Encrypted, it is about 360 characters. That's an increase of about 40 percent, which is typical with many encryption methods. If a company decides to encrypt all data, it means that it will need approximately 40 percent more disk space to store the data. Additionally, when processing the data, it consumes more memory. Last, it takes additional processing time and processing power to encrypt and decrypt the data.

Security experts might say the cost for additional resources is worth it, but executives looking to increase the value of the company don't. Instead, executives have a responsibility to minimize costs without sacrificing security. They do this by looking for the best balance between resource costs and security needs.

Introducing Basic Risk Concepts

One of the basic goals of implementing IT security is to reduce risk. Because risk is so important and so many chapters refer to elements of risk, it's worth providing a short introduction here.

Risk is the possibility or likelihood of a threat exploiting a vulnerability resulting in a loss. A ***threat*** is any circumstance or event that has the potential to compromise confidentiality, integrity, or availability. A ***vulnerability*** is a weakness. It can be a weakness in the hardware, the software, the

configuration, or even the users operating the system.

If a threat (such as an attacker) exploits a vulnerability, it can result in a security incident. A ***security incident*** is an adverse event or series of events that can negatively affect the confidentiality, integrity, or availability of an organization's information technology (IT) systems and data. This includes intentional attacks, malicious software (malware) infections, accidental data loss, and much more.

Threats can come from inside an organization, such as from a disgruntled employee or a malicious insider. They can come from outside the organization, such as from an attacker anywhere in the world with access to the Internet. Threats can be natural, such as hurricanes, tsunamis, or tornadoes, or manmade, such as malware written by a criminal. Threats can be intentional, such as from attackers, or accidental, such as from employee mistakes or system errors.

Reducing risk is also known as risk mitigation. ***Risk mitigation*** reduces the chances that a threat will exploit a vulnerability. You reduce risks by implementing controls (also called countermeasures and safeguards), and many of the actions described throughout this book are different types of controls. You can't prevent most threats. For example, you can't stop a tornado or prevent a criminal from writing malware. However, you can reduce risk by reducing vulnerabilities to the threat, or by reducing the impact of the threat.

For example, access controls (starting with authentication) ensure that only authorized personnel have access to specific areas, systems, or data. If employees do become disgruntled and want to cause harm, access controls reduce the amount of potential harm by reducing what they can access. If a natural disaster hits, business continuity and disaster recovery plans help reduce the impact. Similarly, antivirus software prevents the impact of any malware by intercepting it before it causes any harm.

Remember this

Risk is the likelihood that a threat will exploit a vulnerability. Risk mitigation reduces the chances that a threat will exploit a vulnerability, or reduces the impact of the risk, by implementing security controls.

Understanding Control Types

There are hundreds, perhaps thousands, of security controls that organizations can implement to reduce risk. The good news is that you don't need to be an expert on all of the possible security controls to pass the CompTIA Security+ exam. However, you do need to have a basic understanding of control types.

CompTIA lists the following control types in the objectives:

- Technical controls use technology.
- Administrative controls use administrative or management methods.
- Physical controls refer to controls you can physically touch.
- Preventive controls attempt to prevent an incident from occurring.
- Detective controls attempt to detect incidents after they have occurred.
- Corrective controls attempt to reverse the impact of an incident.
- Deterrent controls attempt to discourage individuals from causing an incident.
- Compensating controls are alternative controls used when a primary control is not feasible.

The first three control types in the list (technical, administrative, and physical) refer to how the security controls are implemented. The remaining control types refer to the goals of the security control.

Remember this

Most security controls can be classified as technical (implemented with technology), administrative (implemented using administrative or management methods), or physical (items you can touch).

Technical Controls

Technical controls use technology to reduce vulnerabilities. An administrator installs and configures a technical control, and the technical control then provides the protection automatically. Throughout this book, you'll come across several examples of technical controls. The following list provides

a few examples:

- **Encryption.** Encryption is a strong technical control used to protect the confidentiality of data. This includes data transferred over a network and data stored on devices, such as servers, desktop computers, and mobile devices.
- **Antivirus software.** Once installed, the antivirus software provides protection against malware infection. Chapter 6, "Comparing Threats, Vulnerabilities, and Common Attacks," covers malware and antivirus software in more depth.
- **Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs).** IDSs and IPSs can monitor a network or host for intrusions and provide ongoing protection against various threats. Chapter 4, "Securing Your Network," covers different types of IDSs and IPSs.
- **Firewalls.** Network firewalls restrict network traffic going in and out of a network. Chapter 3, "Exploring Network Technologies and Tools," covers firewalls in more depth.
- **Least privilege.** The principle of least privilege specifies that individuals or processes are granted only the privileges they need to perform their assigned tasks or functions, but no more. Privileges are a combination of rights and permissions.

Remember this

Technical controls use technology to reduce vulnerabilities. Some examples include encryption, antivirus software, IDSs, IPSs, firewalls, and the principle of least privilege. Technical physical security and environmental controls include motion detectors and fire suppression systems.

Administrative Controls

Administrative controls use methods mandated by organizational policies or other guidelines. For example, management may require personnel to periodically complete assessments and tests to reduce and manage risk. Many of these assessments provide an ongoing review of an organization's risk management capabilities. Some common administrative controls are:

- **Risk assessments.** Risk assessments help quantify and qualify

risks within an organization so that the organization can focus on the serious risks. For example, a quantitative risk assessment uses cost and asset values to quantify risks based on monetary values. A qualitative risk assessment uses judgments to categorize risks based on probability and impact.

- **Vulnerability assessments.** A vulnerability assessment attempts to discover current vulnerabilities or weaknesses. When necessary, an organization implements additional controls to reduce the risk from these vulnerabilities.
- **Penetration tests.** These go a step further than a vulnerability assessment by attempting to exploit vulnerabilities. For example, a vulnerability assessment might discover a server isn't kept up to date with current patches, making it vulnerable to some attacks. A penetration test would attempt to compromise the server by exploiting one or more of the unpatched vulnerabilities.

Chapter 8, "Using Risk Management Tools," covers these assessments and tests in more depth. Some administrative controls focus on physical security and the environment. For example, an access list identifies individuals allowed into a secured area. Guards then verify individuals are on the access list before allowing them in.

