Foundations of Information Security

A Straightforward Introduction

Jason Andress

no starch press

BRIEF CONTENTS

CONTENTS IN DETAIL

ACKNOWLEDGMENTS

I want to thank my wife for bearing with me through another writing project, especially during my excessive complaining and foot dragging over (*ahem*) certain chapters <3.

I also want to thank the whole crew at No Starch Press for all their time and hard work in making this a better book. Without all the many rounds of editing, reviewing, and feedback, this book would have been a considerably less polished version of itself.

INTRODUCTION

When I was in school, I was faced with a choice between pursuing a concentration in either information security or software engineering. The software engineering courses had terribly boring-sounding titles, so information security it was. Little did I know what a twisted and winding path I'd embarked on.

Information security as a career can take you many different places. Over the years, I've dealt with large-scale malware outbreaks, collected forensic information for court cases, hunted for foreign hackers in computer systems, hacked into systems and applications (with permission!),

pored over an astonishing amount of log data, implemented and maintained all manner of security tooling, authored many thousands of lines of code to fit square pegs into round holes, worked on open source projects, spoken at security conferences, taught classes, and written somewhere into the upper regions of hundreds of thousands of words on the topic of security.

This book surveys the information security field as a whole. It's well-suited to anyone wondering what people mean when they use the term *information security*—or anyone interested in the field and wondering where to start. The chapters offer clear, nontechnical explanations of how information security works and how to apply these principles to your own career. It should help you learn about information security without making you consult a massive textbook. I'll first cover the fundamental ideas, such as authentication and authorization, needed to understand the field's key concepts, such as the principle of least privilege and various security models. I'll then dive into a survey of real-world applications of these ideas in the areas of operations, human, physical, network, operating system, mobile, embedded, Internet of Things (IoT), and application security. I'll finish up by looking at how to assess security.

Who Should Read This Book?

This book will be a valuable resource to beginning security professionals, as well as to network and system administrators. You should use the information provided to develop a better understanding of how you protect your information assets and defend against attacks, as well as how to apply these concepts systematically to make your environment more secure.

Those in management positions will find this information useful as well, because it should help you develop better overall security practices for your organizations. The concepts discussed in this book can be used to drive security projects and policies and to mitigate some of the issues discussed.

About This Book

This book is designed to take you through a foundational understanding of information security from the ground up, so it's best read from start to finish. Throughout the book you will see numbered references to the Notes section at the end of the book, where you can find more information on some of these topics. Here's what you'll find in each chapter:

Chapter 1: What Is Information Security?
Introduces some of the most basic concepts of information security, such as the confidentiality,

integrity, and availability triad; basic concepts of risk; and controls to mitigate it.

Chapter 2: Identification and Authentication Covers the security principles of identification and authentication.

Chapter 3: Authorization and Access Controls Discusses the use of authorization and access controls, which are means of determining who or what can access your resources.

Chapter 4: Auditing and Accountability Explains the use of auditing and accountability for making sure you're aware of what people are doing in your environment.

Chapter 5: Cryptography Covers the use of cryptography for protecting the confidentiality of your data.

Chapter 6: Compliance, Laws, and Regulations Outlines the laws and regulations relevant to information security and what it means to comply with them.

Chapter 7: Operations Security Covers operations security, which is the process you use to protect your information.

Chapter 8: Human Element Security Explores issues pertaining to the human element of information security, such as the tools and techniques that attackers use to con us and how to defend against them.

Chapter 9: Physical Security Discusses the physical aspects of information security.

Chapter 10: Network Security Examines how you might protect your networks from a variety of different angles, such as network design, security devices, and security tooling.

Chapter 11: Operating System Security Explores the strategies you can use for securing the operating system, such as hardening and patching, and the steps that you can take to do so.

Chapter 12: Mobile, Embedded, and Internet of Things Security Covers how to ensure security for mobile devices, embedded devices, and IoT devices.

Chapter 13: Application Security Considers the various methods for securing applications.

Chapter 14: Assessing Security Discusses tools such as scanning and penetration testing that you can use to suss out security issues in your hosts and applications.

Writing this book was an adventure for me, as always. I

hope you enjoy the result and that your understanding of the world of information security expands. The security world can be an interesting and, at times, hair-raising field to work in. Welcome and good luck!

1

WHAT IS INFORMATION SECURITY?

Today, many of us work with computers, play on computers at home, go to school online, buy goods from merchants on the internet, take our laptops to the coffee shop to read emails, use our smartphones to check our bank balances, and track our exercise with sensors on our wrists. In other words, computers are ubiquitous.

Although technology allows us to access a host of

information with only a click of the mouse, it also poses major security risks. If the information on the systems used by our employers or our banks becomes exposed to an attacker, the consequences could be dire indeed. We could suddenly find the contents of our bank account transferred to a bank in another country in the middle of the night. Our employer could lose millions of dollars, face legal prosecution, and suffer damage to its reputation because of a system configuration issue that allowed an attacker to gain access to a database containing personally identifiable information (PII) or proprietary information. Such issues appear in the news media with disturbing regularity.

Thirty years ago, such breaches were nearly nonexistent, largely because the technology was at a relatively low level and few people were using it. Although technology changes at an increasingly rapid rate, much of the theory about keeping ourselves secure lags behind. If you can gain a good understanding of the basics of information security, you're on a strong footing to cope with changes as they come.

In this chapter, I'll cover some of the basic concepts of information security, including security models, attacks, threats, vulnerabilities, and risks. I'll also delve into some slightly more complex concepts when discussing risk management, incident response, and defense in depth.

Defining Information Security

Generally speaking, *security* means protecting your assets, whether from attackers invading your networks, natural disasters, vandalism, loss, or misuse. Ultimately, you'll attempt to secure yourself against the most likely forms of attack, to the best extent you reasonably can, given your environment.

You may have a broad range of potential assets you want to secure. These could include physical items with inherent value, such as gold, or those that have value to your business, such as computing hardware. You may also have valuables of a more ethereal nature, such as software, source code, or data.

In today's computing environment, you're likely to find that your logical assets (assets that exist as data or intellectual property) are at least as valuable as your physical assets (those that are tangible objects or materials), if not more valuable. That's where information security comes in.

Information security is defined as "protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction," according to US law.[1] In other words, you want to protect your data and systems from those who seek to misuse them, intentionally or unintentionally, or those who should not have access to them at all.

When Are You Secure?

Eugene Spafford once said, "The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards—and even then, I have my doubts."[2] A system in such a state might be secure, but it's not usable or productive. As you increase the level of security, you usually decrease the level of productivity.

Additionally, when securing an asset, system, or environment, you must consider how the level of security relates to the value of the item being secured. If you're willing to accommodate the decrease in performance, you can apply very high levels of security to every asset for which you're responsible. You could build a billion-dollar facility surrounded by razor-wire fences and patrolled by armed guards and vicious attack dogs, complete with a hermetically sealed vault, to safeguard your mom's chocolate chip cookie recipe, but that would be overkill. The cost of the security you put in place should never outstrip the value of what it's protecting.

In some environments, however, such security measures might not be enough. In any environment where you plan to put heightened levels of security in place, you also need to consider the cost of replacing your assets if you happen to lose them and make sure you establish reasonable levels of protection for their value.

Defining the exact point at which you can be considered secure presents a bit of a challenge. Are you secure if your systems are properly patched? Are you secure if you use strong passwords? Are you secure if you're disconnected from the internet entirely? From my point of view, the answer to all these questions is no. No single activity or action will make you secure in every situation.

That's because even if your systems are properly patched, there will always be new attacks to which you're vulnerable. When you're using strong passwords, an attacker will exploit a different avenue instead. When you're disconnected from the internet, an attacker could still physically access or steal your systems. In short, it's difficult to define when you're truly secure. On the other hand, defining when you're insecure is a much easier task. Here are several examples that would put you in this state:

- Not applying security patches or application updates to your systems
- Using weak passwords such as "password" or "1234"
- Downloading programs from the internet
- Opening email attachments from unknown senders
- Using wireless networks without encryption

I could go on for some time adding to this list. The good

thing is that once you can point out the areas in an environment that can make it insecure, you can take steps to mitigate these issues. This problem is similar to cutting something in half over and over. There will always be some small portion left to cut in half again. Although you may never get to a state that you can definitively call "secure," you can take steps in the right direction.

THIS LAW IS YOUR LAW ...

The bodies of law that define standards for security vary quite a bit from one industry to another and differ wildly from one country to another. An example of this is the difference in data privacy laws between the United States and the European Union. Organizations that operate globally need to take care that they're not violating any such laws while conducting business. When in doubt, consult legal counsel before acting.

Some bodies of law or regulations do try to define what secure means, or at least some of the steps you should take to be "secure enough." The Payment Card Industry Data Security Standard (PCI DSS) applies to companies that process credit card payments, the Health Insurance Portability and Accountability Act of 1996 (HIPAA) is for organizations that handle healthcare and patient records, the Federal Information

Security Management Act (FISMA) defines security standards for many federal agencies in the United States, and there are a host of others. Whether these standards are effective is debatable, but following the security standards defined for the industry in which you're operating is advisable, if not mandated.

Models for Discussing Security Issues

When discussing security issues, it's often helpful to have a model that you can use as a foundation or a baseline. This provides a consistent set of terminology and concepts that we, as security professionals, can refer to.

