Chapter 6

Chapter 6 from *Ethics for the Information Age*, 3rd edition, Michael J. Quinn, 2009, Pearson Education.

"A ship in harbor is safe, but that's not what ships are for." –John Sheed

6.1 Introduction

As computers have increased in power and decreased in price, we have found more ways to use them, both at work and at home. Today, we rely upon computers for such activities as sending and receiving email, surfing the Web, shopping, managing our calendars, and keeping track of personal information. The utility of our computers and the information they hold makes computer security an important issue.

This chapter focuses on threats is to computer security. We begin by surveying computer viruses, worms, and Trojan horses. Through these mechanisms unauthorized programs can enter our computers. When executed, they can steal personal information, destroy data and even launch attacks other computers. System administrators play a key role in defending computers from outside threats. We review some of the ways administrators increase the security of the systems for which they are responsible.

Our focus then shifts from unauthorized programs to unauthorized people. People who use computers without authorization are called hackers. We look at the hacker culture of MIT in the 1950s and 1960s, as well as the origins of phone phreaking. We see how the electronic (and sometimes physical) trespassing of hackers and phreaks into telecommunications systems, combined with the public distribution of information related to break-ins, led to what Bruce Sterling calls the "hacker crackdown."

In the past few years, denial-of-service attacks have temporarily disabled Internet-based servers managed by many organizations. We examine some popular denial-of-service attack strategies and ways of combating them.

The controversy surrounding the 2000 Presidential election in Florida has raised the issue of online voting. Would voting over the Internet be superior to our present methods? We consider the benefits and risks associated with online voting.

6.2 Viruses, Worms, and Trojan Horses

There are a variety of ways in which undesired programs can become active on your computer. If you are lucky, these programs will do nothing other than consume a little CPU time and some disk space. If you are not so lucky, they may destroy valuable data stored in your computer's file system. An invading program may allow outsiders to seize control of your computer. Once this happens, they may use your computer as a depository for stolen credit card information, a Web server dishing out pornographic images, or a launch pad for spam or denial-of-service attacks on a corporate server.

"Computer pathologists" classify destructive programs as viruses, worms, or Trojan horses. In this section we describe these invasive programs and summarize technical means of defending against them.

6.2.1 Viruses

HOW VIRUSES WORK

A virus is a piece. of self-replicating code embedded within another program called the host[1]. ... When a user executes a host program infected with a virus, the virus code executes first. The virus finds another executable program stored in the computer's file system and replaces the program with a virus-infected program. After doing this, the virus allows the host program to execute, which is what the user expected to happen. If the virus does its work quickly enough, the user may be unaware of the presence of the virus.

Because a virus is attached to a host program, you may find viruses anywhere you can find program files: hard disks, floppy disks, CD-ROMs, email attachments, and so on. Viruses can be spread from machine to machine via diskettes or CDs. They may also be passed when a person downloads a file from the Internet. Sometimes viruses are attached to free computer games that people download and install on their computers. A 2003 study revealed that 45 percent of the executable files people downloaded from KaZaA. contained viruses or Trojan horses (which we will cover a little later) [2].

Today, many viruses are spread via email. An **attachment** is a file accompanying an email message. Attachments may be executable programs, or they may be word processing documents or spreadsheets containing macros, which are small pieces of executable code. If the user opens an attachment containing a virus, the virus takes control of the computer, reads the user's email address book and uses these addresses to send virus-contaminated emails to others ...

Some viruses are fairly innocent; they simply replicate. These viruses occupy deskspace and consume CPU time, but the harm they do is relatively minor. Other viruses malicious

and can cause significant damage to a person's file system.

WELL-KNOWN COMPUTER VIRUSES

The Brain virus(c. 1986) was the first virus to move from one IBM PC to another. The virus was written by the owners of a Pakistani computer store called Brain Computer Services. The said their purpose was to determine the level of software piracy in Pakistan. The virus spread internationally, but it was not malicious and caused no significant to the PCs it infected [3].

The Michelangelo virus dates back to 1991. If a PC user executes a program infected with the virus on March 6, the birthday of Renaissance painter an sculptor Michelangelo, the virus overwrites critical records on the boot disk. If the boot disk is in the user's hard drive, the contents of the drive are lost. In 1992 the media widely reported estimates that as many as five million PCs would be affected by the virus. As it turns out, only a few thousand computers were infected. Some say the whole episode was classic example of media hype [4]. Others say the extensive media publicity encouraged institutions to perform checks that would not have been done otherwise. According to them, the outbreak on March 6 was not significant because institutions had already removed the virus [5]. The Melissa virus (c. 1999) lurks inside a macro in a Word document attached to an email message. When a user activates the virus by opening the infected attachment, Melissa sends an email message with the attachment to the first 50 people in the user's address book. When Melissa first appeared, email containing the virus flooded the Internet, crashing many email servers worldwide. It infected about 100,000 computers in its first weekend. David L. Smith of New Jersey pled guilty to posting the virus at an alt.sex.usenet group using a stolen AOL account [3]. IN May 2001 Smith was sentenced to 20 months in federal prison plus 100 hours of community service. He was also fined $5,000 [6].

The Love Bug (c. 2000) is another virus lurking inside an email message. Unlike Melissa, which limits itself to the first 50 people in a victim's address book, the Love Bug creates email messages for everyone in the address book. It deletes some kinds of media files stored on the user's hard disk, and it also collects passwords and emails them to several different accounts in the Philippines. The creator of The Love Bug was a 23-year-old Filipino computer science student. When he created the virus, the Philippine had no laws against computer hacking, and he was not prosecuted [3].

VIRUSES TODAY

Commercial antivirus software packages allow computer users to detect and destroy viruses lurking on their computers. To be most effective, users must keep them up-to-date by downloading patterns corresponding to the latest viruses from the vendor's Web site.

There is evidence few people are diligent about keeping their computers virus-free. When students returned to Oberlin College in August of 2003, they were required to have their

computers checked for viruses. System administrators found viruses in 90 percent of the computers running the Windows operating system [7].

6.2.2 Worms

A **worm** is a self-contained program that spreads through a computer network by exploiting security holes in the computers connected to the network. The technical term "worm" comes from *The Shockwave Rider*, a 1975 science fiction novel written by John Brunner [8].

WANK WORM

In October 1989, NASA scientists prepared for a Space Shuttle mission that would launch a robe to Jupiter. The robot probe, named Galileo, was fueled with radioactive plutonium. Antinuclear protesters create a worm that infiltrated a NASA network. Those who logged onto an infected computer were greeted with this banner:

```
 WORMS       AGAINST       NUCLEAR           KILLERS


------------------------------------------------------------
\__  ____________  ____    _______    ___   ___   __  _____/
 \ \ \    /\    / /   / /\ \      | \ \  | |   | | / /    /
  \ \ \  /  \  / /   / /__\ \     | |\ \ | |   | |/ /    /
   \ \ \/ /\ \/ /   / /    \ \    | | \ \| |   | |\ \   /
    \_\  /__\  /___/ /______\ \___| |__\ | |___| |_\ \_/
     \                                              /
      \   Your System Has Been Officially WANKed    /
       \_____________________________________________/


You talk of times of peace for all, and then prepare for war.
```

The WANK worm took a lot of system-administrator time to eradicate, but it did not delay the launch of the Space Shuttle. It is an example of cyberterrorism: a politically motivated attack against the information technology resources of a government or its people in order to inflict damage, disrupt services, or generate fear.

CODE RED

The Code Red worm, launched on July 19, 2001, exploited a bug in Microsoft's Internet Information Services (IIS) software to spread among Windows Web servers. If U.S. English

was the default language on the server, the worm would change the server's local home page to the following message:

```
HELLO! Welcome to http://www.worm.com!

Hacked by Chinese!
```

Based on the day of the month, the Code Red worm either (1) attempted to propagate to other computers, (2) launched a denial-of-service attack against www.whitehouse.gov, or (3) slept. (We cover denial-of-service attacks in Section 6.4.) The Code Red worm spread to more than 359,000 hosts in less than 14 hours [9].

SAPPHIRE (SLAMMER)

The Sapphire worm, also known as Slammer, was released on January 25, 2003. Sapphire is notable for being the fastest-spreading computer worm in history. The number of hosts it infected doubled every 8.5 seconds, and within 10 minutes 90 percent of the vulnerable hosts were infected. The worm ended up affecting at least 78,000 computers worldwide [10].

Sapphire exploited a bug found in both Microsoft's SQL Server and SQL Server Desktop Engine. While it carried no malicious payload, the Sapphire worm overloaded networks and made database servers inaccessible. It resulted in canceled airline flights, unavailable ATMs, and failures of emergency 911 service [11].

