CHAPTER

Ethical Hacking and the Legal System

This book has not been compiled and written to be used as a tool by individuals who wish to carry out malicious and destructive activities. It is a tool for people who are interested in extending or perfecting their skills to defend against such attacks and damaging acts.

In this chapter, we cover the following topics:

- Why you need to understand your enemy's tactics
- The ethical hacking process
- The rise of cyberlaw
- Vulnerability disclosure

Why You Need to Understand Your Enemy's Tactics

Understanding how attacks work is one of the most challenging aspects of defensive security. By familiarizing yourself with how hackers think and operate, you can better tailor your organization's defenses to emerging threats and trends. If you don't test defenses against attacks, the only people who will be testing your network will be bad guys. By learning offensive security, you will be able to test your defenses and determine which aspects are operating correctly and where any gaps exist.

The criminal community is changing. Over the last few years, their motivation has evolved from the thrill of figuring out how to exploit vulnerabilities to figuring out how to make revenue from their actions and getting paid for their skills. Attackers who were out to "have fun" without any real target in mind have, to a great extent, been replaced by people who are serious about benefiting financially from their activities. Attacks are getting not only more specific, but also increasingly sophisticated. The following are just a few examples of this trend:

- In October 2013, hackers infiltrated Adobe and stole 38 million account credentials as well as encrypted credit card numbers. Portions of the data were exposed on the Internet.[1]
- In July 2013, Harbor Freight was hit with malware that aided in stealing card data from over 400 of its stores. This incident is one of many instances of malware being used to exfiltrate large amounts of credit card data from online retailers.[2]

- In May 2013, the Ponemon Institute released a report sponsored by Symantec that indicated breaches in the United States cost average companies approximately $188 per record.[3] This coupled with reports that breaches resulted in more than 28,000 records being exposed mean that although attackers are making money, it's costing companies more and more to deal with the compromises.
- At the peak of Christmas shopping in 2013, Target suffered one of the largest breaches to date. Between 40,000 and 70,000 individuals were potentially impacted by the losses. Target jumped ahead of the news reports in order to help people understand the breach as well as how the company was reacting to it. Target continues to maintain a site presence to provide information about new security measures put into place as well as how to deal with credit card fraud.[4]

A conservative estimate from Gartner pegs the average hourly cost of downtime for computer networks at $42,000.[5] A company that suffers from a worse than average downtime of 175 hours per year can lose more than $7 million per year. Even when attacks are not newsworthy enough to be reported on TV or talked about in security industry circles, they still negatively affect companies' bottom lines.

In addition to attackers who are trying to profit, some attackers are politically motivated. These attacks are labeled *hacktivism*. Both legal and illegal methods can be used to portray political ideology. Is it right to try to influence social change through the use of technology? Is web defacement covered under freedom of speech? Is it wrong to carry out a virtual "sit in" on a site that provides illegal content? During the 2009 Iran elections, was it unethical for an individual to set up a site that revealed discontent about the potential corrupt government elections? When Israeli invaded Gaza, many website defacements, DoS attacks, and website hijackings occurred. One's viewpoint determines what is ethical or not.

Some attackers also create and sell zero-day attacks. A *zero-day attack* is one for which there is currently no fix available. Whoever is running the particular software that contains that exploitable vulnerability is exposed, with little or no protection. The code for these types of attacks are advertised on special websites and sold to other attackers or organized crime rings.

Recognizing Trouble When It Happens

Network administrators, engineers, and security professionals must be able to recognize when an attack is underway or when one is imminent. It may seem like it should be easy to recognize an attack as it is happening—but only for the very "noisy" or overwhelming attacks such as denial-of-service (DoS) attacks. Many attackers fly under the radar and go unnoticed by security devices and security staff. By knowing *how* different types of attacks work, you can properly recognize and stop them.

You also need to know when an attack may be around the corner. If network staff is educated on attacker techniques and they see a ping sweep followed a day later by a port scan, they know their systems may soon be under attack. Many activities lead up to different types of attacks, so understanding these will help a company protect itself. The argument can be made that we now have more automated security products that identify

these types of activities so we don't have to see them coming. But, depending on the software, those activities may not be put in the necessary context and the software may make a dangerous decision. Computers can outperform any human on calculations and repetitive tasks, but we still have the ability to make necessary judgment calls because we understand the grays in life and do not just see things in 1s and 0s.

Hacking tools are really just software tools that carry out some specific types of procedure to achieve a desired result. The tools can be used for good (defensive) purposes or for bad (offensive) purposes. The good and the bad guys use the same exact toolset; the difference is their intent when operating these tools. It is imperative for security professionals to understand how to use these tools and how attacks are carried out if they are going to be of any use to their customers and to the industry.

The Ethical Hacking Process

To protect themselves, organizations may want to understand the impact and ability of an attacker. In this case, they may employ an *ethical hacker,* also known as a *penetration tester,* to simulate an attack against the environment. The techniques that penetration testers employ are designed to emulate those of real attackers without causing damage; they enable organizations to better protect themselves against attack. But customers and aspiring hackers need to understand how this process works.

By defining penetration testing activities, stages, and steps, you can set expectations between yourself as a tester and your customer. Customers may not be external to an organization; they may be internal as well. Regardless of who you are testing and why, establishing scope and a common language helps those impacted understand what you are doing and why and smooths the process by reducing misunderstandings.

Before describing the process of penetration testing, we need to discuss the difference between penetration testing and vulnerability assessment. These activities have different goals, but are often confused with one another. During a *vulnerability assessment,* some type of automated scanning product is used to probe the ports and services on a range of IP addresses. Most of these products can also test for the type of operating system and application software running and the versions, patch levels, user accounts, and services that are also running. These findings are matched up with correlating vulnerabilities in the product's database. The end result is a large pile of data that basically states, "Here is a list of your vulnerabilities and here is a list of things you need to do to fix them."

The problem with most vulnerability scans is, although they indicate the severity of a vulnerability, they rarely indicate its impact. This is where penetration testing comes in. Vulnerability scanning allows you to identity a piece of software as being vulnerable to exploit; a *penetration test* takes this further by exploiting vulnerabilities and, for example, accessing sensitive information. Most vulnerability scanners indicate what might be vulnerable based on versioning and some more invasive checks, but a penetration test indicates whether the vulnerability scanner finding is real or a false positive.

When penetration testers attack, their ultimate goal is usually to break into a system and hop from system to system until they "own" the domain or environment. Unlike a vulnerability assessment, a penetration test does not stop with the identification of a

possible vulnerability. Penetration testers leverage identified vulnerabilities until they own the domain or environment. Being "owned" means either having root privileges on the most critical Unix or Linux system or owning the domain administrator account that can access and control all of the resources on the network. Testers do this to show the customer (company) what an actual attacker can do under the circumstances and the network's current security posture.