The Confidentiality, Integrity, and Availability Triad

Three of the primary concepts in information security are confidentiality, integrity, and availability, commonly known as the *confidentiality, integrity, and availability (CIA) triad*, as shown in Figure 1-1.

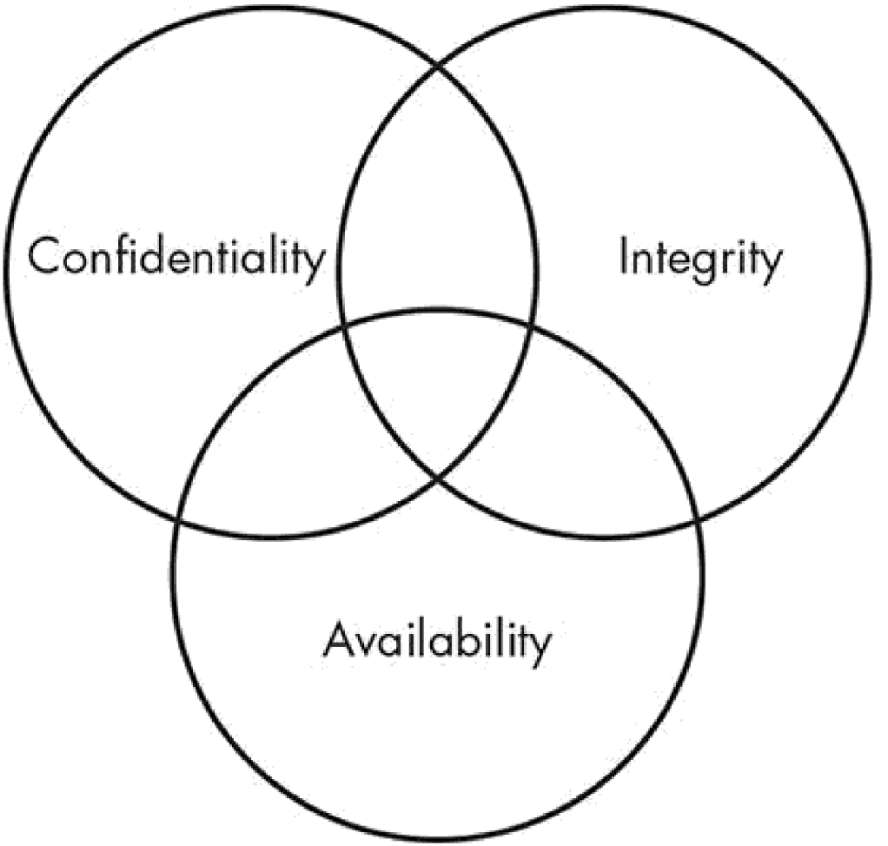

Figure 1-1: The CIA triad

The CIA triad is a model by which you can think about and discuss security concepts. It's also sometimes written as CAI or expressed in its negative form as disclosure, alteration, and denial (DAD).

Confidentiality

Confidentiality refers to our ability to protect our data from those who are not authorized to view it. You could implement confidentiality at many levels of a process.

As an example, imagine a person is withdrawing money from an ATM. The person in question will likely seek to maintain the confidentiality of the personal identification number (PIN) that allows him to draw funds from the ATM if he has his ATM card. Additionally, the owner of the ATM will maintain the confidentiality of the account number, balance, and any other information needed to communicate to the bank from which the funds are being drawn. The bank will also maintain the confidentiality of the transaction with the ATM and the balance change in the account after the funds have been withdrawn.

Confidentiality can be compromised in a number of ways. You could lose a laptop containing data. A person could look over your shoulder while you enter a password. You could send an email attachment to the wrong person, or an attacker could penetrate your systems, to name a few ways.

Integrity

Integrity is the ability to prevent people from changing your data in an unauthorized or undesirable manner. To maintain integrity, not only do you need to have the means to prevent unauthorized changes to your data, but you need the ability to reverse unwanted authorized changes.

A good example of mechanisms that allow you to control integrity are in the file systems of many modern operating systems, such as Windows and Linux. For the purposes of preventing unauthorized changes, such systems often implement permissions that restrict what actions an unauthorized user can perform on a given file. For example, the owner of a file might have permission to read it and write to it, while others might have permission only to read, or no permission to access it at all. Additionally, some such systems and many applications, such as databases, can allow you to undo or roll back changes that are undesirable.

Integrity is particularly important when it concerns data that provides the foundation for other decisions. If an attacker were to alter the data that contained the results of medical tests, a doctor might prescribe the wrong treatment, which could kill the patient.

Availability

The final leg of the CIA triad is availability. *Availability* refers to the ability to access our data when we need it. You could lose availability due to a power loss, operating system or application problems, network attacks, or the compromising of a system, for example. When an outside party, like an attacker, causes such issues, we typically call this a *denial-of-service* (DoS) attack.

How Does the CIA Triad Relate to Security?

Given the elements of the CIA triad, we can begin to discuss security issues with more detail than we otherwise could. For example, let's consider a shipment of backup tapes on which you've stored the only existing, unencrypted copies of some sensitive data.

If you were to lose the shipment in transit, you would have a security issue. This is likely to include a breach of confidentiality since your files were not encrypted. The lack of encryption could also cause integrity issues. If you recover the tapes in the future, it may not be immediately obvious to you if an attacker had altered the unencrypted files, as you would have no good way to discern altered from unaltered data. As for availability, you'll have an issue unless the tapes are recovered since you don't have backup copies of the files.

Although you can describe the situation in this example with relative accuracy using the CIA triad, you might find that the model is too restrictive to describe the entire situation. A more extensive model, the Parkerian hexad, exists for these cases.

The Parkerian Hexad

The Parkerian hexad, a less well-known model named after Donn Parker and introduced in his book *Fighting Computer Crime*, provides a somewhat more complex variation of the

classic CIA triad. Where the CIA triad consists only of confidentiality, integrity, and availability, the *Parkerian hexad* consists of these three principles as well as possession or control, authenticity, and utility,[3] for a total of six principles, as shown in Figure 1-2.

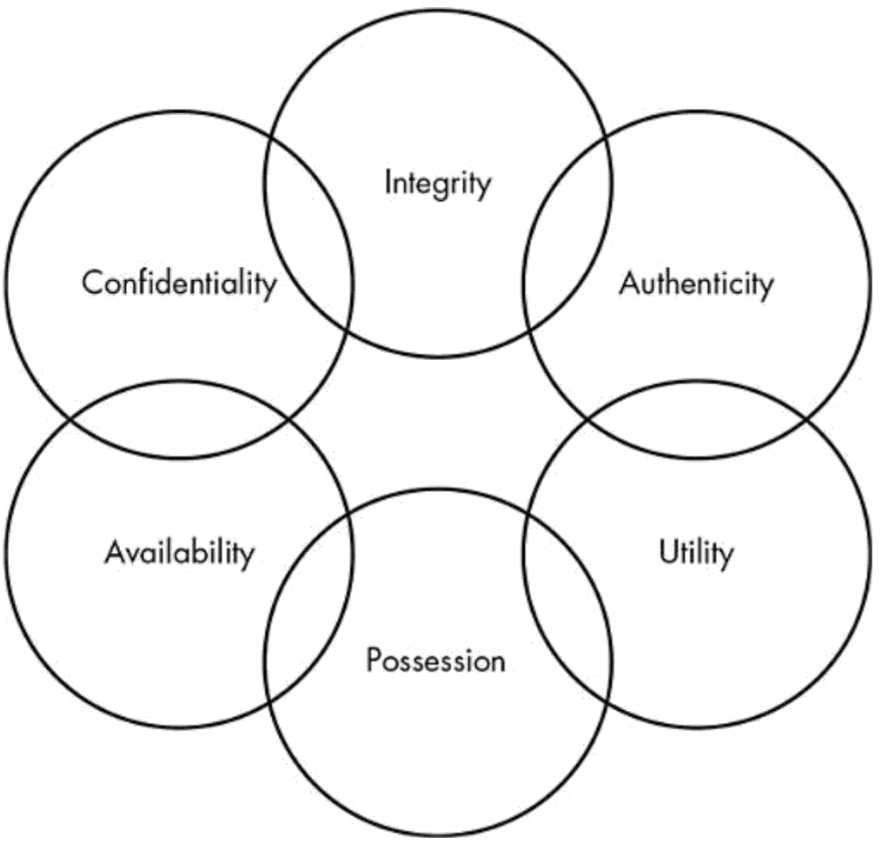

Figure 1-2: The Parkerian hexad

Confidentiality, Integrity, and Availability

As I mentioned, the Parkerian hexad includes the three principles of the CIA triad, with the same definitions just discussed. Parker describes integrity slightly differently; he doesn't account for authorized, but incorrect, modification of data. For him, the data must be whole and completely unchanged from its previous state.

Possession or Control

In the Parkerian hexad, *possession* or *control* refers to the physical disposition of the media on which the data is stored. This enables you to discuss your loss of the data in its physical medium without involving other factors such as availability. Returning to the example of your lost shipment of backup tapes, let's say that some of them were encrypted and some of them were not. The principle of possession would enable you to more accurately describe the scope of the incident; the encrypted tapes in the lot cause a possession problem but not a confidentiality problem, while the unencrypted tapes cause a problem on both counts.