Many administrative controls are also known as operational or management controls. They help ensure that day-to-day operations of an organization comply with the organization's overall security plan. People (not technology) implement these controls. Operational controls include the following families:

- **Awareness and training.** The importance of training to reduce risks cannot be overstated. Training helps users maintain password security, follow a clean desk policy, understand threats such as phishing and malware, and much more.
- **Configuration and change management.** Configuration management often uses baselines to ensure that systems start in a secure, hardened state. Change management helps ensure that changes don't result in unintended configuration errors. Chapter 5 covers configuration and change management in more detail.
- **Contingency planning.** Chapter 9 presents several different methods that help an organization plan and prepare for potential system outages. The goal is to reduce the overall impact on the

organization if an outage occurs.

- **Media protection.** Media includes physical media such as USB flash drives, external and internal drives, and backup tapes.
- **Physical and environmental protection.** This includes physical controls, such as cameras and door locks, and environmental controls, such as heating and ventilation systems.

Physical Controls

Physical controls are any controls that you can physically touch. Some examples include lighting, signs, fences, security guards, and more. CompTIA has placed a lot more emphasis on physical security controls, including environmental controls such as hot and cold aisles and fire suppression. You'll see these covered in more depth in Chapter 9.

However, it's important to realize that many of these are also technical controls. For example, a fire suppression system is a physical security control because you can touch it. However, it's also a technical control because it uses technologies to detect, suppress, or extinguish fires.

Control Goals

Technical and administrative controls categorize the controls based on how they are implemented. Another way of classifying security controls is based on their goals in relationship to security incidents. Some common classifications are preventive, detective, corrective, deterrent, and compensating. The following sections describe them in more depth.

NIST and SP 800 Documents (Sidebar)

The National Institute of Standards and Technology (***NIST***) is a part of the U.S. Department of Commerce, and it includes a Computer Security Division hosting the Information Technology Laboratory (ITL). The ITL publishes Special Publications (SPs) in the 800 series that are of general interest to the computer security community.

Many IT security professionals use these documents as references to design secure IT systems and networks.

Additionally, many security-related certifications (beyond the CompTIA Security+ certification) also reference the SP 800 documents both directly and indirectly.

SP 800-53 Revision 4, "Security and Privacy Controls for Federal Information Systems and Organizations," includes a wealth of information on security controls. It includes three relatively short chapters introducing security controls followed by multiple appendixes. Appendix F is a security control catalog that provides details on hundreds of individual security controls, divided into 21 different families.

Additionally, each of these 21 families includes multiple groups. As an example, the Access Control family (AC) includes 25 different groups (AC-1 through AC-25). The Account Management group (AC-2) describes 110 individual security controls related to account management.

It's worth noting that SP 800-53 Revision 3 attempted to identify every control as technical, management, or operational. However, many controls included characteristics from more than just one of these classifications. NIST removed these references in Revision 4.

If you're interested in pursuing other security-related certifications or making IT security a career, the SP 800 documents are well worth your time. You can download SP 800-53 Revision 4 and other SP 800 documents at *http://csrc.nist.gov/publications/PubsSPs.html.*

Preventive Controls

Ideally, an organization won't have any security incidents and that is the primary goal of

preventive controls—to prevent security incidents. Some examples include:

- **Hardening.** Hardening is the practice of making a system or

application more secure than its default configuration. This uses a defense-in-depth strategy with layered security. This includes disabling unnecessary ports and services, implementing secure protocols, using strong passwords along with a robust password policy, and disabling default and unnecessary accounts. These topics are covered in more depth in Chapter 5.

- **Security awareness and training.** Ensuring that users are aware of security vulnerabilities and threats helps prevent incidents. When users understand how social engineers operate, they are less likely to be tricked. For example, uneducated users might be tricked into giving a social engineer their passwords, but educated users will see through the tactics and keep their passwords secure.
- **Security guards.** Guards prevent and deter many attacks. For example, guards can prevent unauthorized access into secure areas of a building by first verifying user identities. Although a social engineer might attempt to fool a receptionist into letting him into a secure area, the presence of a guard will deter many social engineers from even trying these tactics.
- **Change management.** Change management ensures that changes don't result in unintended outages. In other words, instead of administrators making changes on the fly, they submit the change to a change management process. Notice that change management is an operational control, which attempts to prevent incidents. In other words, it's both an operational control and a preventive control.
- **Account disablement policy.** An account disablement policy ensures that user accounts are disabled when an employee leaves. This prevents anyone, including ex-employees, from continuing to use these accounts. Chapter 2 covers account disablement policies in more depth.

Remember this

Preventive controls attempt to prevent security incidents. Hardening systems increases their basic configuration to prevent incidents. Security guards can prevent unauthorized personnel from entering a secure area. Change management processes help prevent outages from configuration changes. An account disablement policy

ensures that accounts are disabled when a user leaves the organization.

Detective Controls

Although preventive controls attempt to prevent security incidents, some will still occur. ***Detective controls*** attempt to detect when vulnerabilities have been exploited, resulting in a security incident. An important point is that detective controls discover the event after it's occurred. Some examples of detective controls are:

- **Log monitoring.** Several different logs record details of activity on systems and networks. For example, firewall logs record details of all traffic that the firewall blocked. By monitoring these logs, it's possible to detect incidents. Some automated methods of log monitoring automatically detect potential incidents and report them right after they've occurred.
- **Trend analysis.** In addition to monitoring logs to detect any single incident, you can also monitor logs to detect trends. For example, an intrusion detection system (IDS) attempts to detect attacks and raise alerts or alarms. By analyzing past alerts, you can identify trends, such as an increase of attacks on a specific system.
- **Security audit.** Security audits can examine the security posture of an organization. For example, a password audit can determine if the password policy is ensuring the use of strong passwords. Similarly, a periodic review of user rights can detect if users have more permissions than they should.
- **Video surveillance.** A closed-circuit television (CCTV) system can record activity and detect what occurred. It's worth noting that video surveillance can also be used as a deterrent control.
- **Motion detection.** Many alarm systems can detect motion from potential intruders and raise alarms.

Remember this

Detective controls attempt to detect when vulnerabilities have been exploited. Some examples include log monitoring, trend analysis, security audits, and CCTV systems.

Comparing Detection and Prevention Controls

It's worth stressing the differences between detection and prevention controls. A detective control can't predict when an incident will occur and it can't prevent it. In contrast, prevention controls stop the incident from occurring at all. Consider cameras and guards:

- **Video surveillance.** A simple camera without recording capabilities can prevent incidents because it acts as a deterrent. Compare this with a CCTV system with recording abilities. It includes cameras, which can prevent incidents. The full system is also a detection control because of the recording capabilities. Security professionals can review the recordings to detect incidents after they've occurred.
- **Guards.** Guards are primarily prevention controls. They will prevent many incidents just by their presence.