BLASTER

The Blaster worm appeared on August 11, 2003. It exploited a bug on Windows 2000 and Windows XP computers. Blaster infected hundreds of thousands of PCs worldwide. Besides spreading to as many computers as possible, the purpose of the Blaster worm seemed to be to launch a denial-of-service attack against windowsupdate.com, the Microsoft Windows Update Web server. The apparent goal of the worm was to prevent Microsoft customers from accessing the server to download the patch needed to fix the bug. It turns out windowsupdate.com was a shortcut to the actual Web site. Microsoft thwarted the attack by deleting the shortcut [1.2].

However, the Blaster worm did have the effect of slowing down some computer systems. It disrupted the signaling of CSX freight trains and Amtrak passenger trains in the Northeast, leading to service delays [13].

SASSER

The Sasser worm, launched in April 2004, exploited a previously identified security weakness

with Windows computers. Computers with up-to-date software were safe from the worm, but it infected about 18 million computers worldwide nonetheless. The effects of the worm were relatively benign; infected computers simply shut themselves down shortly after booting. Still, the worm made millions of computers unusable and disrupted operations at Delta Airlines, the European Commission, Australian railroads, and the British coast guard [14].

After Microsoft offered a €250,000 award, a fellow student pointed the finger at German teenager Sven Jaschan, who confessed to the crime and then began working for German computer security firm Securepoint. Because he was 17 when he released the worm, Jaschan was tried in a juvenile court, which sentenced him to one-and-a-half years probation and 30 hours of community service [14, 15, 1.6] .h

INSTANT MESSAGING WORMS

Two early worms to strike instant messaging systems were Choke and Hello, which appeared in 2001. Worms were less devastating back then, because only about 141 million people used instant messaging. Today, more than 800 million people rely on instant messaging, so the impact of worms can be much greater. In April 2005 the appearance of the Kelvir worm forced the Reuters news agency to remove 60,000 subscribers from its Microsoft-based instant messaging service for 20 hours [17].

6.2.3 The Internet Worm

The Internet Worm was the first worm to affect thousands of computers. The primary source for this narrative is the excellent biography of Robert Morris in *Cyberpunk: Outlaws and Hackers on the Computer Frontier*, written by Katie Hafner and John Markoff [18].

BACKGROUND OF ROBERT TAPPAN MORRIS, JR

Robert Tappan Morris, Jr., began learning about the Unix operating system when he was still in junior high school. His father was a computer security researcher at Bell Labs, and young Morris was given an account on a Bell Labs computer that he could access from a teletype at home. It didn't take him long to discover security holes in Unix. In a 1982 interview with Gina Kolata, a writer for *Smithsonian* magazine, Morris admitted he had broken into networked computers and read other people's email. "I never told myself that there was nothing wrong with what I was doing," he said, but he acknowledged that he found breaking into systems challenging and exciting, and he admitted that he continued to do it.

As an undergraduate at Harvard, Morris majored in computer science. He quickly gained a reputation for being the computer lab's Unix expert. After his freshman year, Morris worked at Bell Labs. The result of his work was a technical paper describing a security hole

in Berkeley Unix.

While at Harvard, Morris was responsible for several computer pranks. In one of them, he installed a program that required people logging in to answer a question posed by "the Oracle" and then ask the Oracle another question. (The Oracle program worked by passing questions and answers among people trying to log in.)

DESIGNING THE WORM

Morris entered the graduate program in computer science at Cornell University in the fall of 1988. He became intrigued with the idea of creating a computer worm that would exploit bugs he had found in three Unix applications: ftp, sendmai1, and fingerd. Morris's worm used a buffer overflow attack to take control of a target computer. His "wish list" for the worm had about two dozen goals, including:

- Infect three machines per local area network.
- Only consume CPU cycles if the machines are idle.
- Avoid slow machines.
- Break passwords in order to spread to other computers.

The goal of the worm was to infect as many computers as possible. It would not destroy or corrupt data files on the machines it infected.

LAUNCHING THE WORM

On November 2, 1988, Morris learned that a fix for the ftp bug had been posted to the Internet, meaning his worm program could no longer take advantage of that security hole. However, nobody had posted fixes to the other two bugs Morris knew about. After making some last-minute changes to the worm program, he logged in to a computer at the MIT Artificial Intelligence Lab and launched the worm at about 7:30 p.m.

The worm quickly spread to thousands of computers at military installations, medical research facilities, and universities. Unfortunately, due to several bugs in the worm's programming, computers became infected with hundreds of copies of the worm, causing them to crash every few minutes or become practically unresponsive to the programs of legitimate users.

Morris contacted friends at Harvard to discuss what ought to be done next. They agreed that Andy Sudduth would anonymously post a message to the Internet. Sudduth's message is below. Harvard's computers were not affected (the security holes had already been patched), and you can tell from the last sentence that Sudduth was having a hard time believing Morris's story:

```
A Possible virus report:

There may be a virus loose on the Internet.
Here is the gist of a message I got:

I'm sorry.
Here are some steps to prevent further
transmission:

1) don't run finger, or fix it to not
overrun its stack when reading arguments.
2) recompile sendmail w/o DEBUG defined
3) don't run rexed

Hope this helps, but more, I hope it is a hoax.
```

Sudduth's email was supposed to get routed through a computer at Brown University. However, computers at Brown were already infected with the virus and did not have spare cycles to route the message. Also, the email did not have a subject line, which made it less likely to be read during a crisis. The result is that the message was read too late to be of any help to those fighting the worm.

System administrators at various universities worked frantically to stop the spread of the worm. Within a day they had examined the worm's code, discovered the bugs in sendmai1 and fingerd, and published fixes to the Internet community. In all, about 6,000 Unix computers had been infected with the worm.

After some sleuthing by reporter John Markoff, *The New York Times* named Robert Tappan Morris, Jr., as the author of the worm. Morris was suspended from Cornell University. A year later, he was the first person to receive a felony conviction under the U.S. Computer Fraud and Abuse Act. He was sentenced to 3 years probation, 400 hours of community service, and fined $10,000. His legal fees and fines exceeded $150,000.

ETHICAL EVALUATION

Was Robert Morris, Jr., wrong to unleash the Internet Worm?

A Kantian evaluation must focus on Morris's will. Did Morris have good will? His stated goal was to see how many Internet computers he could infect with the worm. While Morris did not want to crash these computers or destroy any data stored on them, his motivation was fundamentally selfish: he wanted the thrill of seeing his creation running on thousands of computers. He used others because he gained access to their machines without their permission. There is also evidence Morris knew he was using others: he took measures designed to prevent people from discovering that he was the author of the worm. From a Kantian point of view, Morris's action was wrong.

From a social contract point of view, Morris's action was also wrong. He violated the property rights of the individuals and organizations whose computers were infected by the worm. They had the right to determine who would use their computers, and they attempted to enforce this right by requiring people to identify themselves by user name and password. Morris took advantage of security holes in these computers to gain unauthorized access to them. When his worm caused these computers to become unresponsive or crash, he denied access to the legitimate users of these computers.

A utilitarian evaluation of the case focuses on the benefits and harms resulting from the spread of the worm. The principal benefit of the Internet Worm was that organizations managing these Unix computers discovered there were two significant security holes in their systems. They received the instructions they needed to patch these holes before a truly malicious intruder took advantage of them to enter their systems and do a lot of damage to their data. Of course, Morris could have produced the same beneficial result simply by contacting the system administrators at UC Berkeley and informing them of the security holes he had found.

The Internet Worm had numerous harmful consequences. A large amount of time was spent by system administrators as they defended their machines from further attacks, tracked down the problem, installed patches, and brought machines back on line. There was a disruption in email and file exchange traffic caused by computers being taken off the network. About 6,000 computers were unavailable for a day or two. During this time, many thousands of people were less productive than they could have been had the systems been up and running. Morris himself was harmed by his actions. He was suspended from Cornell and sentenced to three years of probation and 400 hours of community service. His fines and legal fees exceeded $150,000. From a utilitarian viewpoint, Morris was wrong to have released the Internet Worm.

In conclusion, Morris may not have been acting maliciously, but he was acting selfishly. If he had wanted to experiment with worms, he probably could have gotten permission to try out his creations on a local area network detached from the Internet, so that even if his worm had multiplied out of control, there would have been no fallout to the rest of the computer community. Instead, he chose to use the entire Internet as his experimental laboratory, inconveniencing thousands of people.

6.2.4 Trojan Horses

A **Trojan horse** is a program with a benign capability that conceals another, sinister purpose. When the user executes a Trojan horse, the program performs the expected beneficial task. However, the program is also performing actions unknown to, and not in the best interests of, the user.

