READING MATERIALS

THREATS TO INFORMATION SYSTEMS

TABLE OF CONTENTS

CHAPTER- 1

BUILDING A SECURE ORGANIZATION

1.1 Introduction [1-14]

It seems logical that any business, whether a commercial enterprise or a not-for-profit business, would understand that building a secure organization is important to long-term success. When a business implements and maintains a strong security posture, it can take advantage of numerous benefits. An organization that can demonstrate an infrastructure protected by robust security mechanisms can potentially see a reduction in insurance premiums being paid. A secure organization can use its security program as a marketing tool, demonstrating to clients that it values their business so much that it takes a very aggressive stance on protecting their information. But most important, a secure organization will not have to spend time and money identifying security breaches and responding to the results of those breaches. As of September 2008, according to the National Conference of State Legislatures, 44 states, the District of Columbia, and Puerto Rico had enacted legislation requiring notification of security breaches involving personal information. Security breaches can cost an organization significantly through a tarnished reputation, lost business, and legal fees. And numerous regulations, such as the Health Insurance Portability and Accountability Act (HIPAA), the Gramm–Leach–Bliley Act (GLBA), and the Sarbanes–Oxley Act, require businesses to maintain the security of information. Despite the benefits of maintaining a secure organization and the potentially devastating consequences of not doing so, many organizations have poor security mechanisms, implementations, policies, and culture.

- **Obstacles to Security**

  In attempting to build a secure organization, we should take a close look at the obstacles that make it challenging to build a totally secure organization.

- **Security Is Inconvenient**

Security, by its very nature, is inconvenient, and the more robust the security mechanisms, the more inconvenient the process becomes. Employees in an organization have a job to do; they want to get to work right away. Most security mechanisms, from passwords to multifactor authentication, are seen as roadblocks to productivity. One of the current trends in security is to add whole disk encryption to laptop computers. Although this is a highly recommended security process, it adds a second login step before a computer user can actually start working. Even if the step adds only one minute to the login process, over the course of a year this adds up to four hours of lost productivity. Some would argue that this lost productivity is balanced by the added level of security. But across a large organization, this lost productivity could prove significant.

To gain a full appreciation of the frustration caused by security measures, we have only to watch the Transportation Security Administration (TSA) security lines at any airport. Simply watch the frustration build as a particular item is run through the scanner for a third time while a passenger is running late to board his flight. Security implementations are based on a sliding scale; one end of the scale is total security and total inconvenience, the other is total insecurity and complete ease of use. When we implement any security mechanism, it should be placed on the scale where the level of security and ease of use match the acceptable level of risk for the organization.

- **Computers Are Powerful and Complex**

Home computers have become storehouses of personal materials. Our computers now contain wedding videos, scanned family photos, music libraries, movie collections, and financial and medical records. Because computers contain such familiar objects, we have forgotten that computers are very powerful and complex devices. It wasn't that long ago that computers as powerful as our desktop and laptop computers would have filled one or more very large rooms. In addition, today's computers present a "user-friendly" face to the world. Most people are unfamiliar with the way computers truly function and what goes on "behind the scenes." Things

such as the Windows Registry, ports, and services are completely unknown to most users and poorly understood by many computer industry professionals. For example, many individuals still believe that a Windows login password protects data on a computer. On the contrary— someone can simply take the hard drive out of the computer, install it as a slave drive in another computer, or place it in a USB drive enclosure, and all the data will be readily accessible.

- **Computer Users Are Unsophisticated**

Many computer users believe that because they are skilled at generating spreadsheets, word processing documents, and presentations, they "know everything about computers."

These "power users" have moved beyond application basics, but many still do not understand even basic security concepts. Many users will indiscriminately install software and visit questionable Web sites despite the fact that these actions could violate company policies. The "bad guys"—people who want to steal information from or wreak havoc on computers systems—have also identified that the average user is a weak link in the security chain. As companies began investing more money in perimeter defenses, attackers look to the path of least resistance. They send malware as attachments to email, asking recipients to open the attachment. Despite being told not to open attachments from unknown senders or simply not to open attachments at all, employees consistently violate this policy, wreaking havoc on their networks. The "I Love You Virus" spread very rapidly in this manner. More recently, phishing scams have been very effective in convincing individuals to provide their personal online banking and credit-card information. Why would an attacker struggle to break through an organization's defenses when end users are more than willing to provide the keys to bank accounts? Addressing the threat caused by untrained and unwary end users is a significant part of any security program.