Many times, while a penetration tester is carrying out her procedures to gain total control of the network, she will pick up significant trophies along the way. These trophies can include the CEO's passwords, company trade-secret documentation, administrative passwords to all border routers, documents marked "confidential" that are held on the CFO's and CIO's laptops, or the combination to the company vault. These trophies are collected along the way so the decision makers understand the ramifications of these vulnerabilities. A security professional can talk for hours to the CEO, CIO, or COO about services, open ports, misconfigurations, and potential vulnerabilities without making a point that this audience would understand or care about. But showing the CFO her next year's projections, showing the CIO all of the blueprints to next year's product line, or telling the CEO that his password is "IAmWearingPanties," will likely inspire them to learn more about firewalls and other countermeasures that should be put into place.

CAUTION No security professional should ever try to embarrass customers or make them feel inadequate for their lack of security. This is why the security professional has been invited into the environment. She is a guest and is there to help solve the problem, not point fingers. Also, in most cases, any sensitive data should not be read by the penetration testing team because of the possibilities of future lawsuits pertaining to the use of confidential information.

In this book, we cover advanced vulnerability detection, exploitation tools, and sophisticated penetration techniques. Then we'll dig into the programming code to show you how skilled attackers identify vulnerabilities and develop new tools to exploit their findings. Let's take a look at the ethical penetration testing process and see how it differs from that of unethical hacker activities.

The Penetration Testing Process

Once network administrators, engineers, and security professionals understand how attackers work, they can emulate their activities to carry out a useful penetration test. But why would anyone want to emulate an attack? Because this is the only way to truly test an environment's security level—you must know how it will react when a real attack is being carried out.

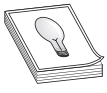
This book is laid out to walk you through these different steps so you can understand how many types of attacks take place. It can help you develop methodologies for emulating similar activities to test your company's security posture.

Just in case you choose to use the information in this book for unintended purposes (malicious activity), later in this chapter, we will also cover several federal laws that

have been put into place to scare you away from this activity. A wide range of computer crimes is taken seriously by today's court system, and attackers are receiving hefty fines and jail sentences for their activities. Don't let that be you. There is just as much fun and intellectual stimulation to be had working as a good guy—and no threat of jail time!

The penetration tester's motivation for testing is going to be driven by the client. Whether it's to access sensitive information, provide additional justification for ongoing projects, or to just test the security of the organization, it's important to understand what the client is looking for before testing starts. Once you understand what the goals are, directing the rest of the testing stages is much easier. Let's look at the typical steps in a penetration test.

1. **Ground rules** Establish the ground rules:
   - Set expectations and contact information between testers and customers.
   - Identify the parties involved and who is aware of the test.
   - Set start and stop dates and blackout periods.
   - Get formalized approval and a written agreement, including scope, signatures, and legal requirements, frequently called a *Statement of Work (SOW)*.



TIP Keep this document handy during testing. You may need it as a "get out of jail free" card

2. **Passive scanning** Gather as much information about the target as possible while maintaining zero contact between the penetration tester and the target. Passive scanning, otherwise known as *Open Source Intelligence (OSINT)*, can include
   - Social networking sites
   - Online databases
   - Google, Monster.com, etc.
   - Dumpster diving
3. **Active scanning and enumeration** Probe the target's public exposure with scanning tools, which might include
   - Commercial scanning tools
   - Network mapping
   - Banner grabbing
   - War dialing
   - DNS zone transfers
   - Sniffing traffic
   - Wireless war driving

4. **Fingerprinting** Perform a thorough probe of the target systems to identify
   - Operating system type and patch level
   - Applications and patch level
   - Open ports
   - Running services
   - User accounts
5. **Selecting target system** Identify the most useful target(s).
6. **Exploiting the uncovered vulnerabilities** Execute the appropriate attack tools targeted at the suspected exposures.
   - Some may not work.
   - Some may kill services or even kill the server.
   - Some may be successful.
7. **Escalating privilege** Escalate the security context so the ethical hacker has more control.
   - Gaining root or administrative rights
   - Using cracked password for unauthorized access
   - Carrying out buffer overflow to gain local versus remote control
8. **Documenting and reporting** Document everything found, how it was found, the tools that were used, vulnerabilities that were exploited, the timeline of activities and successes, and so on.

NOTE A more detailed approach to the attacks that are part of each methodology are included throughout the book.

What Would an Unethical Hacker Do Differently?

1. Target selection
   - Motivated by a grudge or for fun or profit.
   - There are no ground rules, no hands-off targets, and the security team is definitely blind to the upcoming attack.
2. Intermediaries
   - The attacker launches his attack from a different system (intermediary) than his own, or a series of other systems, to make it more difficult to track back to him in case the attack is detected.
   - Intermediaries are often victims of the attacker as well.

3. Penetration testing steps described in the previous section
   - Scanning
   - Footprinting
   - Selecting target system
   - Fingerprinting
   - Exploiting the uncovered vulnerabilities
   - Escalating privilege
4. Preserving access
   - This involves uploading and installing a rootkit, backdoor, Trojaned applications, and/or bots to assure that the attacker can regain access at a later time.
5. Covering tracks
   - Scrubbing event and audit logs
   - Hiding uploaded files
   - Hiding the active processes that allow the attacker to regain access
   - Disabling messages to security software and system logs to hide malicious processes and actions
6. Hardening the system
   - After taking ownership of a system, an attacker may fix the open vulnerabilities so no other attacker can use the system for other purposes.

How the attacker uses the compromised system depends on what his or her overall goals are, which could include stealing sensitive information, redirecting financial transactions, adding the systems to his or her bot network, extorting a company, and so on. The crux is that ethical and unethical hackers carry out basically the same activities only with different intentions. If the ethical hacker does not identify the hole in the defenses first, the unethical hacker will surely slip in and make himself at home.

The Rise of Cyberlaw

We currently live in a very interesting time. Information security and the legal system are becoming intertwined in a way that is straining the resources of both systems. The information security world uses terms like *bits, packets,* and *bandwidth,* and the legal community uses words like *jurisdiction, liability,* and *statutory interpretation.* In the past, these two quite different sectors had their own focus, goals, and procedures and did not collide with one another. But as computers have become the new tools for doing business and for committing traditional and new crimes, the two worlds have had to approach each other independently and then interact in a new space—a space now sometimes referred to as *cyberlaw.*

Today's CEOs and management not only need to worry about profit margins, market analysis, and mergers and acquisitions; now they also need to step into a world of practicing security with due care, understanding and complying with new government privacy and information security regulations, risking civil and criminal liability for security failures (including the possibility of being held personally liable for certain security breaches), and trying to comprehend and address the myriad of ways in which information security problems can affect their companies. Just as businesspeople must increasingly turn to security professionals for advice in seeking to protect their company's assets, operations, and infrastructure, so, too, must they turn to legal professionals for assistance in navigating the changing legal landscape in the privacy and information security area. Legislators, governmental and private information security organizations, and law enforcement professionals are constantly updating laws and related investigative techniques in an effort to counter each new and emerging form of attack that the bad guys come up with. Security technology developers and other professionals are constantly trying to outsmart sophisticated attackers, and vice versa. In this context, the laws being enacted provide an accumulated and constantly evolving set of rules that attempts to stay in step with new types of crimes and how they are carried out.