Here are a few examples of the kinds of malicious tasks performed by Trojan horse programs:

- opening an Internet connection that allows an outsider to gain access to files on the user's computer;

- logging the keystrokes of the user and storing them in a file that the attacker can peruse to learn confidential information, such as passwords;
- looking for passwords stored on the computer and emailing them to the attacker's address;
- destroying files on the user's computer;
- launching a denial-of-service attack on a Web site;
- turning the user's computer into a proxy server that can be used to launch spam or stash information gained from illegal activities (such as stolen credit card numbers).

A **remote access Trojan** (RAT) is a Trojan horse program that gives the attacker access to the victim's computer. Two well-known RATs are Back Orifice and SubSeven. SubSeven is notable because of its easy-to-use point-and-click user interface. SubSeven consists of a client program running on the attacker's computer, and a server program running on the victim's computer. The attacker is able to capture images from the victim's monitor, record keystrokes, read and write files, watch traffic on the victim's local area network, and even control the mouse.

In order to gain access to another person's computer, the attacker must trick that person into downloading the RAT server. The most popular way to do this is to hide it inside a file posted to a Usenet newsgroup specializing in erotica. The attacker advertises the file as containing sexually explicit videos or photos. Those who download the file bring the RAT into their computer.

6.2.5 Bot Networks

A **bot** is a software program that responds to commands sent by a command-and-control program located on an external computer. The first bots supported legitimate applications: Internet Relay Chat channels and multiplayer Internet games. Today, however, bots are frequently used to support illegal activities.

For example, it's been estimated that as much as 90 percent of spam is distributed through bot networks [19]. Other bots are designed to collect personal data that can be used to steal someone's identity. Bot networks can also be used .to support distributed denial-of-service attacks, which we will discuss in Section 6.4.

6.2.6 Defensive Measures

The ability of a computer network to withstand the attacks of viruses, worms, and Trojan horses depends to a great extent on the skill and dedication of its system administrators, as well as the cooperation of the network's users.

System administrators should set up reasonable authorization and authentication mechanisms. **Authorization** is the process of determining that a user has permission to perform

a particular action. For example, a system administrator has authorization to reboot a computer, but a typical user does not. An ordinary user should not be able to examine the email messages of another user. Most operating systems create unique *user identifiers*, or uids, for its users. With each uid is information about the user's privileges. The system administrator should set user privileges appropriately to prevent one user from violating the privacy of another.

Computer security also depends upon **authentication**: determining that a person is who he claims to be. There are a variety of authentication mechanisms. The most common type is knowledge-based authentication, such as a password. Another authentication mechanism is the use of tokens, such as an identification card or smart card. A third authentication mechanism Uses biometric data, such as a fingerprint or retinal scan. It is common for highly secure computer systems to use two different authentication schemes.

The most common knowledge-based authentication scheme is the password. System administrators should install automatic password checking software that prevents users from selecting passwords that are easily guessed, such as the login name, the reverse of the login name, or a circular shift of the login name. To foil a **dictionary attack**–an automated intruder attempting to guess a password by trying every word in the dictionary–a user should always have at least one nonalphabetic character in the password.

A sure-fire way to prevent a network from being attacked by an external virus or worm is to detach it from the Internet. If it is important that the computers on the network be able to communicate with the Internet, installing a firewall is the next best thing. A **firewall** is a computer, positioned between a local network and the Internet, that monitors the packets flowing in and out. One type of firewall is a packet filter, which accepts packets only from certain trusted computers on the Internet. Another use of a firewall is to limit the number of services external computers may access. For example, many attacks have taken advantage of the *finger* program. A way to prevent such attacks is simply to not provide *finger* service to outside computers.

An important responsibility of the system administrator is to keep the operating system up-to-date with the latest patches. When the provider of an operating system announces a security patch, the announcement also informs malicious persons that a vulnerability exists. Sometimes a new worm is launched well after the patch has been made available. Up-to-date systems are not vulnerable to attacks by these worms.

A system administrator can install filters on mail servers that screen out much unwanted mail, including spam and virus-laden email. Still, some contaminated email messages are likely to get through to individual users. Virus filters associated with email readers can check incoming messages for viruses. When such a message is found, it is deleted or put in a quarantine area.

6.3 Phreaks and Hackers

Telephone and computer systems are powerful technologies, prompting some curious people to invest a lot of time and energy into learning more about how they work. A few of these

experts use the knowledge they have gained to enter systems without authorization. Once inside these systems, their actions vary widely, from simply "nosing around" to copying sensitive information to rerouting phone calls. In this section we examine two subcultures of techno-explorers: phreaks and hackers. This section relies upon three principal sources: *Cyberpunk: Outlaws and Hackers on the Computer Frontier* by Katie Hafner and John Markoff [18], *Hackers: Heroes of the Computer Revolution* by Steven Levy [20], and *The Hacker Crackdown* by Bruce Sterling [21].

6.3.1 Hackers

ORIGINAL DEFINITION OF "HACKER"

In its original meaning, a **hacker** is an explorer, a risk-taker, someone who is trying to make the system do something it has never done before. Hackers in this sense of the word abounded at MIT's Tech Model Railroad Club in the 1950s and 1960s. The Club constructed and continuously unproved an enormous HO-scale model train layout. Members of the Signals and Power Subcommittee built an elaborate electronic switching system to control the movement of the trains. Wearing chino pants, short-sleeved shirts, and pocket protectors, the most dedicated members would drink vast quantities of Coca-Cola and stay up all night to improve the system. To them, a "hack" was a newly constructed piece of equipment that not only served a useful purpose, but also demonstrated its creator's technical virtuosity. Calling someone a hacker was a sign of respect; hackers wore the label with pride. In 1959, after taking a newly created course in computer programming, some of the hackers shifted their attention from model trains to electronic computers [20].

After extensive interviews with MIT hackers, Steven Levy has summarized the "hacker ethic" with these precepts, which I quote verbatim [20]:

- Access to computers–and anything which might teach you something about the way the world works–should should be unlimited and total. Always yield to the Hands On Imperative!
- All information should be free.
- Mistrust Authority–Promote Decentralization.
- Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.
- You can create art and beauty on a computer.
- Computers can change your life for the better.

Computer security expert Dorothy Denning has observed that the will of the hacker is to make an improvement–a hacker is not malicious. A hacker is not out to destroy data or

equipment. A hacker does not commit fraud for personal profit [22].

HACKING ON THE PDP-1

The story of MIT's PDP-1 minicomputer illustrates some of the many ways that the hacker ethic translated into particular deeds.

Digital Equipment Corporation (DEC) donated the second PDP-1 it made to MIT in the summer of 1961. The PDP-1 was DEC's first product, and it came with very little software. To help remedy this deficiency, six hackers put in about 250 man-hours in a single weekend to convert an assembler for MIT's TX-0 computer to PDP-1 machine language. In one weekend they produced a program that would have taken a commercial enterprise months to complete.

Steve Russell came up with the idea of writing a shoot-em-up game for the PDP-1 that would utilize its programmable graphics display. He worked on it for over half a year, with help from other MIT hackers. In February 1962 he unveiled Spacewar, the first video game. Two players maneuvered space ships that shot torpedoes (dots) at each other. The game was an instant hit. But rather than commercialize it, the MIT group freely distributed copies of the program to other PDP-1 users.

Hackers also programmed the PDP-1 to produce the sounds needed to activate telephone switching equipment. With this capability, they were able to navigate the international telephone system. However, their excursions were simply for the sake of exploration, not for the purpose of defrauding AT&T. In fact, they reported problems they uncovered to the proper telephone service groups.

Stewart Nelson thought adding a new hardware instruction to the PDP-1 would make it better. Students had been expressly forbidden from working on the computer hardware itself, but they also knew that waiting for permission to modify the hardware would take months. Nelson decided not to ask for permission. One night, he and a few cohorts opened up the cabinet of the PDP-1 and did some rewiring. They tested the computer, and they thought they had increased the capability of the PDP-1 without affecting its other functionality. However, their testing was incomplete. The next morning, a legitimate user of the PDP-1 discovered that her program, an important weather simulation code, no longer worked. Adding a new instruction had caused another instruction to malfunction.

On another occasion, Nelson was making an unauthorized, middle-of-the night adjustment to the power supply on an MIT computer. Needing a large screwdriver, he took one from the locked cabinet of the machine shop craftsman. In the process of making the adjustment, Nelson accidentally shorted out a circuit, melting the screwdriver's handle. When the craftsman came to work the next morning, he opened the cabinet and saw the ruined

screwdriver with this sign attached: USED UP.

ETHICAL EVALUATION

Was Stewart Nelson wrong to modify the PDP-1 hardware without permission? Let's evaluate his action.