- **Computers Created Without a Thought to Security**

During the development of personal computers (PCs), no thought was put into security. Early PCs were very simple affairs that had limited computing power and no keyboards and were

programmed by flipping a series of switches. They were developed almost as curiosities. Even as they became more advanced and complex, all effort was focused on developing greater sophistication and capabilities; no one thought they would have security issues. We only have to look at some of the early computers, such as the Berkeley Enterprises Geniac, the Heathkit EC-1, or the MITS Altair 8800, to understand why security was not an issue back then. The development of computers was focused on what they could do, not how they could be attacked.

As computers began to be interconnected, the driving force was providing the ability to share information, certainly not to protect it. Initially the Internet was designed for military applications, but eventually it migrated to colleges and universities, the principal tenet of which is the sharing of knowledge.

- **Current Trend Is to Share, Not Protect**

Even now, despite the stories of compromised data, people still want to share their data with everyone. And Web-based applications are making this easier to do than simply attaching a file to an email. Social networking sites such as SixApart provide the ability to share material: “Send messages, files, links, and events to your friends. Create a network of friends and share stuff. It’s free and easy . . .” In addition, many online data storage sites such as DropSend and FilesAnywhere provide the ability to share files. Although currently in the beta state of development, Swivel provides the ability to upload data sets for analysis and comparison. These sites can allow proprietary data to leave an organization by bypassing security mechanisms.

- **Data Accessible from Anywhere**

As though employees’ desire to share data is not enough of a threat to proprietary information, many business professionals want access to data from anywhere they work, on a variety of devices. To be productive, employees now request access to data and contact information on their laptops, desktops, home computers, and mobile devices. Therefore, information technology (IT) departments must now provide the ability to sync data with numerous devices. And if the IT

department can't or won't provide this capability, employees now have the power to take matters into their own hands.

Previously mentioned online storage sites can be accessed from both the home and office or anywhere there is an Internet connection. Though it might be possible to block access to some of these sites, it is not possible to block access to them all. And some can appear rather innocuous. For many, Google's free email service Gmail is a great tool that provides a very robust service for free. What few people realize is that Gmail provides more than 7 GB of storage that can also be used to store files, not just email. The Gspace plug-in for the Firefox browser provides an FTP-like interface within Firefox that gives users the ability to transfer files from a computer to their Gmail accounts. This ability to easily transfer data outside the control of a company makes securing an organization's data that much more difficult.

- **Security Isn't About Hardware and Software**

Many businesses believe that if they purchase enough equipment, they can create a secure infrastructure. Firewalls, intrusion detection systems, antivirus programs, and two-factor authentication products are just some of the tools available to assist in protecting a network and its data. It is important to keep in mind that no product or combination of products will create a secure organization by itself. Security is a process; there is no tool that you can "set and forget." All security products are only as secure as the people who configure and maintain them. The purchasing and implementation of security products should be only a percentage of the security budget. The employees tasked with maintaining the security devices should be provided with enough time, training, and equipment to properly support the products. Unfortunately, in many organizations security activities take a back seat to support activities. Highly skilled security professionals are often tasked with help-desk projects such as resetting forgotten passwords, fixing jammed printers, and setting up new employee workstations.

- **The Bad Guys Are Very Sophisticated**

At one time the computer hacker was portrayed as a lone teenager with poor social skills who would break into systems, often for nothing more than bragging rights. As ecommerce has evolved, however, so has the profile of the hacker. Now that there are vast collections of credit-card numbers and intellectual property that can be harvested, organized hacker groups have been formed to operate as businesses.

A document released in 2008 spells it out clearly: "Cybercrime companies that work much like real-world companies are starting to appear and are steadily growing, thanks to the profits they turn. Forget individual hackers or groups of hackers with common goals. Hierarchical cybercrime organizations where each cybercriminal has his or her own role and reward system are what you and your company should be worried about." Now that organizations are being attacked by highly motivated and skilled groups of hackers, creating a secure infrastructure is mandatory.