Authenticity

The principle of *authenticity* allows you to say whether you've attributed the data in question to the proper owner or creator. For example, if you send an email message that is altered so that it appears to have come from a different email

address than the one from which it was actually sent, you would be violating the authenticity of the email. Authenticity can be enforced using digital signatures, which I'll discuss further in Chapter 5.

A similar, but reversed, concept to this is *nonrepudiation*, which prevents people from taking an action, such as sending an email and then later denying that they have done so. I'll discuss nonrepudiation at greater length in Chapter 4 as well.

Utility

Finally, *utility* refers to how useful the data is to you. Utility is also the only principle of the Parkerian hexad that is not necessarily binary in nature; you can have a variety of degrees of utility, depending on the data and its format. This is a somewhat abstract concept, but it does prove useful in discussing certain situations in the security world.

For instance, in the shipment of backup tapes example, imagine that some of the tapes were encrypted and some were not. For an attacker or other unauthorized person, the encrypted tapes would likely be of very little utility, as the data would not be readable. The unencrypted tapes would be of much greater utility, as the attacker or unauthorized person would be able to access the data.

The concepts discussed in both the CIA triad and the Parkerian hexad provide a practical basis to discuss all the ways in which something can go wrong in the world of

information security. These models enable you to better discuss the attacks that you might face and the types of controls that you need to put in place to combat them.

Attacks

You may face attacks from a wide variety of approaches and angles. You can break these down according to the *type* of attack, the *risk* the attack represents, and the *controls* you might use to mitigate it.

Types of Attacks

You can generally place attacks into one of four categories: interception, interruption, modification, and fabrication. Each of the categories can affect one or more of the principles of the CIA triad, as shown in Figure 1-3.

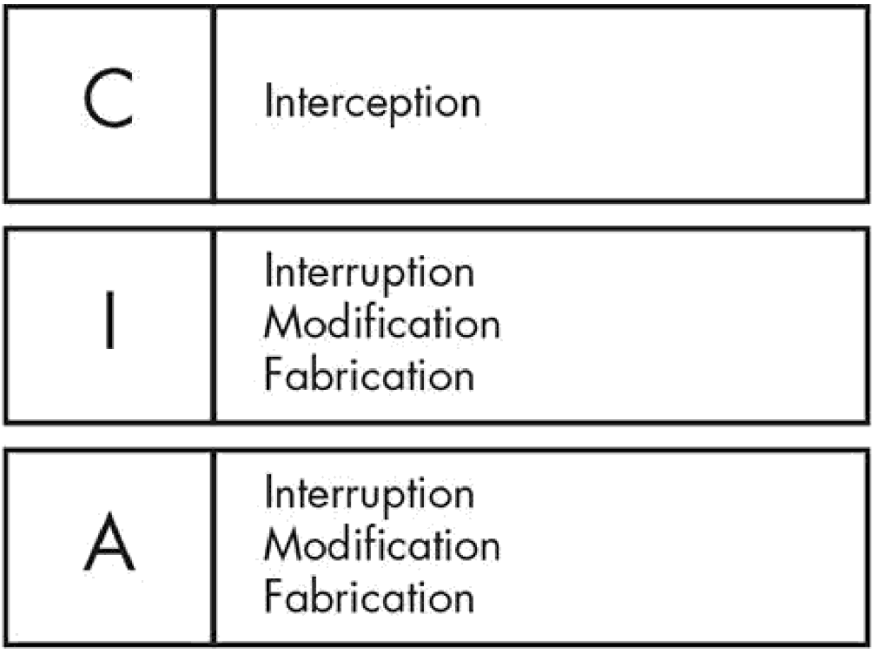

Figure 1-3: The CIA triad and categories of attacks

The line between the categories of attack and the effects they can have are somewhat blurry. Depending on the attack in question, you might include it in more than one category or have more than one type of effect.

Interception

Interception attacks allow unauthorized users to access your data, applications, or environments, and they are primarily attacks against confidentiality. Interception might take the form of unauthorized file viewing or copying, eavesdropping

on phone conversations, or reading someone else's email, and you can conduct it against data at rest or in motion (concepts explained in the "Data at Rest and in Motion" box). When they're properly executed, interception attacks can be difficult to detect.

DATA AT REST AND IN MOTION

You will find, repeatedly throughout this book, that I refer to data being either "at rest" or "in motion," so let's talk about what this means. *Data at rest* is stored data that is not in the process of being moved from one place to another. It may be on a hard drive or flash drive, or it may be stored in a database, for example. This type of data is generally protected with some sort of encryption, often at the level of the file or entire storage device.

Data in motion is data that is moving from one place to another. When you are using your online banking session, the sensitive data flowing between your web browser and your bank is data in motion. Data in motion is also protected by encryption, but in this case the encryption protects the network protocol or path used to move the data from one place to another.

Some may also posit a third category, *data in use*. Data in use would be data that an application or

individual was actively accessing or modifying. Protections on data in use would include permissions and authentication of users. Often you will find the concept of data in use conflated with data in motion. Sound arguments can be made on both sides about whether we should treat this type of data as its own category.

Interruption

Interruption attacks make your assets unusable or unavailable to you on a temporary or permanent basis. These attacks often affect availability but can affect integrity, as well. You would classify a DoS attack on a mail server as an availability attack.

On the other hand, if an attacker manipulated the processes on which a database runs to prevent access to the data it contains, you might consider this an integrity attack because of the possible loss or corruption of data, or you might consider it a combination of the two. You might also consider such a database attack to be a modification attack rather than an interruption attack, as you'll see next.

Modification

Modification attacks involve tampering with an asset. Such attacks might primarily be considered attacks on integrity but

could also represent attacks on availability. If you access a file in an unauthorized manner and alter the data it contains, you've affected the integrity of the file's data. However, if the file in question is a configuration file that manages how a service behaves—perhaps one that is acting as a web server—changing the contents of the file might affect the availability of that service. If the configuration you altered in the file for your web server changes how the server deals with encrypted connections, you could even call this a confidentiality attack.

Fabrication

Fabrication attacks involve generating data, processes, communications, or other similar material with a system. Like the last two attack types, fabrication attacks primarily affect integrity but could affect availability, as well. Generating fake information in a database would be a kind of fabrication attack. You could also generate email, a common method for propagating malware. If you generated enough additional processes, network traffic, email, web traffic, or nearly anything else that consumes resources, you might be conducting an availability attack by rendering the service that handles such traffic unavailable to legitimate users.

Threats, Vulnerabilities, and Risk

To speak more specifically about attacks, I need to introduce a few new terms. When you look at how an attack might

affect you, you can speak of it in terms of threats, vulnerabilities, and the associated risk.

Threats

When I spoke of the types of attacks you might encounter earlier in this chapter, I discussed several types of attacks that could harm your assets—for instance, the unauthorized modification of data. Ultimately, a threat is something that has the potential to cause harm. Threats tend to be specific to certain environments, particularly in the world of information security. For example, although a virus might be problematic on a Windows operating system, the same virus will be unlikely to have any effect on a Linux operating system.

Vulnerabilities

Vulnerabilities are weaknesses, or holes, that threats can exploit to cause you harm. A vulnerability might involve a specific operating system or application that you're running, the physical location of your office building, a data center that is overpopulated with servers and producing more heat than its air-conditioning system can handle, a lack of backup generators, or other factors.

Risk

Risk is the likelihood that something bad will happen. For

you to have a risk in an environment, you need to have both a threat and a vulnerability that the threat could exploit. For example, if you have a structure that is made from wood and you light a fire nearby, you have both a threat (the fire) and a matching vulnerability (the wood structure). In this case, you most definitely have a risk.

Likewise, if you have the same threat of fire but your structure is made of concrete, you no longer have a credible risk because your threat doesn't have a vulnerability to exploit. You could argue that a sufficiently hot flame could damage the concrete, but this is a much less likely event.

We often talk about potential, but unlikely, attacks in computing environments. The best strategy is to spend your time mitigating the most likely attacks. If you sink your resources into trying to plan for every possible attack, however unlikely, you'll spread yourself thin and lack protection where you need it the most.

Impact

Some organizations, such as the US National Security Agency (NSA), add a factor to the threat/vulnerability/risk equation called *impact*. Impact takes into account the value of the asset being threatened and uses it to calculate risk. In the backup tape example, if you consider that the unencrypted tapes contain only your collection of chocolate chip cookie recipes, you may not actually have a risk because the data

exposed contains nothing sensitive and you can make additional backups from the source data. In this case, you might safely say that you have no risk.

Risk Management

Risk management processes compensate for risks in your environment. Figure 1-4 shows a typical risk management process at a high level.

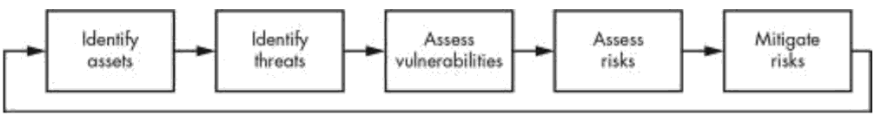

Figure 1-4: A risk management process

As you can see, you need to identify your important assets, figure out the potential threats against them, assess your vulnerabilities, and then take steps to mitigate these risks.