Corrective Controls

Corrective controls attempt to reverse the impact of an incident or problem after it has occurred. Some examples of corrective controls are:

- **IPS.** An intrusion prevention system (IPS) attempts to detect attacks and then modify the environment to block the attack from continuing. Chapter 4 covers IPSs in more depth.
- **Backups and system recovery.** Backups ensure that personnel can recover data if it is lost or corrupted. Similarly, system recovery procedures ensure administrators can recover a system after a failure. Chapter 9 covers backups and disaster recovery plans in more depth.

Deterrent Controls

Deterrent controls attempt to discourage a threat. Some deterrent controls attempt to discourage potential attackers from attacking, and others attempt to discourage employees from violating a security policy.

You can often describe many deterrent controls as preventive controls. For example, imagine an organization hires a security guard to control access to a restricted area of a building. This guard will deter most people from trying to sneak in simply by discouraging them from even trying. This deterrence prevents security incidents related to unauthorized access.

The following list identifies some physical security controls used to deter threats:

- **Cable locks.** Securing laptops to furniture with a cable lock deters thieves from stealing the laptops. Thieves can't easily steal a laptop secured this way. If they try to remove the lock, they will destroy it. Admittedly, a thief could cut the cable with a large cable cutter. However, someone walking around with a four-foot cable cutter looks suspicious. If you're not familiar with a cable lock, refer to the "Using Hardware Locks" section in Chapter 9.
- **Hardware locks.** Other locks such as locked doors securing a wiring closet or a server room also deter attacks. Many server bay cabinets also include locking cabinet doors.

Compensating Controls

Compensating controls are alternative controls used instead of a primary control. As an example, an organization might require employees to use smart cards when authenticating on a system. However, it might take time for new employees to receive their smart card. To allow new employees to access the network and still maintain a high level of security, the organization might choose to implement a Time-based One-Time Password (TOTP) as a compensating control. The compensating control still provides a strong authentication solution.

Combining Control Types and Goals

It's important to realize that the control types (technical, administrative, and physical) and control goals (preventive, detective, corrective, deterrent, and compensating) are not mutually exclusive. In other words, you can describe most controls using more than one category.

As an example, encryption is a preventive technical control. It helps prevent the loss of data confidentiality, so it is a preventive control. You implement it with technology, so it is a technical control. If you understand control categories, you shouldn't have any problems picking out the correct answers on the exam even if CompTIA combines them in a question, such as a preventive technical control.

Implementing Virtualization

Virtualization is a popular technology used within large data centers and can also be used on a regular personal computer (PC). It allows you to host one or more virtual systems, or virtual machines (VMs), on a single physical system. With today's technologies, you can host an entire virtual network within a single physical system and organizations are increasingly using virtualization to reduce costs.

When discussing VMs, you should understand the following terms:

- **Hypervisor.** The software that creates, runs, and manages the VMs is the hypervisor. Several virtualization technologies currently exist, including VMware products, Microsoft Hyper-V products, and Oracle VM VirtualBox. These applications have their own hypervisor software.
- **Host.** The physical system hosting the VMs is the host. It requires more resources than a typical system, such as multiple processors, massive amounts of RAM, fast and abundant hard drive space, and one or more fast network cards. Although these additional resources increase the cost of the host, it is still less expensive than paying for multiple physical systems. It also requires less electricity, less cooling, and less physical space.
- **Guest.** Operating systems running on the host system are guests or guest machines. Most hypervisors support several different operating systems, including various Microsoft operating systems and various Linux distributions. Additionally, most hypervisors support both 32-bit and 64-bit operating systems.
- **Host elasticity and scalability.** Elasticity and scalability refer to the ability to resize computing capacity based on the load. For example, imagine one VM has increased traffic. You can increase the amount of processing power and memory used by this server relatively easily. Similarly, it's relatively easy to decrease the resources when the load decreases.

Virtualization typically provides the best return on investment (ROI) when an organization has many underutilized servers. For example, imagine an organization has nine servers with each using only about 20 percent processing power, memory, and disk space. You could convert three physical

servers to virtual hosts and run three guest servers on each physical server. Assuming all the servers are similar, this wouldn't cost any more money for the physical servers. Additionally, three physical servers consume less electricity and require less heating and ventilation to maintain.

In contrast, imagine the organization has nine servers with each using about 80 percent of their processing power, memory, and disk space. Although it is possible to convert them all to virtual servers, it requires the purchase of additional hardware. The savings from less electricity and less heating and ventilation is offset by the cost of the new servers.

Comparing Hypervisors

Hypervisor virtualization is divided into primarily two different types:

- **Type I.** ***Type I hypervisors*** run directly on the system hardware. They are often called bare-metal hypervisors because they don't need to run within an operating system. For example, VMware has a family of ESX/ESXi products that are Type I hypervisors.
- **Type II.** ***Type II hypervisors*** run as software within a host operating system. For example, the Microsoft Hyper-V hypervisor runs within a Microsoft operating system.

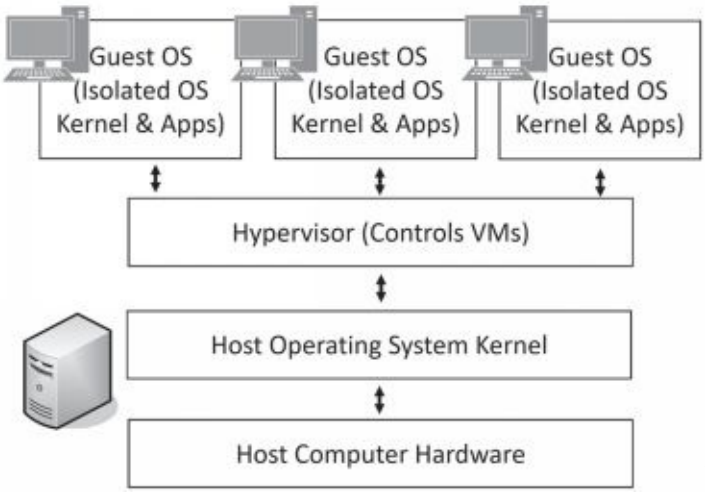

Figure 1.1 shows a single computer hosting three guest operating systems using Type II hypervisor-based virtualization. Notice that each guest has a full operating system, including its own kernel. Don't let the term kernel throw you. Generically, a kernel is just the central part or most important part of something. When referring to a computer, the kernel is the central part of the operating system.