Cyberlaw is a broad term encompassing many elements of the legal structure that are associated with this rapidly evolving area. The increasing prominence of cyberlaw is not surprising if you consider that the first daily act of millions of American workers is to turn on their computers (frequently after they have already made ample use of their other Internet access devices and cell phones). These acts are innocuous to most people who have become accustomed to easy and robust connections to the Internet and other networks as a regular part of life. But this ease of access also results in business risk because network openness can also enable unauthorized access to networks, computers, and data, including access that violates various laws, some of which we briefly describe in this chapter.

Cyberlaw touches on many elements of business, including how a company contracts and interacts with its suppliers and customers, sets policies for employees handling data and accessing company systems, uses computers to comply with government regulations and programs, and so on. An important subset of these laws is the group of laws directed at preventing and punishing unauthorized access to computer networks and data. This section focuses on the most significant of these laws.

Because they are expected to work in the construct the laws provide, security professionals should be familiar with these laws. A misunderstanding of these ever-evolving laws, which is certainly possible given the complexity of computer crimes, can, in the extreme case, result in the innocent being prosecuted or the guilty remaining free. And usually it is the guilty ones who get to remain free.

Understanding Individual Cyberlaws

Many countries, particularly those whose economies have more fully integrated computing and telecommunications technologies, are struggling to develop laws and rules for dealing with computer crimes. We will cover selected US federal computer-crime laws in order to provide a sample of these many initiatives; a great deal of detail

regarding these laws is omitted and numerous laws are not covered. This section is intended neither to provide a thorough treatment of each of these laws, nor to cover any more than the tip of the iceberg of the many US technology laws. Instead, it is meant to raise awareness of the importance of considering these laws in your work and activities as an information security professional. That in no way means that the rest of the world is allowing attackers to run free and wild. With just a finite number of pages, we cannot properly cover all legal systems in the world or all of the relevant laws in the United States. It is important that you spend the time necessary to fully understand the laws that are relevant to your specific location and activities in the information security area.

The following sections survey some of the many US federal computer crime statutes, including

- 18 USC 1029: Fraud and Related Activity in Connection with Access Devices
- 18 USC 1030: Fraud and Related Activity in Connection with Computers
- 18 USC 2510 et seq.: Wire and Electronic Communications Interception and Interception of Oral Communications
- 18 USC 2701 et seq.: Stored Wire and Electronic Communications and Transactional Records Access
- The Digital Millennium Copyright Act
- The Cyber Security Enhancement Act of 2002

18 USC Section 1029: The Access Device Statute

The purpose of the Access Device Statute is to curb unauthorized access to accounts; theft of money, products, and services; and similar crimes. It does so by criminalizing the possession, use, or trafficking of counterfeit or unauthorized access devices or device-making equipment, and other similar activities (described shortly), to prepare for, facilitate, or engage in unauthorized access to money, goods, and services. It defines and establishes penalties for fraud and illegal activity that can take place through the use of such counterfeit access devices.

The *elements* of a crime are generally the things that need to be shown in order for someone to be prosecuted for that crime. These elements include consideration of the potentially illegal activity in light of the precise definitions of *access device, counterfeit access device, unauthorized access device, scanning receiver,* and other definitions that together help to define the scope of the statute's application.

The term *access device* refers to a type of application or piece of hardware that is created specifically to generate access credentials (passwords, credit card numbers, long-distance telephone service access codes, PINs, and so on) for the purpose of unauthorized access. Specifically, it is defined broadly to mean

> any card, plate, code, account number, electronic serial number, mobile identification number, personal identification number, or other telecommunications service, equipment, or instrument identifier, or other means of account access that can be used, alone or in conjunction with another access

> device, to obtain money, goods, services, or any other thing of value, or that can be used to initiate a transfer of funds (other than a transfer originated solely by paper instrument).[6]

One example of a violation would be using a tool to steal credentials and then using those credentials to break into the Pepsi-Cola Network. If you were to steal the soda recipe, you would be guilty of "Using or obtaining an access device to gain unauthorized access and obtain anything of value totaling $1,000 or more during a one-year period." This would result in a fine of upward of $10,000 or twice the value of the damages and up to 10 years in prison. If you were caught twice, you could get up to 20 years in prison.

Section 1029 addresses offenses that involve generating or illegally obtaining access credentials, which can involve just obtaining the credentials or obtaining and *using* them. These activities are considered criminal *whether or not* a computer is involved—unlike the statute discussed next, which pertains to crimes dealing specifically with computers.

18 USC Section 1030 of the Computer Fraud and Abuse Act

The Computer Fraud and Abuse Act (CFAA) (as amended by the USA Patriot Act) is an important federal law that addresses acts that compromise computer network security.[7] It prohibits unauthorized access to computers and network systems, extortion through threats of such attacks, the transmission of code or programs that cause damage to computers, and other related actions. It addresses unauthorized access to government, financial institutions, and other computer and network systems, and provides for civil and criminal penalties for violators. The act outlines the jurisdiction of the FBI and Secret Service.

The term *protected computer,* as commonly put forth in the CFAA, means a computer used by the US government, financial institutions, or any system used in interstate or foreign commerce or communications. The CFAA is the most widely referenced statute in the prosecution of many types of computer crimes. A casual reading of the CFAA suggests that it only addresses computers used by government agencies and financial institutions, but there is a small (but important) clause that extends its reach. This clause says that the law applies also to any system "used in interstate or foreign commerce or communication." The meaning of "used in interstate or foreign commerce or communication" is very broad, and, as a result, CFAA operates to protect nearly all computers and networks. Almost every computer connected to a network or the Internet is used for some type of commerce or communication, so this small clause pulls nearly all computers and their uses under the protective umbrella of the CFAA. Amendments by the USA Patriot Act to the term "protected computer" under CFAA extended the definition to any computers located outside the United States, as long as they affect interstate or foreign commerce or communication of the United States. So if the United States can get the attackers, they will attempt to prosecute them no matter where in the world they live.

The CFAA has been used to prosecute many people for various crimes. Two types of unauthorized access can be prosecuted under the CFAA: these include wholly unauthorized access by outsiders, and also situations where individuals, such as employees,

contractors, and others with permission, exceed their authorized access and commit crimes. The CFAA states that if someone accesses a computer in an unauthorized manner *or* exceeds his or her access rights, that individual can be found guilty of a federal crime. This clause allows companies to prosecute employees who carry out fraudulent activities by abusing (and exceeding) the access rights their company has given them.