Identify Assets

One of the first and, arguably, most important parts of the risk management process is identifying the assets you're protecting. If you can't enumerate your assets and evaluate the importance of each, protecting them can become a difficult task indeed.

Although this may sound like an exceedingly simple task, it can be a more complex problem than it might seem on the

surface, particularly in larger enterprises. In many cases, an organization might have various generations of hardware, assets from acquisitions of other companies lurking in unknown areas, and scores of unrecorded virtual hosts in use, any of which may be critical to the continued functionality of the business.

Once you've identified the assets in use, deciding which of them are critical business assets is another question entirely. Making an accurate determination of which assets are truly critical to conducting business will generally require the input of functions that make use of the asset, those that support the asset itself, and potentially other involved parties as well.

Identify Threats

After enumerating your critical assets, you can then begin to identify the threats that might affect them. It's often useful to have a framework for discussing the nature of a given threat, and the CIA triad or Parkerian hexad discussed earlier in this chapter serves nicely for this purpose.

For instance, let's apply the Parkerian hexad to examine the threats you might face against an application that processes credit card payments.

Confidentiality If you expose data inappropriately, you could potentially have a breach.

Integrity If data becomes corrupt, you may incorrectly process payments.

Availability If the system or application goes down, you won't be able to process payments.

Possession If you lose backup media, you could potentially have a breach.

Authenticity If you don't have authentic customer information, you may process a fraudulent transaction.

Utility If you collect invalid or incorrect data, that data will have limited utility.

While this is clearly a high-level pass at assessing threats for this system, it does point out a few problem areas immediately. You need to be concerned with losing control of data, maintaining accurate data, and keeping the system up and running. Given this information, you can begin to look at areas of vulnerability and potential risk.

Assess Vulnerabilities

When assessing vulnerabilities, you need to do so in the context of potential threats. Any given asset may have thousands or millions of threats that could impact it, but only a small fraction of these will be relevant. In the previous section, you learned about potential threats against a system that processes credit card transactions.

Let's look at the issues that were identified and attempt to determine whether vulnerabilities exist in any of them.

Confidentiality If you expose data inappropriately, you could have a breach.

Your sensitive data is encrypted at rest and in motion. Your systems are regularly tested by an external penetration testing company. *This is not a risk.*

Integrity If data becomes corrupt, you may incorrectly process payments.

You carefully validate that payment data is correct as part of the processing workflow. Invalid data results in a rejected transaction. *This is not a risk.*

Availability If the system or application goes down, you can't process payments.

You do not have redundancy for the database on the back end of the payment processing system. If the database goes down, you can't process payments. *This is a risk.*

Possession If you lose backup media, you could have a breach.

Your backup media is encrypted and hand-carried by a courier. *This is not a risk.*

Authenticity If you don't have authentic customer information, you may process a fraudulent transaction.

Ensuring that valid payment and customer information belongs to the individual conducting the transaction is difficult. You do not have a good way of doing this. *This is a risk.*

Utility If you collect invalid or incorrect data, that data will have limited utility.

To protect the utility of your data, you checksum credit card numbers, make sure that the billing address and email address are valid, and perform other measures to ensure that your data is correct. *This is not a risk.*

These examples are a high-level view of the process you'd need to undertake, but they serve to illustrate the task. From here, you can again see a few areas of concern, namely, in the areas of authenticity and availability, and you can begin to evaluate the areas in which you may have risks.

Assess Risks

Once you've identified the threats and vulnerabilities for a given asset, you can assess the overall risk. As discussed earlier in this chapter, risk is the conjunction of a threat and a vulnerability. A vulnerability with no matching threat or a threat with no matching vulnerability does not constitute a risk.

For example, the following item was both a potential threat and an area of vulnerability:

Availability If the system or application goes down, you can't process payments.

You don't have redundancy for the database on the back end of your payment processing system, so if the database goes down, you won't be able to process payments.

In this case, you have both a threat and a corresponding vulnerability, meaning you risk losing ability to process credit card payments because of a single point of failure on your database back end. Once you've worked through your threats and vulnerabilities in this manner, you can mitigate these risks.

Mitigate Risks

To mitigate risks, you can put measures in place to account for each threat. These measures are called *controls*. Controls are divided into three categories: physical, logical, and administrative.

Physical controls protect the physical environment in which your systems sit, or where your data is stored. Such controls also control access in and out of such environments. Physical controls include fences, gates, locks, bollards, guards, and cameras, but also systems that maintain the physical environment, such as heating and air-conditioning systems, fire suppression systems, and backup power generators.

Although at first glance physical controls may not seem like they'd be integral to information security, they're one of the most critical controls; if you're not able to physically protect your systems and data, any other controls that you put in place become irrelevant. If attackers can physically access your systems, they can steal or destroy them, rendering them unavailable for your use—in the best case. In the worst case, attackers will be able to access your applications and data directly and steal your information and resources or subvert them for their own use.

Logical controls, sometimes called *technical controls*, protect the systems, networks, and environments that process, transmit, and store your data. Logical controls can include items such as passwords, encryption, access controls, firewalls, and intrusion detection systems.

Logical controls enable you to prevent unauthorized activities; if your logical controls are implemented properly and are successful, an attacker or unauthorized user can't access your applications and data without subverting the controls.

Administrative controls are based on rules, laws, policies, procedures, guidelines, and other items that are "paper" in nature. Administrative controls dictate how the users of your environment should behave. Depending on the environment and control in question, administrative controls can represent differing levels of authority. You may have a simple rule such

as "turn the coffee pot off at the end of the day," aimed at avoiding a physical security problem (burning your building down at night). You may also have a more stringent administrative control, such as one that requires you to change your password every 90 days.

One important part of administrative controls is the ability to enforce them. If you don't have the authority or the ability to ensure that people comply with your controls, they are worse than useless because they create a false sense of security. For example, if you create a policy that says employees can't use business resources for personal use, you'll need to be able to enforce this. Outside of a highly secure environment, this can be a difficult task. You'd need to monitor telephone and mobile phone usage, web access, email use, instant message conversations, installed software, and other potential areas for abuse. Unless you were willing to devote a great deal of resources to monitoring these and handling violations of policy, you'd quickly have a policy that you wouldn't be able to enforce. The next time you're audited and asked to produce evidence of policy enforcement, you'll face issues.

Incident Response

If your risk management efforts are not as thorough as you hoped or you're blindsided by something entirely unexpected, you can react with incident response. You should

direct your incident response at the items that you feel are most likely to cause your organization pain. You should have already identified these as part of your risk management efforts.

As much as possible, you should base your reaction to such incidents on documented incident response plans, which should be regularly reviewed, tested, and practiced by those who will be expected to enact them in the case of an actual incident. You don't want to wait until an actual emergency to find out documentation that has been languishing on a shelf is outdated and refers to processes or systems that have changed heavily or no longer exist.

The incident response process, at a high level, consists of the following:

- Preparation
- Detection and analysis
- Containment
- Eradication
- Recovery
- Post-incident activity

I'll go over these phases in more detail next.

Preparation

The preparation phase of incident response consists of all the activities you can perform ahead of time to better handle an incident. This typically involves creating policies and procedures that govern incident response and handling, conducting training and education for both incident handlers and those who are expected to report incidents, and developing and maintaining documentation.

You shouldn't underestimate the importance of this phase of incident response. Without adequate preparation, it is extremely unlikely that the response to an incident will go well or according to your unpracticed plans. The time to determine what needs to be done, who needs to do it, and how to do it is not when you're faced with an emergency.

Detection and Analysis

The detection and analysis phase is where the action begins. In this phase, you detect an issue, decide whether it's actually an incident, and respond to it appropriately.

Most often, you'll detect the issue with a security tool or service, like an intrusion detection system (IDS), antivirus (AV) software, firewall logs, proxy logs, or alerts from a security information and event monitoring (SIEM) tool or managed security service provider (MSSP).

The analysis portion of this phase is often a combination of automation from a tool or service, usually a SIEM tool, and human judgment. While you can often use some sort of

thresholding to say that a certain number of events in a given amount of time is normal or that a certain combination of events is not normal (two failed logins, followed by a success, a password change, and the creation of a new account, for instance), you'll often want human intervention at a some point. Human intervention might include a review of logs output by various security, network, and infrastructure devices; contact with the party who reported the incident; and general evaluation of the situation. (Unfortunately for the incident handler, these situations often occur at 4 PM on a Friday or 2 AM on a Sunday.)

When the incident handler evaluates the situation, that person will decide whether the issue constitutes an incident, evaluate the criticality of the incident, and contact any additional resources needed to proceed to the next phase.

Containment, Eradication, and Recovery

The containment, eradication, and recovery phase is where most of the work to solve the incident takes place, at least in the short term.

Containment involves taking steps to ensure that the situation doesn't cause any more damage than it already has —or at least lessen any ongoing harm. If the problem involves a malware-infected server actively being controlled by a remote attacker, this might mean disconnecting the server from the network, putting firewall rules in place to

block the attacker, and updating signatures or rules on an intrusion prevention system (IPS) to halt the traffic from the malware.