A Kantian evaluation focuses on the will behind the action, rather than its results. We might be tempted to state that Stewart Nelson's will was to improve the PDP-1 but Kant writes that we should avoid a characterization that allows an expected result to provide the motivation for an action [23]. If we ignore the expected result, what do we have left? He appears to have been acting under the maxim "Take advantage of every opportunity to demonstrate your technical skills." In his desire to demonstrate his technical prowess, Nelson made modifications to the PDP-1 without authorization. He disregarded the instructions issued by the person with legitimate authority to control access to the machine. He also disregarded the needs of the PDP-1's legitimate users, whose work depended upon the reliability of the computer. Hence Nelson treated other human beings as means to an end, and his action was wrong.

From the point of view of social contract theory, this moral problem is similar to the case of Robert Tappan Morris, Jr. By modifying a system he did not own, Nelson violated the rights of the legitimate owners and users of the computer. Hence his action was wrong.

A rule utilitarian analysis considers what would happen if everyone engaged in such behavior. Suppose everyone who had an idea about improving a system went ahead and made the change without asking permission. Perhaps most changes would make systems run better, but inevitably some people would accidentally make changes that made the system perform worse. A few supposed improvements would result in systems being broken, perhaps for long periods of time. You can also imagine situations where two different changes are being made to the same system. Either one of the changes, when made in isolation, would improve the system, but when both changes are made, they interfere with each other and make the system.unusable. If changes are not systematically recorded, the missing documentation could make systems much harder to maintain. People who simply want to use the systems would not be able to predict when they would be available and when they would not, so another long-term consequence of such actions would likely be a lowering of productivity. In the long term, allowing people to make unauthorized changes would result in less reliable systems that no one understands. We conclude that Nelson's action was wrong from a rule utilitarian point of view.

Finally, let's evaluate Nelson's action from an act utilitarian point of view. The affected persons were Nelson, the PDP-1 administrator, and the computer's users. By modifying the PDP-1, Nelson learned more about computer engineering, a benefit. We know at least one computer user was harmed as a result of Nelson's failed modification: She spent a lot of time tracking down the problem, and she could not continue with her work until the computer was fixed. In order to repair the computer, it would have to be made unavailable to its programmers, another harm. Fixing the computer had an associated cost, measured

in terms of labor and/or equipment. This cost is another negative effect. Nelson's deed most likely cost the PDP-1 administrator time and stress as he interacted with unhappy programmers and oversaw the repair job. while we have not assigned particular values to the benefit and the harms, it is likely a complete analysis would indicate Nelson's action was wrong.

It is worth considering how our analysis would change if Nelson's midnight modification of the PDP-1 had been successful and the system had worked even better after he operated on it. The Kantian, social contract, and rule utilitarian analysis did not take into account the actual result of Nelson's action, so even if his hacking had been successful, they would still have concluded that he did the wrong thing.

However, the act utilitarian analysis would be completely different. Nelson would have benefited from learning more about computer engineering. The programmers of the computer would have benefited from a more powerful instruction set. with no interruptions in the daily use of the computer, no one would have been harmed. If Nelson's hack had worked, you could conclude he did a good thing, from an act utilitarian point of view. At this point it's fair to ask: what good is an ethical theory if it can only tell you *afterward* whether your action was right or wrong? Does act utilitarianism encourage people to take morally dubious actions and then hope for the best outcome? Would you like to live in a world where everyone lived by the maxim "Better to ask forgiveness than permission"?

DUMPSTER DIVING AND SOCIAL ENGINEERING

In the 1983 movie *War Games*, a teenage hacker breaks into a military computer and nearly causes a nuclear Armageddon. After seeing the movie, a lot of teenagers were excited at the thought that they could prowl cyberspace with a home computer and a modem. A few of them became highly proficient at breaking into government and corporate computer networks.

Typically, you need a login name and password to access a computer system. Sometimes a hacker can guess a valid login name/password combination, particularly when system administrators allow users to choose short passwords or passwords that appear in a dictionary. Two other effective techniques for obtaining login names and passwords are dumpster diving and social engineering.

Dumpster diving means looking through garbage for interesting bits of information. Companies typically do not put a fence around their dumpsters In midnight rummaging sessions hackers have found user manuals, phone numbers, login names, and passwords.

Social engineering, a term coined by hacker Kevin Mitnick, refers to the manipulation of a person inside the organization to gain access to confidential information. Social engineering is easier in large organizations where people do not know each other very well. For example, a hacker may identify a system administrator and call that person, pretending to be the supervisor of his supervisor and demanding to know why he can't access a particular machine. In this situation, a cowed system administrator, eager to please his boss's boss,

may, be talked into revealing or resetting a password [24].

MALICIOUS HACKERS

In the modern use of the word, "hacking" has come to include computer break-ins accompanied by malicious behavior, such as destroying databases or stealing confidential personal information. An example of this use of the word is a story in Computerworld describing how people hacked into USA Today's Web site on July 11, 2002, and inserted fabricated news stories [25].

6.3.2 Phone Phreaking

A phone phreak is someone who manipulates the telephone system in order to communicate with others without paying for the call. The prototypical phone phreaking activity is an hours-long, coast-to-coast conference call charged to the account of a large corporation.

Historically, phone phreaks used a variety of methods to access long-distance service:

1. *Stealing long-distance telephone access codes.*

   The easiest way to do this is by "shoulder surfing" at an airport, train station, or other public place. A The phreak simply looks over people's shoulders as they key in their long distance access codes.

2. *Guessing long-distance telephone access codes.*

   Phreaks learned how to program a computer to try different codes. Running a computer all night typically resulted in about a dozen hits.

3. *Using a "blue box" to get free access to long-distance lines.*

   A "blue box" mimicked the telephone system's own access signal, a high-pitched tone of 2600 hertz.

In the 1980s phreaks used certain computer bulletin board systems (BBSs) called "pirate boards" to share stolen long-distance access codes and credit card numbers with each other.

In response to these activities, telecommunications firms installed software to detect overuse of particular long distance telephone codes. They also installed equipment to detect and trace attempts to guess access codes. The introduction of digital networks has made 2600-hertz blue boxes obsolete.

6.3.3 The Cuckoo's Egg

Clifford Stoll was a physics Ph.D. who took a job as a system administrator at Lawrence Berkeley Laboratory so he could stay in California. When Stoll was still new in the position,

he was asked to reconcile a 75-cent discrepancy between two accounting systems that charged users for computer time. He carefully searched for the missing 75 cents and discovered, to his chagrin, that an unauthorized user was logging onto Lawrence Berkeley Lab's computer. Even worse, the hacker was using LBL computers as a staging point from which to jump to computers at military installations.

Stoll observed the intruder searching these systems for files with information about such topics as the Strategic Defense Initiative and stealth technology. Eventually investigators from the FBI, the CIA, the National Security Agency, the Air Force Office of Special Investigations, and the Defense Intelligence Agency joined Stoll in the search for the hacker. The trail led to a group of West German hackers who had sold various programs, but apparently no classified information, to the KGB, the intelligence service of the Soviet Union [26].

6.3.4 Legion of Doom

Plovernet was a popular phreak/hacker BBS operated in New York and Florida; more than 500 people subscribed to it. In 1984 "Lex Luthor" created an invitation-only BBS called Legion of Doom and recruited the sharpest phreaks from Plovernet. He also created a phreak/hacker group of the same name. According to Luthor, very few users of the Legion of Doom BBS were Legion of Doom members [27]. He took the name Legion of Doom straight out of the comic books, but the authorities did not think the group's activities were the least bit humorous.

One of the ways the Legion of Doom made a name for itself was by publishing *The Legion of Doom Technical Journal*, an obvious poke at AT&T's *Bell Labs Technical Journal*. This electronic publication contained articles of interest to phreaks and hackers. All of the articles were published under pseudonyms, of course.

The introduction to a Lex Luthor article appearing in the first issue, "Identifying, Attacking, Defeating, and Bypassing Physical Security and Intrusion Detection Systems", reveals something about the interests of Legion of Doom members as well as their attitude toward the establishment:

> The reasons for writing this article are twofold:
>
> 1. To prevent the detection and/or capture of various phreaks, hackers and others, who attempt to gain access to: phone company central offices, phone closets, corporate offices, trash dumpsters, and the like.
> 2. To create an awareness and prove to various security managers, guards, and consultants how easy it is to defeat their security systems due to their lack of planning, ignorance, and just plain stupidity.

In September 1988 Legion of Doom member Robert Riggs (a.k.a. "The Prophet") broke into a BellSouth computer known as an Advanced Information Management System. The computer contained employee email, documents, and databases. Because the system had no dial-up lines, BellSouth thought the system was hidden from the public and provided minimal

security for it. It did, not even ask users for passwords. Rummaging around the system, Riggs found a document called "Bell South Standard Practice 660-225-104SV Control Office Administration of Enhanced 911 Services for Special Services and Major Account Centers dated March 1988" (the E911 Document). He copied the E911 Document to his personal computer.