In November 2013, US-CERT released an advisory about CryptoLocker Ransomware that will encrypt the contents of a computer and then charge the victim for the keys to unlock it.[8] One area in which 18 USC Section 1030 would come into play would be if the CryptoLocker software was used to encrypt a government system. The CryptoLocker demands payment, which is considered extortion. Under the CFAA, if the attackers are caught this could yield up to a $250,000 fine as well as up to 10 years in prison for the first offense.

Under the CFAA, the FBI and the Secret Service have the responsibility for handling these types of crimes, and they have their own jurisdictions. The FBI is responsible for cases dealing with national security, financial institutions, and organized crime. The Secret Service's jurisdiction encompasses any crimes pertaining to the Treasury Department and any other computer crime that does not fall within the FBI's jurisdiction.

NOTE The Secret Service's jurisdiction and responsibilities have grown since the Department of Homeland Security (DHS) was established. The Secret Service now deals with several areas to protect the nation and has established an Information Analysis and Infrastructure Protection division to coordinate activities in this area. This division's responsibilities encompass the preventive procedures for protecting "critical infrastructure," which includes such things as power grids, water supplies, and nuclear plants in addition to computer systems.

State Law Alternatives The amount of damage resulting from a violation of the CFAA can be relevant for either a criminal or civil action. As noted earlier, the CFAA provides for both criminal and civil liability for a violation. A criminal violation is brought by a government official and is punishable by either a fine or imprisonment or both. By contrast, a civil action can be brought by a governmental entity or a private citizen and usually seeks the recovery of payment of damages incurred and an *injunction*, which is a court order to prevent further actions prohibited under the statute. The amount of damages is relevant for some but not all of the activities that are prohibited by the statute. The victim must prove that *damages* have indeed occurred. In this case, damage is defined as disruption of the availability or integrity of data, a program, a system, or information. For most CFAA violations, the losses must equal at least $5,000 during any one-year period.

This all sounds great and might allow you to sleep better at night, but not all of the harm caused by a CFAA violation is easily quantifiable, or if quantifiable, may not exceed the $5,000 threshold. For example, when computers are used in distributed denial-of-service attacks or when processing power is being used to brute-force and uncover an encryption key, the issue of damages becomes cloudy. These losses do not

always fit into a nice, neat formula to evaluate whether they total $5,000. The victim of an attack can suffer various qualitative harms that are much harder to quantify. If you find yourself in this type of situation, the CFAA might not provide adequate relief. In that context, this *federal* statute might not be a useful tool for you and your legal team.

Often victims will turn to state laws that may offer more flexibility when prosecuting an attacker. State laws that are relevant in the computer crime arena include both new state laws being passed by state legislatures in an attempt to protect their residents and traditional state laws dealing with trespassing, theft, larceny, money laundering, and other crimes.

Resorting to state laws is not, however, always straightforward. First, there are 50 different states and nearly that many different "flavors" of state law. Thus, for example, trespass law varies from one state to the next, resulting in a single activity being treated in two very different ways under state law. Some states require a demonstration of damages as part of the claim of trespass (not unlike the CFAA requirement), whereas other states do not require a demonstration of damages in order to establish that an actionable trespass has occurred.

Importantly, a company will usually want to bring a case to the courts of a state that has the most favorable definition of a crime so it can most easily make its case. Companies will not, however, have total discretion as to where they bring the case to court. There must generally be some connection, or *nexus,* to a state in order for the courts of that state to have jurisdiction to hear a case.

TIP If you are considering prosecuting a computer crime that affected your company, start documenting the time people have to spend on the issue and other costs incurred in dealing with the attack. This lost paid employee time and other costs may be relevant in the measure of damages or, in the case of the CFAA or those states that require a showing of damages as part of a trespass case, to the success of the case.

As with all of the laws summarized in this chapter, information security professionals must be careful to confirm with each relevant party the specific scope and authorization for work to be performed. If these confirmations are not in place, it could lead to misunderstandings and, in the extreme case, prosecution under the Computer Fraud and Abuse Act or other applicable law. In the case of *Sawyer vs. Department of Air Force,* the court rejected an employee's claim that alterations to computer contracts were made to demonstrate the lack of security safeguards and found the employee liable because the statute only required proof of use of a computer system for any unauthorized purpose.

18 USC Sections 2510, et. Seq., and 2701, et. Seq., of the Electronic Communications Privacy Act

These sections are part of the Electronic Communications Privacy Act (ECPA), which is intended to protect communications from unauthorized access. The ECPA, therefore, has a different focus than the CFAA, which is directed at protecting computers and

network systems. Most people do not realize that the ECPA is made up of two main parts: one that amended the Wiretap Act and the other than amended the Stored Communications Act, each of which has its own definitions, provisions, and cases interpreting the law.

The Wiretap Act has been around since 1918, but the ECPA extended its reach to electronic communication when society moved in that direction. The Wiretap Act protects communications, including wire, oral, and data during transmission, from unauthorized access and disclosure (subject to exceptions). The Stored Communications Act protects some of the same types of communications before and/or after the communications are transmitted and stored electronically somewhere. Again, this sounds simple and sensible, but the split reflects a recognition that there are different risks and remedies associated with active versus stored communications.

The Wiretap Act generally provides that there cannot be any intentional interception of wire, oral, or electronic communication in an illegal manner. Among the continuing controversies under the Wiretap Act is the meaning of the word *interception*. Does it apply only when the data is being transmitted as electricity or light over some type of transmission medium? Does the interception have to occur at the time of the transmission? Does it apply to this transmission *and* to where it is temporarily stored on different hops between the sender and destination? Does it include access to the information received from an active interception, even if the person did not participate in the initial interception? The question of whether an interception has occurred is central to the issue of whether the Wiretap Act applies.

Although the ECPA seeks to limit unauthorized access to communications, it recognizes that some types of *unauthorized* access are necessary. For example, if the government wants to listen in on phone calls, Internet communication, email, network traffic, or you whispering into a tin can, it can do so if it complies with safeguards established under the ECPA that are intended to protect the privacy of persons who use those systems.

Digital Millennium Copyright Act (DMCA)

The DMCA is not often considered in a discussion of hacking and the question of information security, but it is relevant. The DMCA was passed in 1998 to implement the World Intellectual Property Organization Copyright Treaty (WIPO Copyright Treaty).[9] The WIPO Treaty requires treaty parties to "provide adequate legal protection and effective legal remedies against the circumvention of effective technological measures that are used by authors," and to restrict acts in respect to their works that are not authorized. Thus, while the CFAA protects computer systems and the ECPA protects communications, the DMCA protects certain (copyrighted) content itself from being accessed without authorization. The DMCA establishes both civil and criminal liability for the use, manufacture, and trafficking of devices that circumvent technological measures controlling access to, or protection of, the rights associated with copyrighted works.