Figure 1.1: Type II hypervisor-based virtualization

When implementing virtualization on a PC, you will use Type II hypervisor-based virtualization. However, virtualization in large-scale data centers typically uses Type I virtualization.

Application Cell or Container Virtualization

Application cell virtualization or container virtualization runs services or applications within isolated application cells (or containers). For comparison, Figure 1.2 shows an example of container virtualization. Notice that the containers don't host an entire operating system. Instead, the host's operating system and kernel run the service or app within each of the containers. However, because they are running in separate containers, none of the services or apps can interfere with services and apps in other containers.

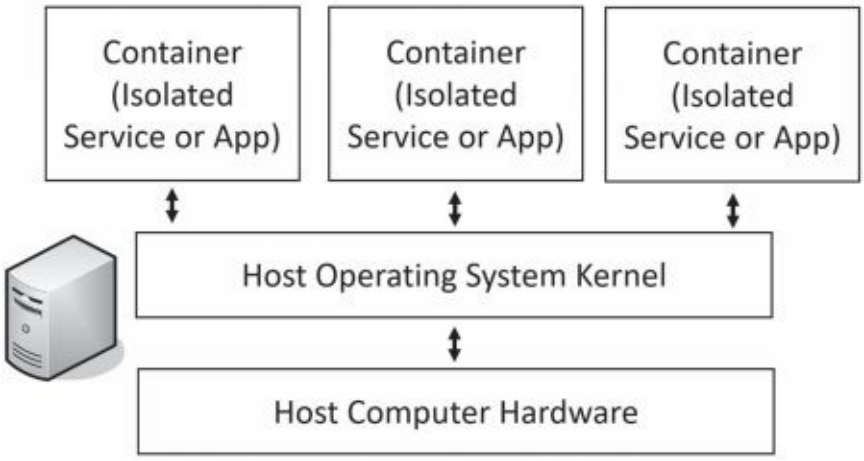

Figure 1.2: Container-based virtualization

A benefit of container virtualization is that it uses fewer resources and can be more efficient than a system using a traditional Type II hypervisor virtualization. Internet Service Providers (ISPs) often use it for customers who need specific applications. One drawback is that containers must use the operating system of the host. As an example, if the host is running Linux, all the containers must run Linux.

Remember this

Type I hypervisors run directly on bare-metal systems without an operating system. Type II hypervisors are software that run within an operating system. Container virtualization runs within isolated cells or containers and does not have its own kernel.

Secure Network Architecture

It is possible to use virtualization as part of an overall secure network architecture. One of the primary benefits is that VMs can provide segregation, segmentation, and isolation of individual systems. One way of doing so is to disable the network interface card (NIC) in the VM. This prevents it from transmitting any data in or out of the VM.

Snapshots

A ***snapshot*** provides you with a copy of the VM at a moment in time, which you can use as a backup. You are still able to use the VM just as you normally would. However, after taking a snapshot, the hypervisor keeps a record of all changes to the VM. If the VM develops a problem, you can revert the VM to the state it was in when you took the snapshot.

Administrators commonly take snapshots of systems prior to performing any risky operation. Risky operations include applying patches or updates, testing security controls, and installing new applications. Ideally, these operations do not cause any problems, but occasionally they do. By creating snapshots before these operations, administrators can easily revert or roll back the system to a known good state with a known good configuration.

Remember this

Virtualization allows multiple virtual servers to operate on a single physical server. It provides increased availability with lower operating costs. Additionally, virtualization provides a high level of flexibility when testing security controls, updates, and patches because they can easily be reverted using snapshots.

VDI/VDE and Non-Persistence

In addition to virtualization servers, it’s also possible to virtualize

desktops. In a virtual desktop infrastructure (VDI) or virtual desktop environment (VDE), a user's desktop operating system runs as a VM on a server.

One benefit of using a ***VDI/VDE*** is that user PCs can have limited hardware resources. If the PC can connect to a server over a network, it can run a full-featured desktop operating system from the server.

A primary consideration when running virtual desktops is whether they will support persistence or ***non-persistence***. In a persistent virtual desktop, each user has a custom desktop image. Users can customize them and save their data within the desktop. A drawback is the amount of disk space required on the server to support unique desktop images for all users.

Virtual desktops that support non-persistence serve the same desktop for all users. When a user accesses the remote server, it provides a desktop operating system from a preconfigured snapshot. Although users can make changes to the desktop as they're using it, it reverts to a known state (the original snapshot) when they log off. Another way of viewing this is that it rolls back to a known configuration.

VMs as Files

It's worth pointing out that virtual machines are simply files. These files certainly have some complexity, but still, they are just files. If you do the labs to create VMs on your computer, you'll be asked where you want to store these files. If you have a Windows computer, you can use File Explorer to browse to that location and view the files.

Because the VM is just a group of files, it becomes relatively easy to move them from one physical server to another. For example, if one of your physical servers becomes overloaded, you can move virtual servers off the overloaded system to another physical server. Some virtual server management software makes this as simple as dragging and dropping the virtual servers from one host to another.

It's also easy to restore a failed virtual server. If you create a backup of the virtual server files and the original server fails, you simply restore the files. You can measure the amount of time it takes to restore a virtual server in minutes. In contrast, rebuilding a physical server can take hours. Many virtualization products allow you to manage multiple virtual systems on a single server, even when the virtual servers are running on separate physical hosts. For example, you might have five physical servers hosting three

virtual servers each, and you can manage all of them through a single management interface. This includes taking snapshots, reverting snapshots, and moving the virtual servers from one physical host to another.

Risks Associated with Virtualization

Despite the strengths of virtualization technologies, you should understand some weaknesses. Many people consider virtual machine escape (VM escape) to be the most serious threat to virtual system security. Loss of confidentiality and loss of availability can also be a concern.

VM Escape

VM escape is an attack that allows an attacker to access the host system from within the virtual system. As previously mentioned, the host system runs an application or process called a hypervisor to manage the virtual systems. In some situations, the attacker can run code on the virtual system and interact with the hypervisor.