During *eradication*, you'll attempt to remove the effects of the issue from your environment. In the case of your malware-infected server, you've already isolated the system and cut it off from its command-and-control network. Now you'll need to clean the malware from the server and ensure that it doesn't exist elsewhere in your environment. This might involve additional scanning of other hosts in the environment to ensure that the malware is not present and perhaps examining logs on the server and network to determine what other systems the infected server has communicated with. With malware, particularly very new malware or variants, this can be a tricky task. Whenever you're in doubt about whether you've truly evicted malware or attackers from your environment, you should err on the side of caution.

Lastly, you need to recover the state you were in prior to the incident. *Recovery* might involve restoring devices or data from backup media, rebuilding systems, or reloading applications. Again, this can be a more painful task than it initially seems because your knowledge of the situation might be incomplete or unclear. You may find that you are unable to verify that backup media is clean and free or infection or that the backup media is entirely bad. Application install bits

may be missing, configuration files may not be available, or many other issues could occur.

Post-Incident Activity

Like preparation, post-incident activity is easy to overlook, but you should ensure that you don't neglect it. In the post-incident activity phase, often referred to as a *post-mortem* (Latin for "after death"), you attempt to determine specifically what happened, why it happened, and what you can do to keep it from happening again. The purpose of this phase is not to point fingers or place blame (although this does sometimes happen) but to ultimately prevent or lessen the impact of future such incidents.

Defense in Depth

Now that you've learned about the potential effects of a security breach, the kinds of attacks you might face, and the strategies for dealing with these attacks, I'll introduce you to a method of working toward preventing these attacks. *Defense in depth* is a strategy common to both military maneuvers and information security. The basic concept is to formulate a multilayered defense that will allow you to still mount a successful resistance should one or more of your defensive measures fail.

In Figure 1-5, you can see an example of layers you might

want to put in place to defend your assets.

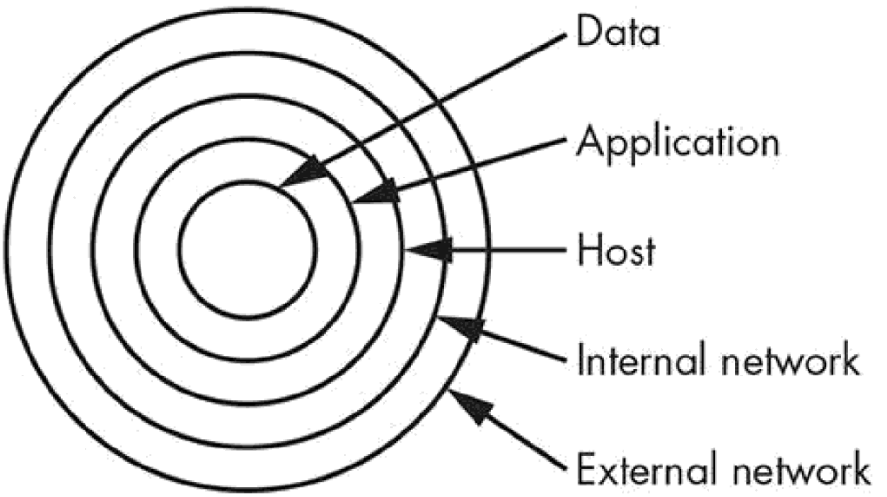

Figure 1-5: Defense in depth

At the least, you would want defenses at the external network, internal network, host, application, and data levels. Well-implemented defenses at each layer make it difficult to successfully penetrate your network and attack your assets directly.

That said, defense in depth is not a magic bullet. No matter how many layers you put in place or how many defensive measures you place at each layer, you won't be able to keep every attacker out for an indefinite period. Nor is this the goal of defense in depth in an information security setting. The goal is to place enough defensive measures between your truly important assets and the attacker so that

you'll notice that an attack is in progress and have enough time to prevent it.

An example of such a delaying tactic is requiring employees to change their passwords every 60 or 90 days. This makes it harder for an attacker to crack a password in time to still use it.

Using stringent password construction rules is another delaying tactic. Consider the password "mypassword," which is ten characters long and uses only one character set. Using a relatively slow off-the-shelf system, an attacker might take a week or two to crack this password. With a purpose-built password cracking system or a botnet, an attacker might take only an hour or two.

If you use more secure password construction rules and go with a password along the lines of MyP@ssword1, which is also ten characters long but uses four character sets, cracking the password would take thousands of years on purpose-built hardware and upward of several years for a large botnet.

If you require employees to both change their passwords frequently and create complex passwords, an attacker won't be able to crack one in time to use it.

ENTROPY IN PASSWORDS

The complex password example discussed previously

uses a classic strong password construction scheme, consisting of eight or more characters and comprising multiple character sets (upper alpha, lower alpha, numbers, and punctuation). Some would argue it contains insufficient entropy (unpredictability) to be truly secure and that you'd be better served with a longer, more entropic, and more easily remembered password like correcthorsebatterystaple.[4]

Ultimately, your primary concern should be in constructing reasonably secure passwords and changing them at regular intervals.

The layers you include in your defense-in-depth strategy will vary given the situation and environment you're defending. As discussed, from a strictly logical (nonphysical) information security perspective, you'd want to look at the external network, network perimeter, internal network, host, application, and data layers as areas to place your defenses.

You could add complexity to your defensive model by including other vital layers, such as physical defenses, policies, or user awareness and training, but I'll stick with a simpler example for the time being.

Table 1-1 lists some of the defenses you might use for each of the layers discussed.

Table 1-1: Defense by Layer

Layer	Defensive measures
External network	DMZ VPN Logging Auditing Penetration testing Vulnerability analysis
Network perimeter	Firewalls Proxy Logging Stateful packet inspection Auditing Penetration testing Vulnerability analysis
Internal network	IDS IPS Logging Auditing Penetration testing Vulnerability analysis
Host	Authentication Antivirus Firewalls IDS IPS Passwords Hashing

	Logging
	Auditing
	Penetration testing
	Vulnerability analysis
Application	SSO
	Content filtering
	Data validation
	Auditing
	Penetration testing
	Vulnerability analysis
Data	Encryption
	Access controls
	Backups
	Penetration testing
	Vulnerability analysis

In some cases, a defensive measure appears in multiple layers because it applies to more than one area. A good example of this is *penetration testing*, a method of finding gaps in your security by using some of the same strategies an attacker would use to break in, which appears in every layer. I'll discuss this in greater depth in Chapter 14. You might want to use penetration testing at every layer of your defense. You can also see where specific controls may be tied to particular layers, such as firewalls and proxies at the network perimeter. As with everything else in the security field, you could argue that some or all of these controls could exist at layers other than what is shown here, but this is a good

general guideline. As you move through the book, I'll discuss each of these areas shown in Table 1-1 in greater detail, as well as the specific defenses you might want to use for each.

Summary

When discussing issues pertaining to information security, such as attacks and controls, it's helpful to have a model by which to do so. This chapter discussed two potential models: the CIA triad, composed of confidentiality, integrity, and availability; and the Parkerian hexad, composed of confidentiality, integrity, availability, possession or control, authenticity, and utility.

As you look toward preventing attacks, it is also helpful to understand the general categories of damage that you might see occur in the event of an attack. Attacks may impact environments through interception, interruption, modification, or fabrication. Each of these effects would impact particular areas of the CIA triad.

When discussing specific threats you might face, it's important to understand the concept of risk. You only face risk from an attack when a threat is present and you have a vulnerability that threat can exploit. To mitigate risk, you use three main types of controls: physical, logical, and administrative.

Finally, this chapter covered defense in depth, a

particularly important concept in the world of information security. To build defensive measures using this concept, you put in place multiple layers of defense to delay an attacker long enough to alert you to the attack and to allow you to mount a more active defense.

The concepts discussed in this chapter are foundational to information security. They're used on a regular basis during normal information security tasks in many organizations; you might hear someone talking about breaches of confidentiality, for example, or the authenticity of a given email message.

Information security is a daily concern for organizations of any size, particularly those that handle any type of personal information, financial data, healthcare data, educational data, or other types of information regulated by the laws of the country in which the organization operates. When an organization doesn't invest in information security, the repercussions can be severe. They might face fines, lawsuits, or even the inability to continue conducting business if they lose control of critical or sensitive data. In short, information security is a key component of the modern business world.

Exercises

Here are some questions to help you review the key concepts of this chapter:

1. Explain the difference between a vulnerability and a threat.
2. What are six items that might be considered logical controls?
3. What term might you use to describe the usefulness of data?
4. Which category of attack is an attack against confidentiality?
5. How do you know at what point you can consider your environment to be secure?
6. Using the concept of defense in depth, what layers might you use to secure yourself against someone removing confidential data from your environment on a USB flash drive?
7. Based on the Parkerian hexad, what principles are affected if you lose a shipment of encrypted backup tapes that contain personal and payment information for your customers?
8. If the web servers in your environment are based on Microsoft's Internet Information Services (IIS) and a new worm is discovered that attacks Apache web servers, what do you not have?
9. If you develop a new policy for your environment that requires you to use complex and automatically

generated passwords that are unique to each system and are a minimum of 30 characters in length, such as "!Qa4(j0nO$&xn1%2AL34ca#!Ps321$," what will be adversely impacted?