The DMCA's anti-circumvention provisions make it criminal to willfully, and for commercial advantage or private financial gain, circumvent technological measures that control access to protected copyrighted works. In hearings, the crime that the

anti-circumvention provision is designed to prevent has been described as "the electronic equivalent of breaking into a locked room in order to obtain a copy of a book."

Circumvention is to "descramble a scrambled work…decrypt an encrypted work, or otherwise…avoid, bypass, remove, deactivate, or impair a technological measure, without the authority of the copyright owner." The legislative history provides that "if unauthorized access to a copyrighted work is effectively prevented through use of a password, it would be a violation of this section to defeat or bypass the password." A "technological measure" that "effectively controls access" to a copyrighted work includes measures that "in the ordinary course of its operation, requires the application of information, or a process or a treatment, with the authority of the copyright owner, to gain access to the work." Therefore, measures that can be deemed to "effectively control access to a work" would be those based on encryption, scrambling, authentication, or some other measure that requires the use of a key provided by a copyright owner to gain access to a work.

Said more directly, the Digital Millennium Copyright Act (DMCA) states that no one should attempt to tamper with and break an access control mechanism that is put into place to protect an item that is protected under the copyright law. If you have created a nifty little program that controls access to all of your written interpretations of the grandness of the invention of pickled green olives, and someone tries to break this program to gain access to your copyright-protected insights and wisdom, the DMCA could come to your rescue.

The fear of many in the information security industry is that this provision could be interpreted and used to prosecute individuals carrying out commonly applied security practices. For example, a penetration test is a service performed by information security professionals in which an individual or team attempts to break or slip by access control mechanisms. Security classes are offered to teach people how these attacks take place so they can understand what countermeasures are appropriate and why. But how will people learn how to hack, crack, and uncover vulnerabilities and flaws if the DMCA indicates that classes, seminars, and the like cannot be conducted to teach the security professionals these skills?

The DMCA provides an explicit exemption allowing "encryption research" for identifying the flaws and vulnerabilities of encryption technologies. It also provides for an exception for engaging in an act of security testing (if the act does not infringe on copyrighted works or violate applicable law such as the CFAA), but it does not contain a broader exemption covering a variety of other activities that information security professionals might engage in. Yes, as you pull one string, three more show up. Again, you see why it's important for information security professionals to have a fair degree of familiarity with these laws to avoid missteps.

Cyber Security Enhancement Act of 2002

Several years ago, Congress determined that the legal system still allowed for too much leeway for certain types of computer crimes and that some activities not labeled "illegal" needed to be. In July 2002, the House of Representatives voted to put stricter laws in place, and to dub this new collection of laws the Cyber Security Enhancement Act (CSEA) of 2002.[10] The CSEA made a number of changes to federal law involving computer crimes.

The act stipulates that attackers who carry out certain computer crimes may now get a life sentence in jail. If an attacker carries out a crime that could result in another's bodily harm or possible death, or a threat to public health or safety, the attacker could face life in prison. This does not necessarily mean that someone has to throw a server at another person's head, but since almost everything today is run by some type of technology, personal harm or death could result from what would otherwise be a run-of-the-mill hacking attack. For example, if an attacker were to compromise embedded computer chips that monitor hospital patients, cause fire trucks to report to wrong addresses, make all of the traffic lights change to green, or reconfigure airline controller software, the consequences could be catastrophic and under the CSEA result in the attacker spending the rest of her days in jail.

NOTE In 2013, a newer version of the Cyber Security Enhancement Act passed the House and is still on the docket for the Senate to take action, at the time of this writing. Its purpose includes funding for cybersecurity development, research, and technical standards.

The CSEA was also developed to supplement the Patriot Act, which increased the US government's capabilities and power to monitor communications. One way in which this is done is that the CSEA allows service providers to report suspicious behavior without risking customer litigation. Before this act was put into place, service providers were in a sticky situation when it came to reporting possible criminal behavior or when trying to work with law enforcement. If a law enforcement agent requested information on a provider's customer and the provider gave it to them without the customer's knowledge or permission, the service provider could, in certain circumstances, be sued by the customer for unauthorized release of private information. Now service providers can report suspicious activities and work with law enforcement without having to tell the customer. This and other provisions of the Patriot Act have certainly gotten many civil rights monitors up in arms.

It is up to you which side of the fight you choose to play on—but remember that computer crimes are not treated as lightly as they were in the past. Trying out a new tool or pressing Start on an old tool may get you into a place you never intended—jail. So as your mother told you—be good, and may the Force be with you.

The Controversy of "Hacking" Tools

In most instances, the toolset used by malicious attackers is the same toolset used by security professionals. Many people do not understand this. In fact, the books, classes, articles, websites, and seminars on hacking could be legitimately renamed to "security professional toolset education." The problem arises when marketing people like to use the word *hacking* because it draws more attention and paying customers.

As covered earlier, ethical hackers go through the same processes and procedures as unethical hackers, so it only makes sense that they use the same basic toolset. It would

not be useful to prove that attackers could not get through the security barriers with Tool A if attackers do not use Tool A. The ethical hacker has to know what the bad guys are using, know the new exploits that are out in the underground, and continually keep her skills and knowledgebase up to date. Why? Because, odds are against the company and the security professional. The security professional has to identify and address all of the vulnerabilities in an environment. The attacker only has to be really good at one or two exploits, or really lucky. A comparison can be made to the US Homeland Security responsibilities. The CIA and FBI are responsible for protecting the nation from the 10 million things terrorists could possibly think up and carry out. The terrorist only has to be successful at *one* of these 10 million things.

Vulnerability Disclosure

For years customers have demanded that operating systems and applications provide more and more functionality. Vendors continually scramble to meet this demand while also attempting to increase profits and market share. This combination of racing to market and maintaining a competitive advantage has resulted in software containing many flaws—flaws that range from mere nuisances to critical and dangerous vulnerabilities that directly affect a customer's protection level.

The hacking community's skill sets are continually increasing. It used to take the hacking community months to carry out a successful attack from an identified vulnerability; today it happens in days or even hours. The increase in interest and talent in the criminal community equates to quicker and more damaging attacks and malware for the industry to combat. It is imperative that vendors not sit on the discovery of true vulnerabilities, but instead work to release fixes to customers who need them as soon as possible.

For this to happen, ethical hackers must understand and follow the proper methods for disclosing identified vulnerabilities to the software vendor. If an individual uncovers a vulnerability and illegally exploits it and/or tells others how to carry out this activity, he is considered a *black hat*. If an individual uncovers a vulnerability and exploits it with authorization, she is considered a *white hat*. If a different person uncovers a vulnerability, does not illegally exploit it or tell others how to do so, and works with the vendor to fix it, this person is considered a *gray hat*.