Most virtual systems run on a physical server with elevated privileges, similar to administrator privileges. A successful VM escape attack often gives the attacker unlimited control over the host system and each virtual system within the host.

When vendors discover VM escape vulnerabilities, they write and release patches. Just as with any patches, it is important to test and install these patches as soon as possible. This includes keeping both the physical and the virtual servers patched.

VM Sprawl

VM sprawl occurs when an organization has many VMs that aren't managed properly. Most organizations have specific policies in place to ensure physical servers are kept up to date and personnel only make changes to these servers after going through a change management process. These same policies should also apply to virtual servers.

Consider this scenario. Bart creates a VM running a Microsoft Windows Server version to test a software application. After testing the application, he leaves the VM running. Later, Microsoft releases security patches for the server. The IT department tests these patches and applies them to all of the known servers that need them. However, because Bart didn't tell anyone he

was creating the VM, it remains unpatched and vulnerable to attack.

Another challenge with VM sprawl is that each VM adds additional load onto a server. If unauthorized VMs are added to physical servers, they can consume system resources. The servers might become slower and potentially crash.

Loss of Confidentiality

As a reminder, each virtual system or virtual machine is just one or more files. Although this makes it easy to manage and move virtual machines, it also makes them easy to steal.

It's worth pointing out that many VMs include the operating system and data, just as a physical system would have both the operating system and data on its physical drives. For example, a virtual machine can include a database with credit card data, company financial records, or any type of proprietary data.

Imagine if an administrator became disgruntled. He has access to the systems and as a malicious insider, he can copy the virtual machine, take it home, and launch it on another physical server. At this point, he has access to the system and the data.

Running Kali Linux in a VM

Kali Linux is a free Linux distribution used by many security professionals for penetration testing and security auditing. Additionally, CompTIA listed Kali as one of the recommended software tools. The good news is that you can download Kali Linux for free and install it as a VM using one of the available free virtualization tools.

You can follow the online labs (*http://gcgapremium.com/501labs/)* to install Kali Linux within a VM. Note that you have some choices:

- **Hyper-V.** If you're running a version of Windows that supports Hyper-V, you can enable it on Windows and run VMs within it. One benefit of this is that in addition to creating a VM, you can also create a virtual switch.
- **VMware Workstation Player.** If your system doesn't support Hyper-V, you can run the free version of VMware Workstation Player. A drawback is that the free version of VMware Workstation Player doesn't support running multiple VMs within a single virtual

environment. However, you can upgrade to the paid version of VMware Workstation Pro to gain this and multiple other features.

- **Oracle VMVirtualBox.** This has been my favorite in the past. However, when writing this chapter, Kali Linux wouldn't successfully install in Oracle VM VirtualBox on my Windows system. This might be due to technical issues that Microsoft or Oracle will resolve with updates. Give it a try if you like. It might work perfectly for you.

Once you have Kali Linux installed in a virtual environment, you'll also be able to run some of the command-line tools mentioned in the CompTIA objectives that only run within Linux.

Using Command-Line Tools

Command-line tools can be invaluable when troubleshooting or analyzing systems. If you know how to use them, they can make many tasks easier. Additionally, the CompTIA Security+ objectives list several command-line tools that you should know to help you assess the security posture of an organization. Some are specific to Windows systems and run through the Windows Command Prompt window. Others are specific to Linux systems and run through the Linux terminal (sometimes called the shell).

As you read through this section and learn about these tools, I strongly encourage you to run the commands. You will also find some basic commands that you can run through in the online labs at *http://gcgapremium.com/501labs/.*

A challenge many test takers have is that they don't have a Linux system to play around with these commands. If you can't enter them and see what they do, you might have trouble with even the easy questions. The online labs include labs you can use to create a virtual Linux environment on a Windows system.

Windows Command Line

Before you can use the command line in Windows, you first need to launch the Windows Command Prompt window. The simplest way to launch the Command Prompt window is to right- click the Start button and select Command Prompt, as shown in Figure 1.3. The Start button is at the lower left of your screen and clicking it displays the Start menu with links to many

commonly used applications.

Figure 1.3: Launching the Windows Command Prompt window

In some situations, you need to start the Command Prompt window with elevated permissions as an administrator. To do this, right-click the Start button and select Command Prompt (Admin).

If you're using a different version of Windows (or Microsoft decides to modify the function of the Start button again), a quick search with your favorite search engine should help you identify how to open the Command Prompt window.

Linux Terminal

The terminal is where you run commands in a Linux system. There are different ways to access the terminal depending on the distribution you're using. If you're running Kali Linux (recommended while using this book), you can start it by simply clicking the terminal icon on the Kali menu.

Figure 1.4 shows an instance of

Kali with the menu on the left. If you hover over the second icon, you'll see "Terminal" appear. Click it and it starts the terminal.

Figure 1.4: Launching the terminal in Kali Linux

For simplicity, instead of stating “Linux or Unix” throughout this book, I’m just stating it as Linux. Note that Linux is a version of Unix and commands that can be run in a Unix terminal can also be run in a Linux terminal.

Understanding Switches and Getting Help

Almost every command you’ll run across has options available that you can invoke with a switch. A Windows command-line switch uses a forward slash (/) or a dash (-) after the command and includes an option that modifies the command to perform a different function.

Linux commands use switches too, but they typically only use a dash. If you use a forward slash, you will typically get unexpected results or an error.

The most used switch in Windows systems is the help switch identified with a question mark. For example, you can use the help switch to get help using these commands:

- ping /? or ping -?
- ipconfig /? or ipconfig -?
- netstat /? or netstat -?

Although Linux terminal commands use switches too, they don’t use the question mark for help. Instead, if you want basic help on a command, you can often just type the command without any switch, or use the pipe symbol (|) and the word help:

- ping
- ping | help

Most Linux distributions include a built-in user manual that you can query. The manual is organized by man pages and you query them with the man command. For example, you can use the following command to get help on ping:

- man ping

Unfortunately, there isn’t a consistent standard to get help for all commands in Linux. Sometimes, one method works, but another one doesn’t. The key is to know the different methods so that you can use an alternate

method if necessary.

Understanding Case

Most Windows commands are not case sensitive. In other words, you can type in a command using uppercase characters, lowercase characters, or any combination. For example, each of the following commands will ping the localhost IPv6 address (::1) and provide the same output:

- ping -6 localhost
- PiNg -6 localHOST
- PING -6 LocalHost

However, this is not true in the Linux terminal. Instead, commands are typically lowercase and if you use uppercase letters, you'll find that the command is not recognized. Of the three commands shown for the Windows command line, only ping -6 localhost will work within the Linux terminal.

