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.

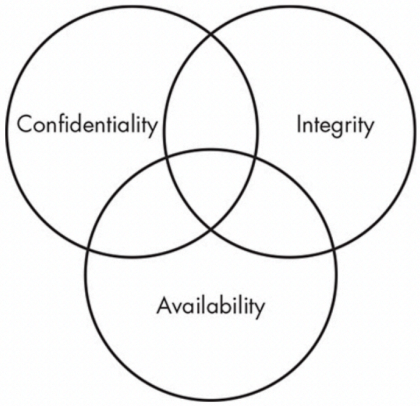

Figure 1-1: The CIA triad

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.

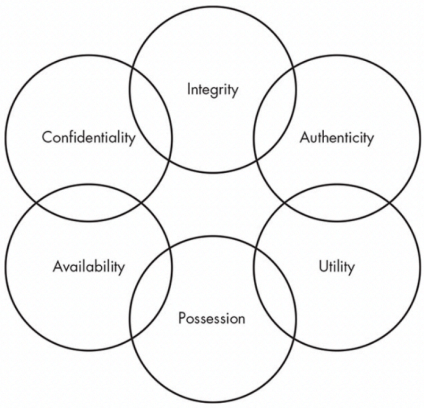

Figure 1-2: The Parkerian hexad

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.

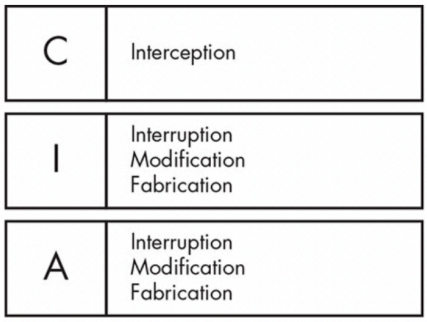

Figure 1-3: The CIA triad and categories of attacks

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