Five months later, Riggs sent a copy of the E911 Document to Craig Neidorf (a.k.a. "Knight Lightning"), a pre-law student at the University of Missouri. Neidorf was the publisher of Phrack, an electronic magazine widely distributed over BBSs. Both Riggs and Neidorf had something to gain from the publication of the E911 Document. Riggs would be able to brag about the trophy he had bagged from a BellSouth computer. Neidorf would be able to demonstrate the power of the hacker underground and thumb his nose at the telecommunications companies. Still, neither wanted to get caught. They edited the E911 Document heavily, deleting the document's NOT FOR USE OR DISCLOSURE warning, phone numbers of BellSouth employees, and other identifying and sensitive information. By the time they were done, they had removed nearly half the material from the report. On February 25, 1989, Phrack published the document under the pseudonym "The Eavesdropper."

6.3.5 Fry Guy

On June 13, 1989, all calls to the Palm Beach County Probation Department in Delray Beach, Florida, were picked up by a phone-sex hotline in New York State. Phone phreaks thought it was a hilarious practical joke, but BellSouth was not amused. It immediately began a high-intensity, around-the-clock search for evidence of tampering with its computerized phone switching equipment. Investigators discovered that intruders had created new telephone numbers for themselves, manipulated proprietary databases, and reprogrammed diagnostic functions so that they could eavesdrop on conversations. If intruders Could do these things, BellSouth reasoned, they could also reprogram 911 service. what if everyone dialing 911 were connected to a phone-sex hotline?

Within a matter of weeks, police investigating the phone-sex switcheroo got a lucky break. Someone called Indiana Bell to brag about the terrible things his friends in the Legion of Doom were about to do to the telephone system, including bringing the entire network down the next Fourth of July. Indiana Bell traced the call back to its source, and the Secret Service installed pen registers at his home. The pen registers revealed long-distance telephone access code fraud. The Secret Service obtained a warrant, and on July 22 it seized all the equipment and notes of an Indiana 16-year-old with the nickname "Fry Guy."

Fry Guy had earned his nickname by using a password stolen from a local McDonald's manager to log into a McDonald's mainframe and give raises to some of his friends. He had moved on to stealing long-distance access codes and credit card numbers. He had used these stolen-credit card numbers to purchase goods and get cash advances from Western Union.

The U.S. Attorney charged Fry Guy with 11 counts of computer fraud, unauthorized computer access, and wire fraud. In September 1990 he was sentenced to 44 months probation and 400 hours of community service. By Secret Service standards, a 16-year-old

hacker was small fry. They were after his heroes, the members of the Legion of Doom, who were instigating all sorts of illegal activity through their publication of The Legion of Doom Technical Journal.

On January 15,1990–Martin Luther King, Jr. Day–AT&T's long distance service failed. Sixty thousand people lost all their telephone service, and about 70 million telephone calls could not be completed. As we will see in Chapter 7, the crash was the result of a software bug in the switching equipment used to route long-distance calls. It took AT&T engineers about nine hours to understand the general cause of the crash. A few weeks later, they found the bug.

Despite this information from AT&T, law enforcement officials had their own theories about what had caused the crash. After all, they had interviewed numerous hackers who had claimed that the Legion of Doom could bring down the nationwide telephone switching system. It seemed too great a coincidence that the system should collapse on a national holiday, just as Fry Guy had predicted. The U.S. Attorney's Office in Chicago and the Secret Service decided it was time to take serious action against hackers and phreaks.

6.3.6 *U.S. v. Riggs*

Three days after the collapse of AT&T's long distance system, two U.S. Secret Service agents visited Craig Neidorf and accused him of causing the failure. They also confronted him with the stolen E911 Document. Neidorf cooperated with the Secret Service agents. He admitted that he had received the document from Riggs, and he also admitted that he knew the document had been taken from a BellSouth computer. The next day, Secret Service appeared at Neidorf's fraternity house with a warrant, searched his room, and seized his computer.

The U.S. Attorney in Chicago charged Riggs and Neidorf with wire fraud, interstate transportation of stolen property valued at $79,449, and computer fraud. Robert Riggs pleaded guilty to wire fraud for his unauthorized access of the BellSouth computer; he ended up serving time in a federal prison. Neidorf pleaded innocent to all charges, and the case went to trial in Chicago in July 1990.

The trial was short, lasting only four days. The defense quickly established that the information in the E911 Document was in the public domain. BellSouth was actually selling to the public two documents containing more detailed information about enhanced 911 service. These documents, which could be ordered by calling a toll-free number, sold for $13 and $21, respectively, belying BellSouth's contention that the E911 Document was worth $79,449. In light of this new information, the prosecution moved to dismiss the indictments against Neidorf. The judge agreed to the motion, dismissed the jury, and declared a mistrial.

The trial against Craig Neidorf is notable for a couple of reasons. First, it demonstrates how the long history of break-ins at telecommunications companies, the posting of information on BBSs about the inner workings of phone switches, and the collapse of AT&T's long distance service all combined to created an atmosphere in which the justice system was eager to "do something" about phone phreaking and computer hacking. In its zeal to prosecute, the government uncritically accepted AT&T's inflated valuation of the E911 Doc-

ument. When the true value of the document was revealed, the government's case against Neidorf collapsed.

Second, the prosecution was careful to depict Neidorf as a thief, rather than a publisher. They could do this because Neidorf's "newsletter" was completely electronic. Viewing him as a publisher would have brought up a variety of First Amendment issues they were eager to avoid. In the early 1970s *The New York Times* and the *Washington Post* had published the Pentagon Papers, documents Daniel Ellsberg had stolen from the Pentagon describing government policies regarding the Vietnam war. The government never prosecuted these newspapers for publishing the documents. Should *Phrack* have been entitled to the same protection as *The New York Times?* The prosecutors didn't want to go there.

6.3.7 Steve Jackson Games

Another victim of the "hacker crackdown" was Steve Jackson Games (SJG) of Austin, Texas. SJG produces and sells role-playing games. In the late 1980s SJG operated a small BBS called Illuminati that provided various kinds of support to its customers, including email. Loyd Blankenship, a.k.a. "The Mentor" and an outspoken member of the Legion of Doom, happened to be a professional game designer and managing editor at SJG. Blankenship had already published the stolen E911 Document on his own BBS, called Phoenix Project.

On March 1, 1990, the Secret Service entered Blankenship's home and SJG. It seized four computers, including the one running the Illuminati BBS. According to the search warrant, which was only unsealed months later, the authorities had expected to find a copy of the stolen E911 Document on the Illuminati BBS. There was no copy of the document on any of the seized computers, and no charges were ever filed against SJG. Four months after the raid, the government returned most (but not ail) of the hardware it had seized. The disruption in business caused by the Secret Service raid forced SJG to lay off half of its employees in order to survive.

The Secret Service raid of SJG is one of the key events that led to the creation of the Electronic Frontier Foundation, a nonprofit organization that speaks out for the Constitutional rights of Americans in cyberspace. With the financial backing of the Electronic Frontier Foundation, SJG and four Illuminati BBS users sued the Secret Service. The case went to trial in 1993. The court ruled that the Secret Service had violated the Electronic Communications Privacy Act when it seized, read, and (in some cases) deleted email on the Illuminati BBS without a court order. The judge noted that investigators simply could have logged on to the Illuminati BBS to determine if the E911 Document had been posted there. He awarded SJG $50,000 in damages plus over $250,000 in attorney's fees.