As you go through these commands, note that many of them support both IPv4 addresses and IPv6 addresses. By querying help, you can see how to do each.

Ping

Ping is a basic command used to test connectivity for remote systems. You can also use it to verify a system can resolve valid host names to IP addresses, test the NIC, and check the security posture of a network.

The ping command checks connectivity by sending Internet Control Message Protocol (ICMP) echo request packets. Remote systems answer with ICMP echo reply packets and if you receive echo replies, you know that the remote system is operational. As a simple example, the following command verifies that your computer can connect with another computer on your network:

ping 192.168.1.1

On Windows systems, ping sends out four ICMP echo requests. Systems that receive the ICMP echo requests respond with ICMP echo replies. On Linux-based systems, ping continues until you press the Ctrl + C keys to stop it. You can mimic this behavior on Windows systems by using the -t switch like this:

ping -t 192.168.1.1

Similarly, you can mimic the behavior of a Windows ping on a Linux system using the -c switch (for count) like this:

ping -c 4 192.168.1.1

This example tested connectivity with an IP address in a local network, but you can just as easily test connectivity with any system. For example, if you knew the IP address of a system hosting a web site on the Internet, you could ping its IP address.

Using Ping to Check Name Resolution

The name resolution process resolves a host name (such as getcertifiedgetahead.com) to an IP address. There are several elements of name resolution. Typically, a computer will query a Domain Name System (DNS) with the host name and DNS will respond with an IP address.

Some malware attempts to break the name resolution process for specific hosts. For example, Windows systems get updates from a Windows Update server. In some cases, malware changes the name resolution process to prevent systems from reaching the Windows Update server and getting updates.

You can ping the host name of a remote system and verify that name resolution is working. As an example, the following command will resolve the host name (getcertifiedgetahead.com) to an IP address:

ping getcertifiedgetahead.com

Here's the result when the command is executed at the command prompt on a Windows 10 system. Notice that the first line shows that ping resolved the host name (getcertifiedgetahead.com) to its IP address (72.52.206.134):

```
Pinging getcertifiedgetahead.com [72.52.206.134] with 32 bytes of data:
Reply from 72.52.206.134: bytes=32 time=45ms TTL=116
Reply from 72.52.206.134: bytes=32 time=45ms TTL=116
Reply from 72.52.206.134: bytes=32 time=48ms TTL=116
Reply from 72.52.206.134: bytes=32 time=45ms TTL=116
Ping statistics for 72.52.206.134:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds: Minimum =
45ms, Maximum = 48ms, Average = 45ms
```

Beware of Firewalls

If you receive replies from a system, it verifies the other system is operational and reachable. However, if the ping command fails, it doesn't necessarily mean that the remote system is operational or not reachable. Ping might show a "Reply Timed Out" error even if the remote system is functioning properly.

Many denial-of-service (DoS) attacks use ICMP to disrupt services on Internet-based systems. To protect systems, firewalls commonly block ICMP traffic to prevent these attacks from succeeding. In other words, a remote system might be operational, but a ping will fail because the firewall is blocking ICMP traffic.

As an example, you might be able to connect to the *http://blogs.getcertifiedgetahead.com* web site using a web browser, but ping might fail. This indicates that the web site is operational with Hypertext Transfer Protocol (HTTP), but a firewall is blocking ICMP traffic.

Using Ping to Check Security Posture

You can also use ping to check the security posture of a network. For example, if you've configured firewalls and routers to block ping traffic, you can verify the firewalls and routers are blocking the traffic by using ping to check it.

Chapter 4 covers intrusion prevention systems (IPSs) in more depth, but as an introduction, they can often detect attacks and block them automatically. For example, a simple distributed denial-of-service (DDoS) attack can send thousands of pings to a server and overload it. An IPS can detect the attack and automatically block ICMP traffic, effectively preventing any impact from the attack.

You can use ping to simulate an attack from a couple of computers to repeatedly send ping requests. If the IPS is working, it will block these attacks and the pings will stop receiving replies.

Remember this

Administrators use ping to check connectivity of remote systems and verify name resolution is working. They also use ping to check the security posture of systems and networks by verifying that routers, firewalls, and IPSs block ICMP traffic when configured to do

so.

Ipconfig, ifconfig, and ip

The ***ipconfig*** command (short for Internet Protocol configuration) shows the Transmission Control Protocol/Internet Protocol (TCP/IP) configuration information for a system. This includes items such as the computer's IP address, subnet mask, default gateway, MAC address, and the address of a Domain Name System (DNS) server. The command shows the configuration information for all network interface cards (NICs) on a system, including both wired and wireless NICs. Technicians often use ipconfig as a first step when troubleshooting network problems.

Linux-based systems use ***ifconfig*** (short for interface configuration) instead of ipconfig. A benefit is that ifconfig has more capabilities than ipconfig, allowing you to use it to configure the NIC in addition to listing the properties of the NIC.

The following list shows some common commands:

- **ipconfig.** Entered by itself, the command provides basic information about the NIC, such as the IP address, subnet mask, and default gateway.
- **ipconfig /all.** This command shows a comprehensive listing of TCP/IP configuration information for each NIC. It includes the media access control (MAC) address, the address of assigned DNS servers, and the address of a Dynamic Host Configuration Protocol (DHCP) server if the system is a DHCP client. You can use ifconfig -a on Linux systems.
- **ipconfig /displaydns.** Each time a system queries DNS to resolve a host name to an IP address, it stores the result in the DNS cache and this command shows the contents of the DNS cache. It also shows any host name to IP address mappings included in the hosts file.
- **ipconfig /flushdns.** You can erase the contents of the DNS cache with this command. Use this when the cache has incorrect information and you want to ensure that DNS is queried for up-to-date information.

The following commands are unique to Linux systems:

- **ifconfig eth0.** This command shows the configuration of the first Ethernet interface (NIC) on a Linux system. If the system has multiple

NICs, you can use eth1, eth2, and so on. You can also use wlan0 to view information on the first wireless interface.