We promote using the knowledge that we are sharing with you in a responsible manner that will only help the industry—not hurt it. To do this, you should understand the policies, procedures, and guidelines that have been developed to allow hackers and vendors to work together.

Different Teams and Points of View

Unfortunately, almost all of today's software products are riddled with flaws. These flaws can present serious security concerns for consumers. For customers who rely extensively on applications to perform core business functions, bugs can be crippling and, therefore, must be dealt with properly. How best to address the problem is a complicated issue because it involves two key players who usually have very different views on how to achieve a resolution.

The first player is the consumer. An individual or company buys a product, relies on it, and expects it to work. Often, the consumer owns a community of interconnected systems (a network) that all rely on the successful operation of software to do business. When the consumer finds a flaw, he reports it to the vendor and expects a solution in a reasonable timeframe.

The second player is the software vendor. The vendor develops the product and is responsible for its successful operation. The vendor is looked to by thousands of customers for technical expertise and leadership in the upkeep of its product. When a flaw is reported to the vendor, it is usually one of many that the vendor must deal with, and some fall through the cracks for one reason or another.

The issue of public disclosure has created quite a stir in the computing industry because each group views the issue so differently. Many believe knowledge is the public's right, and all security vulnerability information should be disclosed as a matter of principle. Furthermore, many consumers feel that the only way to get truly quick results from a large software vendor is to pressure it to fix the problem by threatening to make the information public. Vendors have had the reputation of simply plodding along and delaying the fixes until a later version or patch is scheduled for release, which will address the flaw. This approach doesn't always consider the best interests of consumers, however, as they must sit and wait for the vendor to fix a vulnerability that puts their business at risk.

The vendor looks at the issue from a different perspective. Disclosing sensitive information about a software flaw causes two major problems. First, the details of the flaw will help attackers exploit the vulnerability. The vendor's argument is that if the issue is kept confidential while a solution is being developed, attackers will not know how to exploit the flaw. Second, the release of this information can hurt the company's reputation, even in circumstances when the reported flaw is later proven to be false. It is much like a smear campaign in a political race that appears as the headline story in a newspaper. Reputations are tarnished, and even if the story turns out to be untrue, a retraction is usually printed on the back page a week later. Vendors fear the same consequence for massive releases of vulnerability reports.

Because of these two distinct viewpoints, several organizations have rallied together to create policies, guidelines, and general suggestions on how to handle software vulnerability disclosures. This section will attempt to cover the issue from all sides and help educate you on the fundamentals behind the ethical disclosure of software vulnerabilities.

How Did We Get Here?

Before the mailing list Bugtraq was created, individuals who uncovered vulnerabilities and ways to exploit them just communicated directly with each other. The creation of Bugtraq provided an open forum for these individuals to discuss the same issues and work collectively. Easy access to ways of exploiting vulnerabilities gave way to the numerous script-kiddie point-and-click tools available today, which allow people who do not even understand a vulnerability to exploit it successfully. Bugtraq led to an increase in attacks on the Internet, on networks, and against vendors. Many vendors were up in arms, demanding a more responsible approach to vulnerability disclosure.

In 2002, Internet Security Systems (ISS) discovered several critical vulnerabilities in products like Apache web server, Solaris X Windows font service, and Internet Software Consortium BIND software. ISS worked with the vendors directly to come up with solutions. A patch that was developed and released by Sun Microsystems was flawed and had to be recalled. An Apache patch was not released to the public until after the vulnerability was posted through public disclosure, even though the vendor knew about the vulnerability. Although these are older examples, these types of activities—and many more like them—left individuals and companies vulnerable; they were victims of attacks and eventually developed a deep feeling of distrust of software vendors. Critics also charged that security companies, like ISS, have alternative motives for releasing this type of information. They suggest that by releasing system flaws and vulnerabilities, they generate "good press" for themselves and thus promote new business and increased revenue.

Because of the failures and resulting controversy that ISS encountered, it decided to initiate its own disclosure policy to handle such incidents in the future. It created detailed procedures to follow when discovering a vulnerability and how and when that information would be released to the public. Although their policy is considered "responsible disclosure," in general, it does include one important caveat—vulnerability details would be released to its customers and the public at a "prescribed period of time" after the vendor has been notified. ISS coordinates their public disclosure of the flaw with the vendor's disclosure. This policy only fueled the people who feel that vulnerability information should be available for the public to protect themselves.

This dilemma, and many others, represent the continual disconnect among vendors, security companies, and gray hat hackers today. Differing views and individual motivations drive each group down various paths. The models of proper disclosure that are discussed in upcoming sections have helped these entities to come together and work in a more concerted effort, but much bitterness and controversy around this issue remains.

NOTE The range of emotion, the numerous debates, and controversy over the topic of full disclosure has been immense. Customers and security professionals alike are frustrated with software flaws that still exist in the products in the first place and the lack of effort from vendors to help in this critical area. Vendors are frustrated because exploitable code is continually released just as they are trying to develop fixes. We will not be taking one side or the other of this debate, but will do our best to tell you how you can help, and not hurt, the process.

CERT's Current Process

The first place to turn to when discussing the proper disclosure of software vulnerabilities is the governing body known as the *CERT Coordination Center (CC)*. CERT/CC is a federally funded research and development operation that focuses on Internet security and related issues. Established in 1988 in reaction to the first major virus outbreak on the Internet, the CERT/CC has evolved over the years, taking on more substantial roles in the industry, which include establishing and maintaining industry standards for the

way technology vulnerabilities are disclosed and communicated. In 2000, the organization issued a policy that outlined the controversial practice of releasing software vulnerability information to the public. The policy covered the following areas:

- Full disclosure will be announced to the public within 45 days of being reported to CERT/CC. This timeframe will be executed even if the software vendor does not have an available patch or appropriate remedy. The only exception to this rigid deadline will be exceptionally serious threats or scenarios that would require a standard to be altered.
- CERT/CC will notify the software vendor of the vulnerability immediately so a solution can be created as soon as possible.
- Along with the description of the problem, CERT/CC will forward the name of the person reporting the vulnerability unless the reporter specifically requests to remain anonymous.
- During the 45-day window, CERT/CC will update the reporter on the current status of the vulnerability without revealing confidential information.

CERT/CC states that its vulnerability policy was created with the express purpose of informing the public of potentially threatening situations while offering the software vendor an appropriate timeframe to fix the problem. The independent body further states that all decisions on the release of information to the public are based on what is best for the overall community.

The decision to go with 45 days was met with controversy as consumers widely felt that was too much time to keep important vulnerability information concealed. The vendors, on the other hand, felt the pressure to create solutions in a short timeframe while also shouldering the obvious hits their reputations would take as news spread about flaws in their product. CERT/CC came to the conclusion that 45 days was sufficient enough time for vendors to get organized, while still taking into account the welfare of consumers.