10. Considering the CIA triad and the Parkerian hexad, what are the advantages and disadvantages of each model?

2

IDENTIFICATION AND AUTHENTICATION

When you're developing security measures, whether they're specific mechanisms or entire infrastructures, identification and authentication are key concepts. In short, *identification* makes a claim about what someone or something is, and *authentication* establishes whether this claim is true. You can see such processes taking place daily in a wide variety of ways.

One common example of an identification and

authentication transaction is the use of payment cards that require a personal identification number (PIN). When you swipe the magnetic strip on the card, you're asserting that you're the person indicated on the card. At this point, you've given your identification, but nothing more. When you're prompted to enter the PIN associated with the card, you're completing the authentication portion of the transaction, proving you're the legitimate cardholder.

Some of the identification and authentication methods that we use daily are particularly fragile, meaning they depend largely on the honesty and diligence of those involved in the transaction. If you show your ID card to buy alcohol, for example, you're asking people to trust that your ID is genuine and accurate; they can't authenticate it unless they have access to the system that maintains the ID in question. We also depend on the competence of the person or system performing the authentication; they must be capable not only of performing the act of authentication but also of detecting false or fraudulent activity.

You can use several methods for identification and authentication, from requiring simple usernames and passwords to implementing purpose-built hardware tokens that serve to establish your identity in multiple ways. In this chapter, I'll discuss several of these methods and explore their uses.

Identification

Identification, as you just learned, is simply an assertion of who we are. This may include who we claim to be as people, who a system claims to be over the network, or who the originating party of an email claims to be. You'll see some methods for determining identity and examine how trustworthy those methods are.

Who We Claim to Be

Who we claim to be is a tenuous concept at best. We can identify ourselves by our full names, shortened versions of our names, nicknames, account numbers, usernames, ID cards, fingerprints, or DNA samples. Unfortunately, with a few exceptions, such methods of identification are not unique, and even some of the supposedly unique methods of identification, such as fingerprints, can be duplicated.

Who we claim to be can, in many cases, be subject to change. For instance, women often change their last names upon getting married. In addition, we can generally change logical forms of identification—such as account numbers or usernames—easily. Even physical identifiers, such as height, weight, skin color, and eye color, can change. One of the most crucial factors to realize is that a claim of identity alone is not enough.

Identity Verification

Identity verification is a step beyond identification, but it's still a step short of authentication, which I'll discuss in the next section. When you're asked to show a driver's license, Social Security card, birth certificate, or other similar form of identification, this is generally for identity verification, not authentication. It's the rough equivalent of someone claiming the identity John Smith; you asking if the person is indeed John Smith and being satisfied with an answer of "Sure, I am" from the person (plus a little paperwork).

We can take the example a bit further and validate the form of identification (say, a passport) against a database holding an additional copy of the information that it contains, matching the photograph and physical specifications with the person standing in front of us. This may get us a bit closer to ensuring we've correctly identified the person, but it still doesn't qualify as authentication; we may have validated the status of the ID itself, and we know that the person meets the general specifications of the person it was originally issued to, but we've taken no steps to prove that the person is really the right one. The more than we trend toward verification and away from authentication, the weaker our controls are.

Computer systems use identity verification, too. When you send an email, the identity you provide is taken to be

true; the system rarely takes any additional steps to authenticate you. Such gaps in security contribute to the enormous amount of spam traffic, which Cisco's Talos Intelligence Group estimated to have accounted for approximately 85 percent of all emails sent from mid-2017 to mid-2018.[1]

Falsifying Identification

As I've discussed, methods of identification are subject to change. As such, they are also subject to falsification. Minors often use fake IDs to get into bars or nightclubs, while criminals and terrorists might use them for a variety of more nefarious tasks. You could use some methods of identification, such as birth certificates, to gain additional forms of identification, such as Social Security cards or driver's licenses, thus strengthening a false identity.

Identity theft based on falsified information is a major concern today; identity thieves stole an estimated $16.8 billion from US consumers in 2017.[2] This type of attack is unfortunately common and easy to execute. Given a minimal amount of information—usually a name, address, and Social Security number are sufficient—it's possible to impersonate someone just enough to be able to conduct a variety of transactions in their name, such as opening a line of credit. Such crimes occur because many activities lack authentication

requirements. Although most people think identity verification is sufficient, verification is easy to circumvent by using falsified forms of identification.

Many of the same difficulties exist in computer systems and environments. For example, it's entirely possible to send an email from a falsified email address. Spammers use this tactic on a regular basis. I'll address such issues at greater length in Chapter 9.

Authentication

In information security, authentication is the set of methods used to establish whether a claim of identity is true. Note that authentication does not decide what the party being authenticated is permitted to do; this is a separate task, known as *authorization*. I'll discuss authorization in Chapter 3.

Factors

There are several approaches to authentication: something you know, something you are, something you have, something you do, and where you are. These approaches are known as *factors*. When you're attempting to authenticate a claim of identity, you'll want to use as many factors as possible. The more factors you use, the more positive your results will be.

Something you know, a common authentication factor, includes passwords or PINs. However, this factor is somewhat weak, because if the information the factor depends on is exposed, your authentication method may no longer be unique.

Something you are is a factor based on the relatively unique physical attributes of an individual, often referred to as *biometrics*. Although biometrics can include simple attributes such as height, weight, hair color, or eye color, these aren't usually distinctive enough to make very secure identifiers. Complex identifiers such as fingerprints, iris or retina patterns, or facial characteristics are more common. These are a bit stronger than, say, a password, because forging or stealing a copy of a physical identifier is somewhat more difficult, although not impossible. There is some question as to whether biometrics truly count as an authentication factor or whether they only constitute verification. I'll discuss this again later in this chapter, when I cover biometrics in greater depth.

Something you have is a factor generally based on a physical possession, although it can extend into some logical concepts. Common examples are automatic teller machine (ATM) cards, state or federally issued identity cards, or software-based security tokens, as shown in Figure 2-1.[3] Some institutions, such as banks, have begun to use access to logical devices, such as cell phones or email accounts, as methods of

authentication, as well.

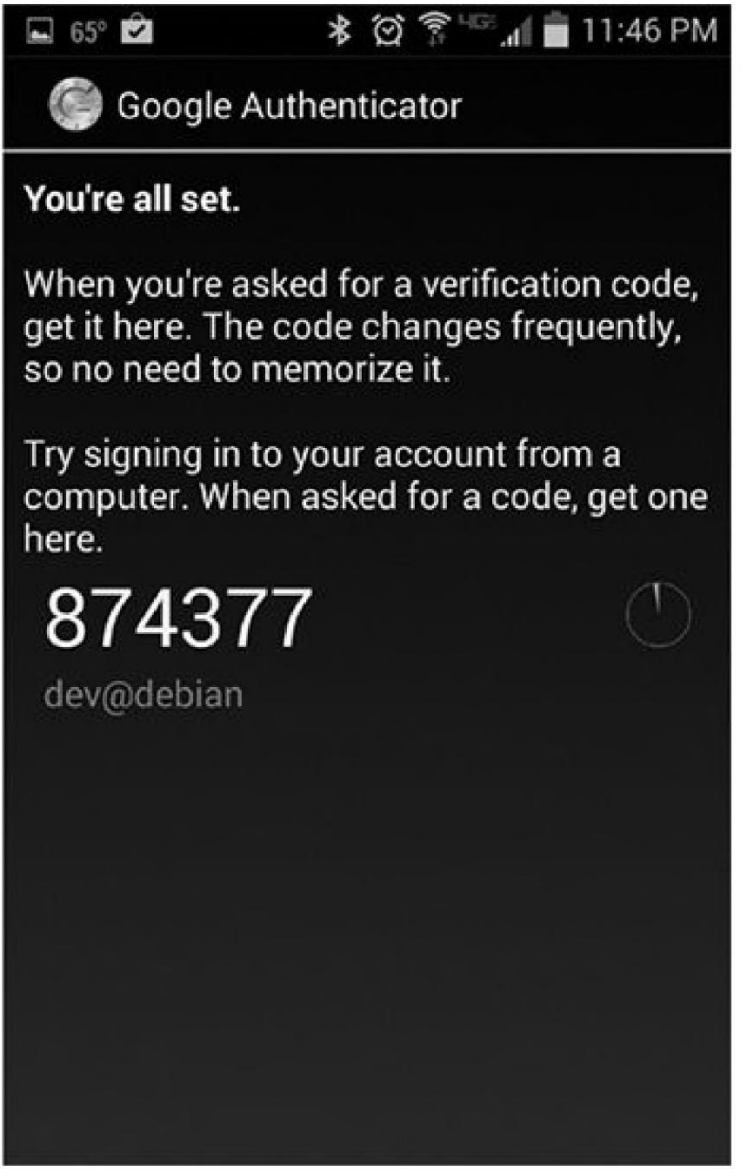

Figure 2-1: Sending a security token to a mobile phone is a

common authentication method.

This factor can vary in strength depending on the implementation. If you wanted to use a security token sent to a device that doesn't belong to you, you'd need to steal the device to falsify the authentication method. On the other hand, if the security token was sent to an email address, it would be much easier to intercept, and you'd have a measure of considerably less strength.

Something you do, sometimes considered a variation of something you are, is a factor based on the actions or behaviors of an individual. This may include an analysis of the individual's gait or handwriting or of the time delay between keystrokes as he or she types a passphrase. These factors present a strong method of authentication and are difficult to falsify. They do, however, have the potential to incorrectly reject legitimate users at a higher rate than some of the other factors.