- **ifconfig eth0 promisc.** This command enables promiscuous mode on the first Ethernet interface. Promiscuous mode allows a NIC to process all traffic it receives. Normally, a NIC is in non-promiscuous mode and it ignores all packets not addressed to it. You can disable promiscuous mode with this command: ifconfig eth0 -promisc.
- **ifconfig eth0 allmulti.** This command enables multicast mode on the NIC. This allows the NIC to process all multicast traffic received by the NIC. Normally, a NIC will only process multicast traffic for multicast groups that it has joined. You can disable multicast mode with this command: ifconfig eth0 -allmulti.

Normally, a NIC uses non-promiscuous mode and only processes packets addressed directly to its IP address. However, when you put it in promiscuous mode, it processes all packets regardless of the IP address. This allows the protocol analyzer to capture all packets that reach the NIC.

The ifconfig command was deprecated in 2009 in Debian distributions of Linux. Deprecated means that its use is discouraged but tolerated. The ifconfig command is part of the net-tools package and Linux Debian developers are no longer maintaining that package. However, you'll still see ifconfig and other tools in the net-tools package on most Linux systems, including Kali Linux.

Instead of using ifconfig, Linux developers recommend you use ***ip*** instead. Although the ip command can display information and configure network interfaces, it doesn't use the same commands or have the same abilities. For example, it doesn't have a command you can use to enable promiscuous mode on a NIC. Here are a few commands that you can use with ip:

- **ip link show.** Shows the interfaces along with some details on them
- **ip link set eth0 up.** Enables a network interface
- **ip -s link.** Shows statistics on the network interfaces

Remember this

Windows systems use ipconfig to view network interfaces. Linux systems use ifconfig, and ifconfig can also manipulate the settings on the network interfaces. You can enable promiscuous mode on a NIC with ifconfig. The ip command is similar to ifconfig

and can be used to view and manipulate NIC settings.

Netstat

The ***netstat*** command (short for network statistics) allows you to view statistics for TCP/IP protocols on a system. It also gives you the ability to view active TCP/IP network connections. Many attacks establish connections from an infected computer to a remote computer. If you suspect this, you can often identify these connections with netstat.

Some of the common commands you can use with netstat are:

- **Netstat.** Displays a listing of all open TCP connections.
- **Netstat -a.** Displays a listing of all TCP and User Datagram Protocol (UDP) ports that a system is listening on, in addition to all open connections. This listing includes the IP address followed by a colon and the port number, and you can use the port number to identify protocols. As an example, if you see an IP address followed by :80, it indicates the system is listening on the default port of 80 for HTTP. This indicates this system is likely a web server.
- **Netstat –r.** Displays the routing table.
- **Netstat -e.** Displays details on network statistics, including how many bytes the system sent and received.
- **Netstat -s.** Displays statistics of packets sent or received for specific protocols, such as IP, ICMP, TCP, and UDP.
- **Netstat -n.** Displays addresses and port numbers in numerical order. This can be useful if you're looking for information related to a specific IP address or a specific port.
- **Netstat -p *protocol.*** Shows statistics on a specific protocol, such as TCP or UDP. For example, you could use netstat -p tcp to show only TCP statistics.

You can combine many of the netstat switches to show different types of information. For example, if you want to show a listing of ports that the system is listening on (-a), listed in numerical order (-n), for only the TCP protocol (-p tcp), you could use this command:

netstat -anp tcp

Netstat displays the state of a connection, such as ESTABLISHED to indicate an active connection. RFC 793 (*https://tools.ietf.org/rfc/rfc793.txt*)

formally defines these states. Some of the common states are:

- **ESTABLISHED.** This is the normal state for the data transfer phase of a connection. It indicates an active open connection.
- **LISTEN.** This indicates the system is waiting for a connection request. The well-known port a system is listening on indicates the protocol.
- **CLOSE_WAIT.** This indicates the system is waiting for a connection termination request.
- **TIME_WAIT.** This indicates the system is waiting for enough time to pass to be sure the remote system received a TCP-based acknowledgment of the connection.
- **SYN_SENT.** This indicates the system sent a TCP SYN (synchronize) packet as the first part of the SYN, SYN-ACK (synchronize-acknowledge), ACK (acknowledge) handshake process and it is waiting for the SYN-ACK response.
- **SYN_RECEIVED.** This indicates the system sent a TCP SYN-ACK packet after receiving a SYN packet as the first part of the SYN, SYN-ACK, ACK handshake process. It is waiting for the ACK response to establish the connection. An excessive number of SYN_RECEIVED states indicate a SYN attack where an attacker is flooding a system with SYN packets but never finalizes the connection with ACK packets.

Tracert

The ***tracert*** command lists the routers between two systems. In this context, each router is referred to as a hop. Tracert identifies the IP address and sometimes the host name of each hop in addition to the round-trip times (RTTs) for each hop. Windows-based systems use tracert and Linux-based systems use traceroute, but they both function similarly. For simplicity, I'm using the command name tracert in this section, but this section applies to both equally.

Network administrators typically use tracert to identify faulty routers on the network. Ping tells them if they can reach a distant server. If the ping fails, they can use tracert to identify where the traffic stops. Some of the hops will succeed, but at some point, tracert will identify where packets are lost, giving them insight into where the problem has occurred. Other times, they will see where the RTTs increase as traffic is routed around a faulty router. Tracing a path is especially valuable when troubleshooting issues through a wide area network (WAN).

From a security perspective, you can use tracert to identify modified paths. As an example, consider Figure 1.5. Users within the internal network normally access the Internet directly through Router 1. However, what if an attacker installed an unauthorized router between Router 1 and the Internet?

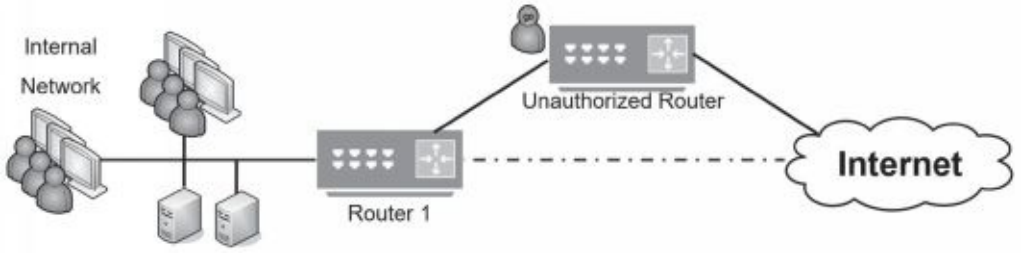

Figure 1.5: Tracing a path with tracert

Traffic will still go back and forth to the users. However, the attacker could capture the traffic with a protocol analyzer and view any data sent in cleartext. The attacker may also launch other attacks, such as some of the attacks discussed in Chapter 7.