6.3.8 Retrospective

In *The Hacker Crackdown* Bruce Sterling writes

> Hackers perceive hacking as a "game." This is not an entirely unreasonable or sociopathic perception. You can win or lose at hacking, succeed or fail, but it

> never feels "real." It's not simply that imaginative youngsters sometimes have a hard time telling "make-believe" from "real life." Cyberspace is *not real!* "Real" things are physical objects, such as trees and shoes and cars. Hacking takes place on a screen. Words aren't physical, numbers (even telephone numbers and credit card numbers) aren't physical. Sticks and stones may break my bones, but data will never hurt me. Computers *simulate* reality, such as computer games that simulate tank battles or dogfights or spaceships. Simulations are just make-believe, and the stuff in computers is *not real.*
>
> Consider this: If "hacking" is supposed to be so serious and real-life and dangerous, then how come *nine-year-old kids* have computers and modems? You wouldn't give a nine-year-old his own car, or his own rifle, or his own chainsaw, those things are "real."
>
> People underground are perfectly aware that the "game" is frowned upon by the powers that be. Word gets around about busts in the underground. Publicizing busts is one of the primary functions of pirate boards, but they also promulgate an attitude about them, and their own idiosyncratic ideas of justice. The users of underground boards won't complain if some guy is busted for crashing systems, spreading viruses, or stealing money by wire fraud. They may shake their heads with a sneaky grin, but they won't openly defend these practices. But when a kid is charged with some theoretical amount of theft, $264,846.14, for instance, because he sneaked into a computer and copied something, and kept it in his house on a floppy disk–this is regarded as a sign that they've drastically mistaken the immaterial game of computing for their real and boring everybody world of fatcat corporate money! [21]

We quote Sterling at length because there are parallels between this viewpoint and the mentality of the millions of people who download MP3 files containing copyrighted music. The first parallel is the attitude that intellectual property is overvalued by the establishment. How can an AT&T technical document be worth $79,000? How, can distributing songs over the Internet be a $100 billion offense? The second parallel is the use of technology as a joyride: "Hey, I can make a long-distance phone call without getting a bill!" "Hey, I can make a music CD that costs me 17 cents instead of 17 bucks!" The knowledge that actions are wrong actually makes them more fun [28]. The third parallel is the idea that breaking certain laws is not that big a deal. There is the assumption that the chance of actually getting caught is small.

There are also parallels between the response of the Secret Service to the BBSs that posted information about hacking and phreaking, and the response of the Recording Industry Association of America (RIAA) to those who made available large number of MP3 files.

On May 9, 1990, in Operation Sundevil, the Secret Service shut down 25 BBSs for posting stolen long-distance telephone access codes and facilitating the exchange of stolen credit card numbers. A press release stated:

> Today, the Secret Service is sending a clear message to those computer hackers who have decided to violate the laws of this nation in the mistaken belief that

> they can successfully avoid detection by hiding behind the relative anonymity of their computer terminals...
>
> Underground groups have been formed for the purpose of exchanging information relevant to their criminal activities. These groups often communicate with each other through message systems between computers called "bulletin boards."
>
> Our experience shows that many computer hacker suspects are no longer misguided teenagers, mischievously playing games with their computers in their bedrooms. Some are now high tech computer operators using computers to engage in unlawful conduct[21].

On September 8, 2003, the RIAA announced that its member companies had filed 261 federal lawsuits against what it called "major offenders," each of whom on average had been distributing more than 1,000 copyrighted music files through peer-to-peer networks. RIAA President Cary Sherman said:

> Nobody likes playing the heavy. There comes a time when you have to stand up and take appropriate action We've been telling people for a long time that file sharing copyrighted music is illegal, that you are not anonymous when you do it and that engaging in it can have real consequences... We hope that today's actions will convince doubters that we are serious about protecting our rights.[29]

The message from the Secret Service and the RIAA is consistent: cyberspace is real, those who break the law can be tracked down, and illegal actions in cyberspace can have severe consequences.

6.3.9 Penalties for Hacking

Under U.S. law, the maximum penalties for hacking are severe. The Computer Fraud and Abuse Act criminalizes a wide variety of hacker-related activities, including

- transmitting code (such as a virus or worm) that causes damage to a computer system;
- accessing without authorization any computer connected to the Internet, even if no files are examined, changed, or copied;
- transmitting classified government information;
- trafficking in computer passwords; computer fraud;
- computer extortion.

The maximum penalty imposed for violating the Computer Fraud and Abuse Act is 20 years in prison and a $250,000 fine.

Another federal statute related to computer hacking is the Electronic Communications Privacy Act. This law makes it illegal to intercept telephone conversations, email, or any other data transmissions. It also makes it a crime to access stored email messages without authorization.

The use of the Internet to commit fraud or transmit funds can be prosecuted under the Wire Fraud Act and/or the National Stolen Property Act. Adopting the identity of another person to carry out an illegal activity is a violation of the Identity Theft and Assumption Deterrence Act.

6.3.10 Recent Incidents

Despite potentially severe penalties for convicted hackers, computer systems continue to be compromised by outsiders. Many break-ins are orchestrated by individuals or groups with a high degree of expertise, but others are committed by ordinary computer users who simply take advantage of a security weakness.

In 2003 a hacker broke into computers at the University of Kansas and copied the personal files of 1,450 foreign students. The files contained names, Social Security numbers, passport numbers, countries of origin, and birthdates. The University of Kansas had collected the information in one place in order to comply with a Patriot Act requirement that it report the information to the Immigration and Naturalization Service [30]. In a similar incident two years later, an intruder broke into a University of Nevada, Las Vegas computer containing personal information on 5,000 foreign students [31].

Another recent case demonstrates the time and effort sometimes required to identify those responsible for computer break-ins. In April 2004 several American supercomputer installations reported that hackers had broken into computers connected to a high-speed network called TeraGrid. Before the culprits could be apprehended, they had broken into thousands of computers at American research laboratories and military installations. The hackers also accessed computers at Cisco Systems and stole some of that company's software. Security experts, FBI agents, and Swedish police worked for more than a year to identify the European culprits and bring the break-ins to an end [32].

In March 2005 someone discovered a security flaw in the online-admissions software produced by ApplyYourself and used by six business schools. The discoverer posted instructions on a Business Week online forum explaining how business school applicants could circumvent the software security system and take a look at the status of their applications. It took only nine hours to fix the flaw, but in the interim period hundreds of eager applicants had exploited the bug and peeked at their files. A week later, Carnegie Mellon University, Harvard University, and the Massachusetts Institute of Technology announced that they would not admit any of the applicants who had accessed their computer systems without authorization [33].

In 2004 and 2005 Internet cafe employee Jeanson James Ancheta created a network of about 400,000 bots, including computers operated by the U.S. Department of Defense. Adware companies, spammers, and others paid Ancheta for the use of these computers. After being arrested by the FBI, Ancheta pleaded guilty to a variety of charges, including

conspiring to violate the Computer Fraud Abuse Act and the CAN-SPAM Act. In May 2005 a federal judge sentenced Ancheta to 57 months in prison and required him to pay $15,000 in restitution to the U.S. government for infecting Department of Defense computers. Ancheta also forfeited to the government the proceeds of his illegal activity, including his 1993 BMW, more than $60,000 in cash, and his computer equipment [34, 35].

6.4 Denial-of-Service Attacks

A denial-of-service (DoS) attack is an intentional action designed to prevent legitimate users from making use of a computer service [36]. A DoS attack may involve unauthorized access to one or more computer systems, but the goal of a DoS attack is not to steal information. Instead, the aim of a DoS attack is to disrupt a computer server's ability to respond to its clients. Interfering with the normal use of computer services can result in significant harm. A company selling products and services over the Internet may lose business. A military organization may find its communications disrupted. A nonprofit organization may be unable to get its message out to the public.

A DoS attack is an example of an "asymmetric" attack, in which a single person can harm a huge organization, such as a multinational corporation or even a government. Since terrorist organizations specialize in asymmetric attacks, some fear that DoS attacks will become an important part of the terrorist arsenal [37, 38].

During the week of February 7-11, 2000, a 15-year-old initiated DoS attacks that disabled many Web sites, including Amazon.com, eBay, Yahoo, CNN.com, and Dell. The teenager, who went by the nickname "Mafiaboy," was sentenced to eight months in a juvenile detention center and a year of probation [39].

In October 2002 a DoS attack was launched against the Internet's 13 root servers which act as the Internet's ultimate authority with respect to matching domain names to IP addresses [40].

Recently, many DoS attacks have focused on blacklist services, used by ISPs to shield their customers from spam. "We're usually under attack from 5,000 to 10,000 servers at once," says Steve Linford, CEO of Spamhaus [41].

The Cooperative Association for Internet Data Analysis at the University of California estimates that 4,000 web sites suffer DoS attacks each week [42].

In this section we describe a variety of kinds of DoS attacks and some of the defensive measures that organizations can take to guard themselves against such attacks. Attackers do not want to give themselves away by initiating attacks from their own systems. Instead, they identify other computers they can use to launch their attacks. For this reason, all system administrators, not just those at targeted organizations, play a role in preventing DoS attacks.

6.4.1 Attacks that Consume Scarce Resources

The most common DoS attack is against a target system's network connection. A low-tech but effective way to do this is to cut the physical connection between the target computer and its network. Hence, it is important that organizations provide their servers with adequate physical security.

The rest of the DoS attacks we are going to describe are electronic attacks on the server or its network.

Two Internet processes establish a TCP communication link by following a precise series of steps called a "three-way handshake." The three-way handshake assures each process that the other process is ready to communicate. Suppose process X wishes to communicate with process Y. Process X initiates the handshake by sending Y a SYN message. If Y agrees to communicate with X, it replies with a SYN-ACK message, acknowledging receipt of X's SYN message. At this point the communication channel is half open. In the third step of the handshake, X sends an ACK message to Y, acknowledging receipt of Y's SYN-ACK message. At this point the connection between X and Y is open.