Where you are is a geographically based authentication factor. This factor operates differently than the other factors, as it requires a person to be present in a specific location. For example, when changing an ATM PIN, most banks will require you to go into a branch, at which point you will also be required to present your identification and account number. If the bank allowed the PIN to be reset online, an attacker could change your PIN remotely and proceed to

clean out your account. Although potentially less useful than some of the other factors, this factor is difficult to counter without entirely subverting the system performing the authentication.

Multifactor Authentication

Multifactor authentication uses one or more of the factors discussed in the preceding section. When you're using only two factors, this practice is also sometimes called *two-factor authentication*.

Let's return to the ATM example because it illustrates multifactor authentication well. In this case, you use something you know (your PIN) and something you have (your ATM card). Your ATM card serves as both a factor for authentication and a form of identification. Another example of multifactor authentication is writing checks. In this case, you're using something you have (the checks themselves) and something you do (signing them). Here, the two factors involved in writing a check are rather weak, so you sometimes see a third factor—a fingerprint—used with them.

Depending on the factors selected, you can assemble stronger or weaker multifactor authentication schemes particular to each situation. In some cases, although certain methods may be more difficult to defeat, they're not practical to implement. For example, DNA makes for a strong method

of authentication but isn't practical in most situations. In Chapter 1, I said that your security should be proportional to what you're protecting. You certainly could install iris scanners on every credit card terminal, but this would be expensive, impractical, and potentially upsetting to customers.

Mutual Authentication

Mutual authentication is an authentication mechanism in which both parties in a transaction authenticate each other. These parties are typically software-based. In the standard, one-way authentication process, the client authenticates to the server. In mutual authentication, not only does the client authenticate to the server, but the server authenticates to the client. Mutual authentication often relies on digital certificates, which I'll discuss in Chapter 5. Briefly, both the client and the server would have a certificate to authenticate the other.

In cases where you don't perform mutual authentication, you leave yourself open to impersonation attacks, often referred to as *man-in-the-middle attacks*. In a man-in-the-middle attack, the attacker inserts himself between the client and the server. The attacker then impersonates the server to the client and the client to the server, as shown in Figure 2-2, by circumventing the normal pattern of traffic and then intercepting and forwarding the traffic that would normally

flow directly between the client and the server.

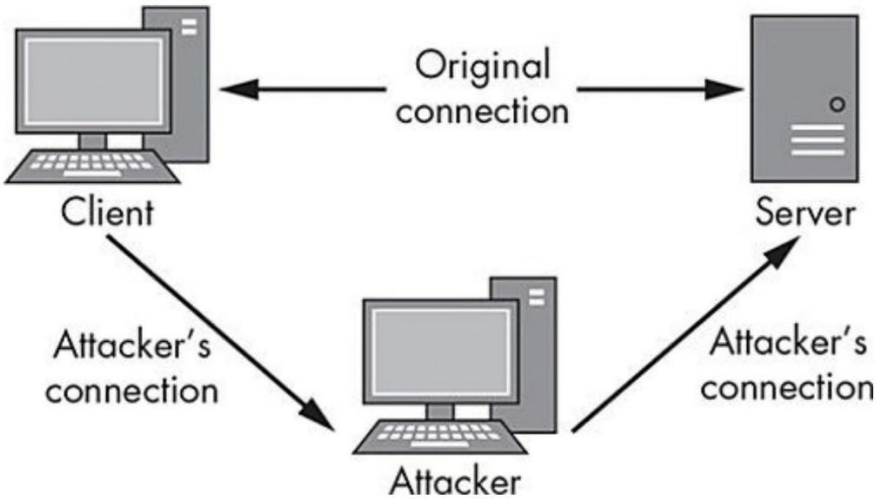

Figure 2-2: A man-in-the-middle attack

This is typically possible because the attacker needs to subvert or falsify authentication only from the client to the server. If you implement mutual authentication, this becomes a considerably more difficult attack because the attacker would have to falsify two different authentications.

You can also combine mutual authentication with multifactor authentication, although the latter generally takes place only on the client side. Multifactor authentication from the server back to the client would be not only technically challenging but also impractical in most environments because it would involve some technical heavy-lifting on the client side, potentially on the part of the user. You'd likely

lose a significant amount of productivity.

Common Identification and Authentication Methods

I'll conclude this discussion by exploring three common identification and authentication methods in detail: passwords, biometrics, and hardware tokens.

Passwords

Passwords are familiar to most us who use computers regularly. When combined with a username, a password will generally allow you access to a computer system, an application, a phone, or a similar device. Although they're only a single factor of authentication, passwords can represent a relatively high level of security when constructed and implemented properly.

People often describe certain passwords as being *strong*, but a better descriptive term might be *complex*. If you construct a password that uses lowercase letters only and is eight characters long, you can use a password-cracking utility to crack it quickly, as discussed in Chapter 1. Adding character sets to the password makes it increasingly harder to figure out. If you use uppercase letters, lowercase letters, numbers, and symbols, you'll end up with a password that is

potentially more difficult to remember, such as *$sU&qw!3*, but much harder to crack.

In addition to constructing strong passwords, you also need to practice good password hygiene. Don't write your password down and post it under your keyboard or on your monitor; doing so completely defeats the purpose of having a password in the first place. Applications called *password managers* exist to help us manage all the logins and passwords we have for different accounts, some as locally installed software and others as web or mobile device applications. There are many arguments for and against such tools; some people think keeping all of your passwords in one place is a bad idea, but when used carefully, they can help you maintain good password hygiene.

Another common problem is the manual synchronization of passwords—in short, using the same password everywhere. If you use the same password for your email, for your login at work, and for your online knitting discussion forum, you're putting the security of all the accounts in the hands of those system owners. If any one of them is compromised, all of your accounts become vulnerable; all an attacker needs to do to access the others is look up your account name on the internet to find your other accounts and log in using your default password. By the time the attacker gets into your email account, the game is over because an attacker can generally use it reset account credentials for any other

accounts you have.

Biometrics

Although some biometric identifiers may be more difficult to falsify than others, this is only because of the limitations of today's technology. At some point in the future, we'll need to develop more robust biometric characteristics to measure or else stop using biometrics as an authentication mechanism.

Using Biometrics

Biometrics-equipped devices are becoming increasingly common and inexpensive. You can find a wide selection of them for less than $20. It pays to research such devices carefully before you depend on them for security, as some of the cheaper versions are easy to bypass.

You can use biometric systems in two ways. You can use them to verify the identity claim someone has put forth, as discussed earlier, or you can reverse the process and use biometrics as a method of identification. This process is commonly used by law enforcement agencies to identify the owner of fingerprints left on various objects. It can be a time-consuming effort, considering the sheer size of the fingerprint libraries held by such organizations. To use a biometric system in either manner, you need to put the user through some sort of enrollment process. Enrollment involves recording the user's chosen biometric characteristic

—for instance, making a copy of a fingerprint—and saving it in a system. Processing the characteristic may also include noting elements that appear at certain parts of the image, known as *minutiae* (Figure 2-3).

Figure 2-3: Biometric minutiae

You can use the minutiae later to match the characteristic

to the user.

Characteristics of Biometric Factors

Biometric factors are defined by seven characteristics: universality, uniqueness, permanence, collectability, performance, acceptability, and circumvention.[4]

Universality means you should be able to find your chosen biometric characteristic in the majority of people you expect to enroll in the system. For instance, although you might be able to use a scar as an identifier, you can't guarantee that everyone will have a scar. Even if you choose a common characteristic, such as a fingerprint, you should take into account the fact that some people may not have an index finger on their right hand and be prepared to compensate for this.

Uniqueness is a measure of how unique a characteristic is among individuals. For example, if you choose to use height or weight as a biometric identifier, you'd stand a good chance of finding several people in any given group who have the same height or weight. You should try to select characteristics with a high degree of uniqueness, such as DNA or iris patterns, but even these could be duplicated, whether intentionally or otherwise. For example, identical twins have the same DNA, and an attacker could replicate a fingerprint.

Permanence tests how well a characteristic resists change over time and with advancing age. If you choose a factor that can easily vary, such as height, weight, or hand geometry, you'll eventually find yourself unable to authenticate a legitimate user. It's better to use factors such as fingerprints, which are unlikely to change without deliberate action.

Collectability measures how easy it is to acquire a characteristic. Most commonly used biometrics, such as fingerprints, are relatively easy to acquire, which is one reason they are common. On the other hand, a DNA sample is more difficult to acquire because the user must provide a genetic sample to enroll and to authenticate again later.

Performance measures how well a given system functions based on factors such as speed, accuracy, and error rate. I'll discuss the performance of biometric systems at greater length later in this section.

Acceptability is a measure of how acceptable the characteristic is to the users of the system. In general, systems that are slow, difficult to use, or awkward to use are less likely to be acceptable to the user.[5] Systems that require users to remove their clothes, touch devices that have been repeatedly used by others, or provide tissue or bodily fluids are unlikely to have a high degree of acceptability.