To accommodate vendors and their perspective of the problem, CERT/CC performs the following:

- CERT/CC will make good faith efforts always to inform the vendor before releasing information so there are no surprises.
- CERT/CC will solicit vendor feedback in serious situations and offer that information in the public release statement. In instances when the vendor disagrees with the vulnerability assessment, the vendor's opinion will be released as well, so both sides can have a voice.
- Information will be distributed to all related parties that have a stake in the situation prior to the disclosure. Examples of parties that could be privy to confidential information include participating vendors, experts who could provide useful insight, Internet Security Alliance members, and groups that may be in the critical path of the vulnerability.

Although there have been other guidelines developed and implemented after CERT's model, CERT is usually the "middle man" between the bug finder and the vendor to try and help the process and enforce the necessary requirements of all of the parties involved.

Organization for Internet Safety

There are three basic types of vulnerability disclosures: full disclosure, partial disclosure, and nondisclosure. Each type has its advocates, and long lists of pros and cons can be debated regarding each type. The *Organization for Internet Safety (OIS)* was created to help meet the needs of all groups and is the policy that best fits into a partial disclosure classification.[11] This section gives an overview of the OIS approach, as well as provides the step-by-step methodology that has been developed to provide a more equitable framework for both the user and the vendor.

A group of researchers and vendors formed the OIS with the goal of improving the way software vulnerabilities are handled. The OIS members included @stake, Bind-View Corp., The SCO Group, Foundstone, Guardent, Internet Security Systems, McAfee, Microsoft Corporation, Network Associates, Oracle Corporation, SGI, and Symantec. The OIS shut down after serving its purpose, which was to create the vulnerability disclosure guidelines.

The OIS believed that vendors and consumers should work together to identify issues and devise reasonable resolutions for both parties. It tried to bring together a broad, valued panel that offered respected, unbiased opinions to make recommendations. The model was formed to accomplish two goals:

- Reduce the risk of software vulnerabilities by providing an improved method of identification, investigation, and resolution.
- Improve the overall engineering quality of software by tightening the security placed on the end product.

Responsible Disclosure Phases

Understanding the steps of responsible disclosure under the OIS model are critical. This process is summarized here; however, a detailed methodology with examples and process maps are available as part of the standard:

1. **Discovery** A flaw has been found. The researcher must discover if a vulnerability has already been reported or patched, ensure it can be reproduced consistently, and ensure it impacts the default configuration. If so, the discoverer creates a *vulnerability summary report (VSR)*.
2. **Notification** The discoverer submits his contact information as well as the VSR to the vendor referencing the vendor's security policy. These details are sent to the address listed in its security policy or to one of the standard email addresses laid out in the OIS standard. The vendor must respond to this step.
3. **Validation** The vendor researches and validates the vulnerability. Regular status updates to the reporter are suggested during this phase.

4. **Findings** Once the vendor finishes its investigation, it confirms, disproves, or indicates inconclusive findings. The vendor is required to demonstrate research was done and typically meets this requirement by providing lists of products, versions, and tests performed.
5. **Resolution** If a flaw is inconclusive or is disproven, the weakness may be made public. If it is confirmed, the vendor typically has 30 days to issue a patch or fix.
6. **Release** The remedy is released as well as the notification.

Conflicts Will Still Exist

Those who discover vulnerabilities *usually* are motivated to protect the industry by identifying and helping remove dangerous software from commercial products. A little fame, admiration, and bragging rights are also nice for those who enjoy having their egos stroked. Vendors, on the other hand, are motivated to improve their product, avoid lawsuits, stay clear of bad press, and maintain a responsible public image.

There's no question that software flaws are rampant. The Common Vulnerabilities and Exposures (CVE) list is a compilation of publicly known vulnerabilities. This list is over ten years old and catalogs more than 40,000 bugs. This list is frequently updated, and through a joint collaboration with MITRE and National Institute of Standards and Technology (NIST), the National Vulnerability Database(NVD) provides a searchable database for these CVE advisories at http://nvd.nist.gov/.

Vulnerability reporting considerations include financial, legal, and moral ones for both researchers and vendors alike. Vulnerabilities can mean bad public relations for a vendor that, to improve its image, must release a patch once a flaw is made public. But, at the same time, vendors may decide to put the money into fixing software after it's released to the public, rather than making it perfect (or closer to perfect) beforehand. In that way, they use vulnerability reporting as after-market security consulting.

Public disclosure helps improve security, according to information security expert Bruce Schneier.[12] He says that the only reason vendors patch vulnerabilities is because of full disclosure, and that there's no point in keeping a bug a secret—hackers will discover it anyway. Before full disclosure, he says, it was too easy for software companies to ignore the flaws and threaten the researcher with legal action. Ignoring the flaws was easier for vendors especially because an unreported flaw affected the software's users much more than it affected the vendor.

Security expert Marcus Ranum takes a dim view of public disclosure of vulnerabilities.[13] He says that an entire economy of researchers is trying to cash in on the vulnerabilities that they find and selling them to the highest bidder, whether for good or bad purposes. His take is that researchers are constantly seeking fame and that vulnerability disclosure is "rewarding bad behavior," rather than making software better.

But the vulnerability researchers who find and report bugs have a different take, especially when they aren't getting paid. Another issue that has arisen is that researchers are tired of working for free without legal protection.

"No More Free Bugs"

In 2009, several gray hat hackers—Charlie Miller, Alex Sotirov, and Dino Dai Zovi—publicly announced a new stance: "No More Free Bugs."[14] They argue that the value of software vulnerabilities often doesn't get passed on to independent researchers who find legitimate, serious flaws in commercial software. Along with iDefense and ZDI, the software vendors themselves have their own employees and consultants who are supposed to find and fix bugs. ("No More Free Bugs" is targeted primarily at the for-profit software vendors that hire their own security engineer employees or consultants.)

The researchers involved in "No More Free Bugs" also argue that independent researchers are putting themselves at risk when they report vulnerabilities to vendors. They have no legal protection when they disclose a found vulnerability—so they're not only working for free, but also opening themselves up to threats of legal action, too. And independent researchers don't often have access to the right people at the software vendor, those who can create and release the necessary patches. For many vendors, vulnerabilities mainly represent threats to their reputation and bottom line, and they may stonewall researchers' overtures, or worse. Although vendors create responsible disclosure guidelines for researchers to follow, they don't maintain guidelines for how they treat the researchers.

Furthermore, these researchers say that software vendors often depend on them to find bugs rather than investing enough in finding vulnerabilities themselves. Uncovering flaws in today's complex software takes time and skill, and the founders of the "No More Free Bugs" movement feel as though either the vendors should employ people to uncover these bugs and identify fixes or they should pay gray hats who uncover them and report them responsibly.