From another perspective, you can identify if Internet paths have been modified. Imagine that you often connect to a server in New York from a

New York location. Today, the connection seems abnormally slow. You could use tracert to verify the path. If you notice that traffic is now going through IP addresses in foreign countries, it indicates a problem.

Give it a try. Launch a command prompt and use the following commands to see some common uses and outputs from the tracert command:

- Type in **tracert blogs.getcertifiedgetahead.com** and press Enter. Identify how many hops are between your system and this web server. Identify if any RTTs are significantly longer than others.
- Type in **tracert –d blogs.getcertifiedgetahead.com** and press Enter. Notice that the -d switch forces tracert to not resolve IP addresses to host names, allowing the command to finish more quickly.

Arp

Arp is a command-line tool that is related to the Address Resolution Protocol (ARP); however, arp (the command) and ARP (the protocol) are not the same thing. Chapter 3 discusses ARP (the protocol), but as a short introduction, ARP resolves IP addresses to MAC addresses and stores the result in the ARP cache.

You can use the arp command to view and manipulate the ARP cache. Here are some sample commands:

- **arp.** Without a switch, shows help on Windows
- **arp.** Without a switch, shows the ARP cache on Linux
- **arp -a.** Shows the ARP cache on Windows
- **arp -a 192.168.1.1.** Displays the ARP cache entry for the specified IP address

You can also use arp to identify the MAC address of other systems on your local network. As an example, imagine you want to identify the MAC address of server1. You can ping server1 and ARP will identify server1's IP address. You can then use arp -a to show the ARP cache, which includes the MAC address for server1.

Chapter 7 covers several attacks, including ARP cache poisoning where attackers manipulate the ARP cache. If you suspect an ARP cache poisoning attack, you can use arp to check the cache.

Chapter 1 Exam Topic Review

When preparing for the exam, make sure you understand these key concepts covered in this chapter.

Understanding Core Security Goals

- A use case helps professionals identify and clarify requirements to achieve a goal.
- Confidentiality ensures that data is only viewable by authorized users. Encryption is the best choice to provide confidentiality. Access controls also protect the confidentiality of data.
- Steganography (hiding data inside other data) is one method of supporting obfuscation by making the hidden data harder to see.
- Integrity provides assurances that data has not been modified, tampered with, or corrupted through unauthorized or unintended changes. Data can be a message, a file, or data within a database. Hashing is a common method of ensuring integrity.
- Non-repudiation prevents entities from denying they took an action. Digital signatures and audit logs provide non-repudiation. Digital signatures also provide integrity for files and email.
- Availability ensures that data and services are available when needed. A common goal is to remove single points of failure. Methods used to increase or maintain availability include fault tolerance, failover clusters, load balancing, backups, virtualization, HVAC systems, and generators.

Introducing Basic Risk Concepts

- Risk is the possibility of a threat exploiting a vulnerability and resulting in a loss.
- A threat is any circumstance or event that has the potential to compromise confidentiality, integrity, or availability.
- A vulnerability is a weakness. It can be a weakness in the hardware, software, configuration, or users operating the system.
- Risk mitigation reduces risk by reducing the chances that a threat will exploit a vulnerability or by reducing the impact of the risk.

- Security controls reduce risks. For example, antivirus software is a security control that reduces the risk of virus infection.

Understanding Control Types

- The three primary security control types are technical (implemented with technology), administrative (using administrative or management methods), and physical (using controls that you can physically touch).
- A technical control is one that uses technology to reduce vulnerabilities. Encryption, antivirus software, IDSs, firewalls, and the principle of least privilege are technical controls.
- Administrative controls are primarily administrative and include items such as risk and vulnerability assessments. Some administrative controls help ensure that day-to-day operations of an organization comply with their overall security plan. Some examples include security awareness and training, configuration management, and change management.
- Preventive controls attempt to prevent security incidents. Examples include system hardening, user training, guards, change management, and account disablement policies.
- Detective controls attempt to detect when a vulnerability has been exploited. Examples include log monitoring, trend analysis, security audits (such as a periodic review of user rights), video surveillance systems, and motion detection systems.
- Corrective controls attempt to reverse the impact of an incident or problem after it has occurred. Examples include intrusion prevention systems (IPSs), backups, and system recovery plans.
- Deterrent controls attempt to prevent incidents by discouraging threats.
- Compensating controls are alternative controls used when it isn't feasible or possible to use the primary control.

Implementing Virtualization

- Virtualization allows multiple servers to operate on a single physical host. They provide increased availability with various tools such as snapshots and easy restoration.
- Type I hypervisors run directly on the system hardware. They are

often called bare-metal hypervisors because they don’t need to run within an operating system.

- Type II hypervisors run as software within a host operating system.
- Container virtualization is a specialized version of a Type II hypervisor. It allows services or applications to run within their own isolated cells or containers. Containers don’t have a full operating system but instead use the kernel of the host.
- Snapshots capture the state of a VM at a moment in time. Administrators often take a snapshot before performing a risky operation. If necessary, they can revert the VM to the snapshot state.
- VM sprawl can occur if personnel within the organization don’t manage the VMs.
- VM escape attacks allow an attacker to access the host system from the VM. The primary protection is to keep the host and guests up to date with current patches.

Using Command-Line Tools

- You run command-line tools in the Command Prompt window (in Windows) and the terminal (in Linux).
- The ping command can be used to check connectivity; check name resolution; and verify that routers, firewalls, and intrusion prevention systems block ICMP.
- The ipconfig command on Windows allows you to view the configuration of network interfaces.
- Linux uses ifconfig and/or ip to view and manipulate the configuration of network interfaces. You can enable promiscuous mode on a NIC with ifconfig.
- Netstat allows you to view statistics for TCP/IP protocols and view all active network connections. This can be useful if you suspect malware is causing a computer to connect with a remote computer.
- Tracert lists the routers (also called hops) between two systems. It can be used to verify a path has not changed.
- The arp command allows you to view and manipulate the ARP cache. This can be useful if you suspect a system’s ARP cache has been modified during an attack.

Online References

- Remember, you can access online references such as labs and sample performance- based questions at *http://gcgapremium.com/501-extras*.