In a **SYN flood attack**, the attacker's computer uses IP spoofing to send the target computer a SYN message from a phony client. When the target computer receives this message, it sets up its side of the connection and replies with a SYN-ACK message. This message travels to the phony client, which cannot respond to the SYN-ACK message. While the target computer waits for the ACK message, the connection remains half open. The attacker sends the target many such spoofed messages. Since a server can only handle so many clients at one time, it may turn away legitimate users while it waits futilely for connections to complete [43].

Another form of network attack consumes all the bandwidth on the target's network by generating a large number of messages directed to that network. The **smurf attack** is an example of this form of DoS attack. The attacker first identifies routers that support broadcasting of messages to all of the computers on their local area networks. The attacker sends "ping" messages to these routers, which multiply them. A computer receiving a "ping" message is supposed to echo it. In this case, the attacker has spoofed the IP address, making it look as if the ping came from the target computer. All of the computers receiving the ping message send an echo to the target computer. In a successful attack, the flood of incoming messages saturates the target server's network.

In a third kind of DoS attack, the attacker attempts to fill all of the available space on the target computer's disk. Here are three was to fill a target computer's disk:

1. In **email bombing**, the attacker sends the target a flood of email messages. The target computer stores these email messages on its disk. By sending very long messages, the attacker can quickly fill the target's disk drive. Email bombing is usually combined with email spoofing (changing the email address of the sender) to disguise the identify of the attacker from the target.

2. The attacker creates a worm that intentionally generates a very long stream of errors. Since the target computer logs errors in a data file eventually the disk fills up.

3. The attacker breaks in to the target computer and copies over files from another site.

Most computers have a limit on the number of processes that may be active at one time. An attacker can disable the target's computer program by penetrating it with a worm program that quickly replicates. (This is how Morris's Internet Worm crashed many of the computers it infected.) Even if the target computer doe not crash, the presence of many active processes can significantly degrade the performance of the computer's CPU.

Another form of DoS attack crashes the target computer by sending it unexpected data, such as an oversized IP packet.

6.4.2 Defensive Measures

System administrators can take a variety of defensive measures to reduce the threat of DoS attacks throughout the Internet.

Ensuring the physical security of a server is an important defensive measure. Beyond the server itself, physical security encompasses the network access point, the wiring closet, and the air conditioning and power systems.

System administrators should benchmark the performance of their computer systems in order to establish baselines. Once the baselines are known, it is easier to detect aberrations that may indicate a breach of security.

Disk quota systems are another good security measure. If single users have limits on the amount of disk space they may use, then it is tougher for an intruder to create files that eat up all the disk space.

Disabling unused network services is and the prudent policy. Reducing available services reduces the options given potential attackers.

Another security measure is turning off the amplifier network capability of routers, taking a weapon out of the hands of those who wish to launch a smurf attack.

Companies have begun to create pattern-recognition software to detect DoS attacks. The software is used to discard requests for service that are coming from "clients" that have proven to be unreliable.

6.4.3 Distributed Denial-of-Service Attacks

In a distributed denial-of-service (DDoS) attack, the attacker rents access to a bot network from a bot-herder. At the selected time, the command-and-control computer sends the appropriate instructions to the bots, which launch their attack on the targeted system. Typically a DDoS attack is a smurf attack, except that now the initial "pings" are being sent from thousands of computers, so there are thousands of times more responses being echoed to the target system.

To defend against DDoS attacks, system administrators must be able to secure their computers to keep them from being infected by bots. They can also install filters that check outgoing messages for forged IP addresses. An outgoing message packet should have a "from"

address matching one of the local machines. If it does not, then the packet has been forged and should not be forwarded. Filtering outgoing messages means that even if someone has gotten into a machine, he can't use it for an attack that depends on spoofing the addresses of IP packets.

6.4.4 Blue Security

Israeli company Blue Security created a spam-deterrence system for people tired of receiving unwanted email. Blue Security sold the service to businesses, but individuals could protect their home computers for free. About half a million people signed up for this free service. Users loaded a bot called Blue Frog on their computers. The bot integrated with Yahoo! Mail, Gmail, and Hotmail, checking incoming email messages for spam. when it discovered a spam message, the bot would contact a Blue Securiserver to determine the source of the email. Then the bot would send the spammer an opt-out message [44].

Spammers who indiscriminately sent emails to millions of addresses started receiving hundreds of thousands of opt-out messages, disrupting their operations. Six of the world's top 10 spammers agreed to use Blue Security's filtering software to remove Blue Frog users from their email lists [44].

One spammer nicknamed PharmaMaster, did not back down. He threatened Blue Frog users with messages such as this one: "Unfortunately, due to the tactics used by Blue Security, you will end up receiving this message or other nonsensical spams 20-40 times more than you would normally" [19]. He followed through on his threats on May I, 2006, by sending Blue Frog users 10 to 20 times as much spam as they would normally receive[44].

The next day, PharmaMaster went after Blue Security itself. He launched a massive DDoS attack from tens of thousands of bots targeting Blue Security's servers. The huge torrent of incoming messages disabled the Blue Frog service. Later DDoS attacks focused on other companies providing Internet services to Blue Security. Finally, the spammer targeted the businesses that paid for Blue Security's services. When Blue Security realized it could not protect its business customers from DDoS attacks and virus-laced emails, it reluctantly discontinued its service. "We cannot take the responsibility for an ever- escalating cyberwar through our continued operations," wrote Eran Reshef, CEO of Blue Security. "We are discontinuing all of our anti-spam activities" [44]. Blue Security's decision to fight bots with bots always controversial was ultimately unsuccessful.

6.4.5 SATAN

In 1995 computer-security expert Dan Farmer released a program called Security Administrator Tool for Analyzing Networks (SATAN). System administrators could use SATAN to probe their computers for security weaknesses. Farmer said, "SATAN was written because we realize that computer systems are becoming more and more dependent on the network, and more vulnerable to attack" [45]. In the first few days after its release, tens of thousands of copies were downloaded.

Critics fretted that SATAN, with its easy-to-use interface, would turn relatively unskilled

teenagers into computer hackers. A security official noted it would be easy to create a script that would enable a hacker to probe hundreds of sites and report on their security holes [46]. Farmer admitted that SATAN was "a two-edged sword that can be used for good and evil."

As it turns out, a flood of SATAN-enabled computer break-ins never materialized. Apparently, it served its purpose: helping system administrators, particularly novices, identify and fix security problems with their networks.

Still, nearly two years after the release of SATAN, Dan Farmer used it to survey the security of more than 2,200 Web sites. Farmer reported that more than 60 percent of the sites were vulnerable to break-ins. About half of these sites had major security problems, even though all of the security holes probed by SATAN had been publicized by the Computer Emergency Response Team (CERT) [47].

6.5 Online Voting

6.5.1 Motivation for Online Voting

The 2000 Presidential election was one of the closest contests in U.S. history. Florida was the pivotal state; without Florida's electoral votes, neither Democrat Al Gore nor. Republican George. W. Bush had a majority of votes in the Electoral College. After a manual recount of the votes in four heavily Democratic counties, the Florida Secretary of State declared that Bush had received 2,912,790 votes to Gore's total of 2,912,253. Bush's margin of victory was incredibly small: less than 2 votes out of every 10,000 votes cast.

Most of these counties used a keypunch voting machine in which voters select a candidate by using a stylus to poke out a hole in a card next to the candidate's name. Two voting irregularities were traced to the use of these machines. The first irregularity was that sometimes the stylus doesn't punch the hole cleanly, leaving a tiny, rectangular piece of card hanging by one or more corners. Votes with "hanging chad" are typically not counted by automatic vote tabulators. The manual recount focused on identifying ballots with hanging chad that ought to have been counted. The second irregularity was that some voters in Palm Beach County were confused by its "butterfly ballot" and mistakenly punched the hole corresponding to Reform Party candidate Pat Buchanan rather than the hole for Democratic candidate Al Gore. This confusion may have cost Al Gore the votes he needed to win Florida [48].

6.5.2 Proposals

The problems with the election in Florida have led to a variety of actions to improve the reliability of voting systems in the United States. Many people have suggested that voting via the Internet be used, at least as a way of casting absentee ballots. In fact, online voting is already a reality. It was used in the 2000 Alaska Republican Presidential preference poll and the 2000 Arizona Democratic Presidential primary [49]. Local elections in the United Kingdom used online voting in 2001. One hundred thousand Americans in the military and living overseas were going to have the opportunity to vote over the Internet in the 2004

Presidential primaries as part of the Secure Electronic Registration and Voting Experiment, until the government canceled the experiment at the last minute [50].