This group of researchers also calls for more legal options when carrying out and reporting on software flaws. In some cases, researchers have uncovered software flaws and vendors have then threatened these individuals with lawsuits to keep them quiet and help ensure the industry did not find out about the flaws.

NOTE For a sample list of security research that resulted in legal action as well as the outcome, visit http://attrition.org/errata/legal_threats/.

Bug Bounty Programs

In recent years, vendors have adopted some of the previous principles as part of Bug Bounty programs. Microsoft, for example, says it won't sue researchers "that responsibly submit potential online services security vulnerabilities." And Mozilla runs a "bug bounty program" that offers researchers a flat $500 fee (plus a T-shirt!) for reporting valid, critical vulnerabilities.[15] In 2009, Google offered a cash bounty for the best vulnerability found in Native Client. Organizations have even developed a business plan on managing these bug bounty programs. One example is BugCrowd, a site that puts testers together with clients who want software tested and are willing to pay for it.[16]

Although more and more software vendors are reacting appropriately when vulnerabilities are reported (because of market demand for secure products), many people believe that vendors will not spend the extra money, time, and resources to carry out this process properly until they are held legally liable for software security issues. The possible legal liability issues software vendors may or may not face in the future is a can of worms we will not get into, but these issues are gaining momentum in the industry.

The Zero-Day Initiative (ZDI) is another organization that pays for vulnerability disclosure. It offers a web portal for researchers to report and track vulnerabilities. ZDI performs identity checks on researchers who report vulnerabilities, including checking that the researcher isn't on any government "do not do business with" lists. ZDI then validates the bug in a security lab before offering the researcher payment and contacting the vendor. ZDI also maintains its intrusion prevention system (IPS) program to write filters for whatever customer areas are affected by the vulnerability. The filter descriptions are designed to protect customers, but remain vague enough to keep details of unpatched flaws secret. ZDI works with the vendor on notifying the public when the patch is ready, giving the researcher credit if he or she requests it.

Summary

Before you can embark on an exploration of ethical hacking, you need to understand where ethical hacking and criminal activity are similar and deviate. With this knowledge, you can better understand what steps you need to take to model this malicious activity in order to help assess the security of environments with realistic benchmarks. While doing this, it's also important to understand the legal aspects of the business process as well as any applicable local, state, and federal laws.

Through this chapter, we covered why understanding how malicious individuals work is important, and how the steps of the ethical hacking process map to the methodology of an attacker. We also covered a number of laws that impact ethical hackers in the United States, including DCMA and CFAA. We also detailed reasons to check on local laws before preforming penetration testing to ensure that there aren't laws that are more strict than federal ones.

Finally, we covered why ethical disclosure is important and how to deal properly with the disclosure process. Armed with this information, you should understand the steps of getting work as an ethical hacker, ensuring that you stay safe while testing, and as you discover new flaws, how to contribute back to the community effectively.

References

1. Adobe Breach Impacted at Least 38 Million Users (2013, October 19). Retrieved from Krebs on Security: krebsonsecurity.com/2013/10/adobe-breach-impacted-at-least-38-million-users/.

2. Kitten, Tracy (2013, August 7). "New Retail Breach Amount 2013's Biggest?" Retrieved from *BankInfo Security*: www.bankinfosecurity.com/impact-harbor-freight-attack-grows-a-5970/op-1.
3. 2013 Cost of Data Breach Study: Global Analysis (2013, May). Retrieved from Symantec: www4.symantec.com/mktginfo/whitepaper/053013_GL_NA_WP_Ponemon-2013-Cost-of-a-Data-Breach-Report_daiNA_cta72382.pdf.
4. Data Breach FAQ. Target. Retrieved from Target: corporate.target.com/about/shopping-experience/payment-card-issue-FAQ.
5. Pisello, Tom, and Bill Quirk (2004, January 5). "How to Quantify Downtime." Retrieved from *Network World:* www.networkworld.com/article/2329877/infrastructure-management/how-to-quantify-downtime.html.
6. 18 U.S. Code § 1029. Fraud and Related Activity in Connection with Access Devices. Retrieved from the Legal Information Institute: www.law.cornell.edu/uscode/text/18/1029.
7. 18 U.S. Code §1030. Fraud and Related Activity in Connection with Computers. Retrieved from: gpo.gov/fdsys/pkg/USCODE-2010-title18/html/USCODE-2010-title18-partI-chap47-sec1030.htm.
8. Alert (TA13-309A) CryptoLocker Ransomware Infections. Retrieved from US - CERT: www.us-cert.gov/ncas/alerts/TA13-309A.
9. The Digital Millennium Copyright Act of 1998. Retrieved from US Copyright Office: www.copyright.gov/legislation/dmca.pdf.
10. Cyber Security Enhancement Act of 2002. Retrieved from The Library of Congress: thomas.loc.gov/cgi-bin/query/z?c107:hr3482.
11. Guidelines for Security Vulnerability Reporting and Response (2004, September 1, 2004). Retrieved from Symantec: www.symantec.com/security/OIS_Guidelines%20for%20responsible%20disclosure.pdf.
12. Schneier, Bruce (2007, January 9). "Full Disclosure of Software Vulnerabilities a 'Damned Good Idea,'" Retrieved from CSO: www.csoonline.com/article/216205/Schneier_Full_Disclosure_of_Security_Vulnerabilities_a_Damned_Good_Idea_.
13. Ranum, Marcus J. (2008, March 1). "The Vulnerability Disclosure Game: Are We More Secure?" Retrieved from CSO: www.csoonline.com/article/440110/The_Vulnerability_Disclosure_Game_Are_We_More_Secure_?CID=28073.
14. Miller, Charlie, Alex Sotirov, and Dino Dai Zovi. No More Free Bugs. Retrieved from: www.nomorefreebugs.com.
15. Mozilla Security Bug Bounty Program. Retrieved from: www.mozilla.org/security/bug-bounty.html.
16. Bugcrowd (2013, December 1). Retrieved from: bugcrowd.com/.

For Further Reading

Computer Crime & Intellectual Property Section, United States Department of Justice www.cybercrime.gov.

Federal Trade Commission, Identity Theft Site www.ftc.gov/bcp/edu/microsites/idtheft/.**IBM Internet Security Systems Vulnerability Disclosure Guidelines (X-Force team)** ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/sel03008usen/SEL03008USEN.PDF.

Privacy Rights Clearinghouse, Chronology of Data Breaches, Security Breaches 2005-Present www.privacyrights.org/data-breach.

Software Vulnerability Disclosure: The Chilling Effect, January 1, 2007 (Scott Berinato) www.csoonline.com/article/221113/Software_Vulnerability_Disclosure_The_Chilling_Effect?page=1.

Zero-Day Attack Prevention http://searchwindowssecurity.techtarget.com/generic/0,295582,sid45_gci1230354,00.html.