Circumvention describes how easy it is to trick a system by using a falsified biometric identifier. The classic example of a

circumvention attack against the fingerprint as a biometric identifier is the "gummy finger." In this type of attack, a fingerprint is lifted from a surface and used to create a mold with which the attacker can cast a positive image of the fingerprint in gelatin. Some biometric systems have secondary features specifically designed to defeat such attacks by measuring skin temperature, pulse, or pupillary response.

Measuring Performance

There are many ways to measure the performance of a biometric system, but a few primary metrics are particularly important. The *false acceptance rate (FAR)* and *false rejection rate (FRR)* are two of these.[6] FAR measures how often you accept a user who should be rejected. This is also called a *false positive*. FRR measures how often we reject a legitimate user and is sometimes called a *false negative*.

You want to avoid both of these situations in excess. You should aim for a balance between the two error types, referred to as an *equal error rate (EER)*. If you plot both the FAR and the FRR on a graph, as I've done in Figure 2-4, the EER marks the point where the two lines intersect. We sometimes use EER as a measure of the accuracy of biometric systems.

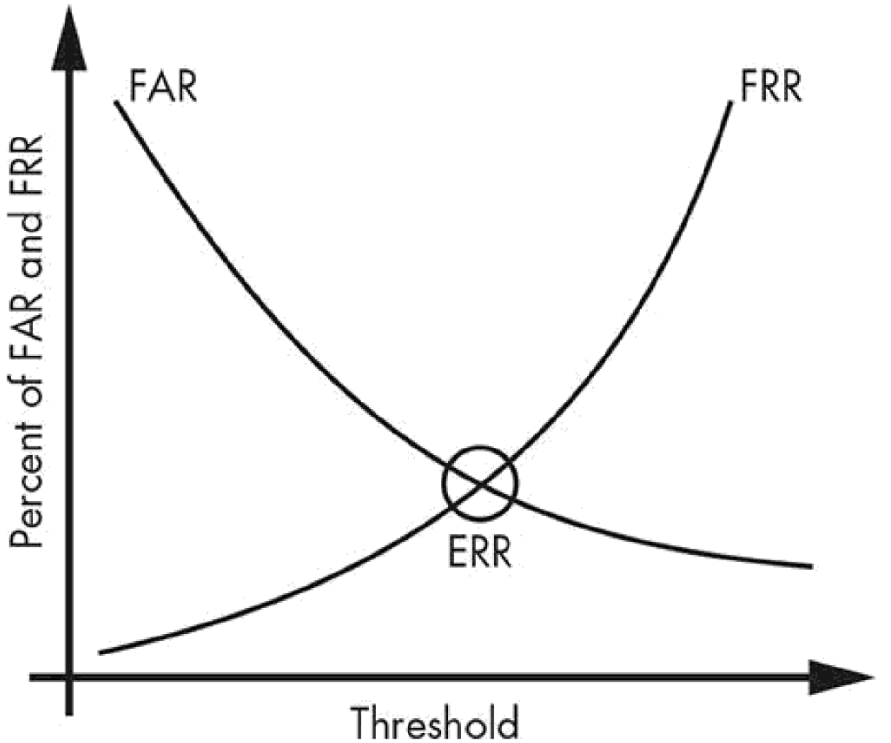

Figure 2-4: The equal error rate is the intersection of the false acceptance rate and false rejection rate.

Flaws in Biometric Systems

Biometric systems are prone to several common issues. As I mentioned when discussing circumvention, it's easy to forge some biometric identifiers. Moreover, once they're forged, it's hard to re-enroll a user in the system. For example, if you enroll a user with both index fingers and those fingerprints get compromised, you could remove these from the system

and enroll two of their other fingers. However, if you've already enrolled all of their fingers in the system, you'd have no means of re-enrolling them using fingers at all. Depending on the system in question, you may be able to select a different set of minutiae for the same identifier, but this avoids the point of the discussion, which is that biometric identifiers are finite. This issue became tangible in 2015, when an attacker hacked the US Office of Personnel Management and stole the fingerprint records of 5.6 million federal employees holding security clearances.[7]

You also face possible privacy issues in the use of biometrics. When you're enrolled in a biometric system, you're essentially giving away a copy of the identifier, whether it's a fingerprint, iris pattern, or DNA sample. Once such an item has been entered into a computer system, you have little, if any, control over what happens to it. We can hope that once you're no longer associated with the institution in question, the institution would destroy such materials, but you have no way to guarantee this. Particularly in the case of DNA sampling, the repercussions of surrendering genetic material could affect you for the rest of your life.

Hardware Tokens

A standard hardware token (Figure 2-5) is a small device,

typically in the general form factor (size and shape) of a credit card or keychain fob.[8] The simplest hardware tokens look identical to universal serial bus (USB) flash drives and contain a certificate or unique identifier. They're often called *dongles*. More complex hardware tokens incorporate liquid-crystal displays (LCDs), keypads for entering passwords, biometric readers, wireless devices, and additional features to enhance security.

Figure 2-5: A hardware token

Many hardware tokens contain an internal clock that generates a code based on the device's unique identifier, an input PIN or password, and other potential factors. Usually,

the code is output to a display on the token and changes on a regular basis, often every 30 seconds. The infrastructure used to keep track of these tokens can predict what the proper output will be at any given time in order to authenticate the user.

The simplest kind of hardware token represents only the something you have factor and is thus susceptible to theft and potential use by a knowledgeable criminal. Although these devices represent an increased level of security for the user's accounts and aren't generally useful without the associated account credentials, you do need to remember to safeguard them.

More sophisticated hardware tokens could represent the something you know or something you are factors, as well. They might require a PIN or fingerprint, which enhances the security of the device considerably; in addition to getting the hardware token, an attacker would need to either subvert the infrastructure that uses the device or extract the something you know or something you are factor from the legitimate owner of the device.

Summary

Identification is an assertion of the identity of some party, whether it be a person, process, system, or other entity. Identification is only a claim of identity; it doesn't say

anything about any privileges that might be associated with the identity.

Authentication is the process used to validate whether the claim of identity is correct. It's different than verification, which is a much weaker way of testing someone's identity.

When you perform authentication, you can use several factors. The main factors are something you know, something you are, something you have, something you do, and where you are. An authentication mechanism that includes more than one factor is known as multifactor authentication. Using multiple factors gives you a much stronger authentication mechanism than you might otherwise have.

The common set of tools used for authentication includes passwords, tokens, and biometric identifiers. Each of these has its own set of unique challenges that you will need to deal with when you are implementing them as part of your set of security controls.

In the next chapter, I'll discuss the steps that take place after identification and authentication: authorization and access control.

Exercises

1. What is the difference between verification and

authentication of an identity?

2. How do you measure the rate at which you fail to authenticate legitimate users in a biometric system?
3. What do you call the process in which the client authenticates to the server and the server authenticates to the client?
4. A key would be described as which type of authentication factor?
5. What biometric factor describes how well a characteristic resists change over time?
6. If you're using an identity card as the basis for your authentication scheme, what steps might you add to the process to allow you to move to multifactor authentication?
7. If you're using an eight-character password that contains only lowercase characters, would increasing the length to ten characters represent any significant increase in strength? Why or why not?
8. Name three reasons why an identity card alone might not make an ideal method of authentication.
9. What factors might you use when implementing a multifactor authentication scheme for users who are logging onto workstations that are in a secure environment and are used by more than one person?

10. If you're developing a multifactor authentication system for an environment where you might find larger-than-average numbers of disabled or injured users, such as a hospital, which authentication factors might you want to use or avoid? Why?

3

AUTHORIZATION AND ACCESS CONTROLS

After you've received a party's claim of identity and established whether that claim is valid, as discussed in Chapter 2, you have to decide whether to allow the party access to your resources. You can achieve this with two main concepts: authorization and access control. *Authorization* is the process of determining exactly what an authenticated

party can do. You typically implement authorization using *access controls*, which are the tools and systems you use to deny or allow access.

You can base access controls on physical attributes, sets of rules, lists of individuals or systems, or other, more complex factors. When it comes to logical resources, you'll probably find simple access controls implemented in everyday applications and operating systems and elaborate, multilevel configurations in military or government environments. In this chapter, you'll learn about access controls in more detail and look at some ways of implementing them.

What Are Access Controls?

Although the term *access controls* may sound technical, like it belongs only in high-security computing facilities, we all deal with access controls daily.

- When you lock or unlock the doors of your house, you're using a form of physical access control, based on your keys. (Your keys are something you have, as discussed in Chapter 2; in this case, they function as methods of both authentication and authorization.)
- When you start your car, you're also likely to use a key. For some newer cars, your key may even include an extra layer of security with radio-frequency

identification (RFID) tags, which are certificate-like identifiers stored on the key.

- Upon reaching your place of employment, you might use a badge (again, something you have) to enter the building.
- When you sit down in front of your computer at work and enter your password (something you know), you're authenticating yourself and using a logical access control system to access the resources for which you've been given permission.

Most of us regularly encounter multiple implementations like these while working, going to school, and performing the other activities that make up our day.

You'll probably want to use access controls to carry out four basic tasks: allowing access, denying access, limiting access, and revoking access. We can describe most access control issues or situations using these four actions.

Allowing access is giving a party access to a given resource. For example, you might want to give a user access to a file, or you may want to give an entire group of people access to all the files in a given directory. You might also allow someone physical access to a resource by giving your employees a key or badge to your facility.

Denying access is the opposite of granting access. When you