6.5.3 Ethical Evaluation

In this section we make a utilitarian evaluation of the morality of online voting by weighing its benefits and risks. The discussion assumes that online voting would be implemented via a Web browser, though similar arguments could be made if another technology were employed.

BENEFITS OF ONLINE VOTING

Advocates of online voting say it would have numerous advantages[51]:

Online voting would give many people who ordinarily could not get to the polls the opportunity to cast a ballot from their homes.

Votes cast via the Internet could be counted much more quickly than votes cast on paper.

Electronic votes will not have any of the ambiguity associated with physical votes, such as hanging chad, erasures, etc.

Elections conducted online will cost less money than traditional elections.

Online voting will eliminate the risk of somebody tampering with a ballot box containing physical votes.

While in most elections people vote for a single candidate, other elections allow a person to vote for multiple candidates. For example, a school board may have three vacancies, and cotes may be asked to cote for three candidates. It would be easy to program the voting form to prevent people from accidentally overvoting–choosing too many candidates.

Sometimes a long, complicated ballot results in undervoting–where a voter accidentally forgets to mark a candidate for a particular office. A Web form could be designed in multiple pages so that each page had the candidates for a single office. Hence online voting could reduce undervoting.

RISKS OF ONLINE VOTING

Critics of online voting have pointed to numerous risks associated with casting ballots over the Web [51].

Online voting is unfair because it gives an unfair advantage to those who are financially better off. It will be easier for people with computers and Internet connections at home to vote.

The same system that authenticates the voter also records the ballot. This makes it more difficult to preserve the privacy of the voter.

Online voting increases the opportunities for vote solicitation and vote selling. Suppose person X agrees to vote for candidate Y in return for getting a payment from Z. If person X votes from his personal computer, he could allow person Z to watch as he cast his vote for Y, proving that he fulfilled his end of the bargain. This is much less likely to occur at an official polling place monitored by election officials.

A Web site hosting an election is an obvious target for a DDoS attack. Unlike corporate Web sites, which have attracted the attention of teenage hackers, a national election Web site could attract the attention of foreign governments or terrorists trying to disrupt the electoral process. What happens if the Web site is unavailable and people are not able to access it before the election deadline?

If voting is done from home computers, the security of the election depends on the security of these home computers. The next few paragraphs describe ways in which the security of home computers could be compromised.

A virus could change a person's vote without that person even suspecting what had happened. Many people have physical access to other people's computers, giving them the opportunity to install voter-deceiving applications in the weeks leading up to the election. Alternatively, a rogue programmer or group of programmers within Microsoft, AOL, or another consumer software company could sneak in a vote-tampering virus.

A remote access Trojan such as SubSeven lurking in a voter's computer could allow a person's vote to be observed by an outsider. A RAT could even allow an outsider to cast a ballot in lieu of the rightful voter.

An attacker could fool a user into thinking he was connected to the vote server when in actuality he Was connected to a phony vote server controlled by the attacker. For example, the attacker could send an email telling voters to click on a link to reach the polling site. When voters did so, they would be connected to the phony voting site. The attacker could ask for the voter's credentials, then use this information to connect to the real voter site and cast a vote for the candidate(s) desired by the attacker.

UTILITARIAN ANALYSIS

A utilitarian analysis must add up, the positive and negative outcomes to determine whether allowing online voting is a good action to take. Recall from Section 2.6.2 that not all outcomes have equal weight. We must consider the probability of the outcome, the value of the outcome on each affected person, and the number of people affected.

Sometimes this calculation is relatively straightforward. For example, one of the benefits of online voting is that people who voted online would not have to travel to a polling place and wait in line. Suppose online voting replaced polling places in the United States. This change would affect about 50 percent of adult Americans (the ones who actually vote) [52]. We can estimate that the average voter spends about an hour traveling to a polling place, waiting in line, and traveling back. The average annual salary in the United States is about $37,000, or about $18.00 per hour [53]. We could compute, then, that the time savings

associated with replacing polling places with online voting would be worth about $18.00 times one-half the adult population, or $9.00 for every adult.

It is more difficult to come up with reasonable weights for other outcomes. For example, a risk of online voting is that a DDoS attack may prevent legitimate voters from casting their votes before the deadline. While an election result that does not reflect the will of the voters is a great harm, the weight of this harm is reduced by three probabilities: the probability that someone would attempt a DDoS attack, the probability that a DDoS attack would be successful, and the probability that a successful DDoS attack would change the outcome of the election. Experts could have vastly different estimates of these probabilities, allowing the scales of the utilitarian evaluation to tip one way or the other.

KANTIAN ANALYSIS

A Kantian analysis of any voting system would focus on the principle that the will of each voter should be reflected in that voter's ballot. The integrity of each ballot is paramount. For this reason, every vote should leave a paper record, so that in the event of controversy a recount can be held to ensure the correctness of the election result. Eliminating paper records in order to achieve the ends of saving time and money or boosting voter turnout is wrong from a Kantian perspective.

CONCLUSIONS

We have surveyed the potential benefits and risks of holding elections online, and we have .examined the morality of online voting from a utilitarian and a Kantian point of view.

Are we holding computers up to too high a standard? After all, existing voting systems are imperfect. There are two key differences, however, between existing mechanical or electromechanical systems and the proposed online system.

Existing systems are highly localized. A single person may be able to corrupt the election process at a few voting places, but it is impossible to taint the election results across an entire state. A Web-based election system would make it much easier for a single malicious person to taint the process on a wide scale.

The second difference is that most current systems produce a paper record of the vote. where paper records do not exist, there is a push to make them mandatory [54]. When all else fails, the hard copy can be consulted to try to discern the intent of the voters. A Web-based voting system would not have paper records verified by citizens as true representations of their votes.

There is already evidence of tampering in online elections. In April 2002 Vivendi Universal, a Paris media conglomerate, held an online vote of its shareholders. Hackers caused ballots of some large shareholders to be counted as abstentions [51]. If a private election can draw the attention of a hacker, imagine how much more attractive a target a California

election web site will be!

Bruce Schneier has written, "A secure Internet voting system is theoretically possible, but it would be the first secure networked application ever created in computing history" [55].

Any election system that relies upon the security of personal computers managed by ordinary citizens will be vulnerable to electoral fraud. For this reason alone, there is a strong case to be made that a government should not allow online voting to be conducted in this way.

Summary

As computers become more fully integrated into our lives, the issue of computer security becomes more important. This chapter has described ways in which programs or people can gain unauthorized access into computer systems.

Unauthorized programs are categorized as viruses, worms, or Trojan horses. A virus is a piece of self-replicating code embedded within another program. Viruses can be found anywhere programs can be found. People can spread viruses by exchanging floppy disks or CDs or sharing files on peer-to-peer networks. A worm is a self-contained program that takes advantage of security holes to spread throughout a network. A worm is more autonomous than a virus. Once launched, a worm can spread without any human assistance. A Trojan horse is an apparently benign program that conceals a malicious purpose. Remote access Trojan horses (RATS) are often concealed inside files containing sexually explicit videos or photos. Once downloaded, a RAT enables the attacker to access the victim's computer. System administrators play an important role in securing systems against these external threats.

A person who accesses a computer without authorization is called a hacker. A Phreak is someone who manipulates the phone system in order to make free calls. As telecommunications companies began computerizing their equipment in the 1980s, the line between hackers and phreaks got blurry. A well-known group of hackers in the 1980s was the Legion of Doom. Its members wrote "how-to" articles for hackers and phreaks. These stories were widely published on BBSs. In 1990 the U.S. Justice Department and the Secret Service made a number of widely publicized raids to curtail the activities of hackers and phreaks. Many hackers and phreaks served prison sentences for their activities. However, the Secret Service violated the Electronic Communications Privacy Act when it shut down the BBS of Steve Jackson Games.

Denial-of-service (DoS) attacks prevent legitimate users from making use of a computer service. There are different kinds of DoS attacks, including physical attacks on a server, attacks that tie up a server's memory or disk space, attacks that consume all the network bandwidth to the server, and attacks that attempt to "crash" the server. In the past few years, distributed denial-of-service (DDoS) attacks have become a significant new threat to prominent Web sites. Again, system administrators can take a variety of actions to ensure the computers they are responsible for do not contribute to DoS at- tacks.

Online voting has been suggested as one way of eliminating problems associated with traditional voting systems, and experiments in online voting have already begun. While online elections would result in some benefits, the risks are extensive. In particular, a networked application is only as strong as its weakest link. If people are allowed to vote from their home computer, that is likely to be a weak link that could be exploited by those determined to affect the outcome of an election.