- **Management Sees Security as a Drain on the Bottom Line**

For most organizations, the cost of creating a strong security posture is seen as a necessary evil, similar to purchasing insurance. Organizations don't want to spend the money on it, but the risks of not making the purchase outweigh the costs. Because of this attitude, it is extremely challenging to create a secure organization. The attitude is enforced because requests for security tools are often supported by documents providing the average cost of a security incident instead of showing more concrete benefits of a strong security posture. The problem is exacerbated by the fact that IT professionals speak a different language than management. IT professionals are generally focused on technology, period. Management is focused on revenue. Concepts such as profitability, asset depreciation, return on investment, realization, and total cost of ownership are the mainstays of management. These are alien concepts to most IT professionals.

Realistically speaking, though it would be helpful if management would take steps to learn some fundamentals of information technology, IT professionals should take the initiative and learn some fundamental business concepts. Learning these concepts is beneficial to the organization because the technical infrastructure can be implemented in a cost-effective manner, and they are beneficial from a career development perspective for IT professionals.

A Google search on "business skills for IT professionals" will identify numerous educational programs that might prove helpful. For those who do not have the time or the inclination to attend a class, some very useful materials can be found online. One such document provided by the Government Chief Information office of New South Wales is A Guide for Government Agencies Calculating Return on Security Investment. Though extremely technical, another often cited document is Cost-Benefit Analysis for Network Intrusion Detection Systems, by Huaqiang Wei, Deb Frinke, Olivia Carter, and Chris Ritter.

Regardless of the approach that is taken, it is important to remember that any tangible cost savings or revenue generation should be utilized when requesting new security products, tools, or policies. Security professionals often overlook the value of keeping Web portals open for employees. A database that is used by a sales staff to enter contracts or purchases or check inventory will help generate more revenue if it has no downtime. A database that is not accessible or has been hacked is useless for generating revenue.

Strong security can be used to gain a competitive advantage in the marketplace. Having secured systems that are accessible 24 hours a day, seven days a week means that an organization can reach and communicate with its clients and prospective clients more efficiently. An organization that becomes recognized as a good custodian of client records and information can incorporate its security record as part of its branding. This is no different than a car company being recognized for its safety record. In discussions of cars and safety, for example, Volvo is always the first manufacturer mentioned. What must be avoided is the "sky is falling" mentality. There are indeed numerous threats to a network, but we need to be realistic in allocating resources to

protect against these threats. As of this writing, the National Vulnerability Database sponsored by the National Institute of Standards and Technology (NIST) lists 33,428 common vulnerabilities and exposures and publishes 18 new vulnerabilities per day.

In addition, the media is filled with stories of stolen laptops, credit-card numbers, and identities. The volume of threats to a network can be mind numbing. It is important to approach management with "probable threats" as opposed to "describable threats." Probable threats are those that are most likely to have an impact on your business and the ones most likely to get the attention of management.

Perhaps the best approach is to recognize that management, including the board of directors, is required to exhibit a duty of care in protecting their assets that is comparable to other organizations in their industry. When a security breach or incident occurs, being able to demonstrate the high level of security within the organization can significantly reduce exposure to lawsuits, fines, and bad press.

The goal of any discussion with management is to convince them that in the highly technical and interconnected world we live in, having a secure network and infrastructure is a "nonnegotiable requirement of doing business". An excellent resource for both IT professionals and executives that can provide insight into these issues is CERT's technical report, Governing for Enterprise Security.

1.2 Ten Steps to Building a Secure Organization [15-50]

Having identified some of the challenges to building a secure organization, let's now look at 10 ways to successfully build a secure organization. The following steps will put a business in a robust security posture.

1.2.1 Evaluate the Risks and Threats

In attempting to build a secure organization, where should you start? One commonly held belief is that you should initially identify your assets and allocate security resources based on the value of each asset. Though this approach might prove effective, it can lead to some significant vulnerabilities. An infrastructure asset might not hold a high value, for example, but it should be protected with the same effort as a high-value asset. If not, it could be an entry point into your network and provide access to valuable data. Another approach is to begin by evaluating the threats posed to your organization and your data.

- **Threats Based on the Infrastructure Model**

The first place to start is to identify risks based on an organization's infrastructure model. What infrastructure is in place that is necessary to support the operational needs of the business? A small business that operates out of one office has reduced risks as opposed to an organization that operates out of numerous facilities, includes a mobile workforce utilizing a variety of handheld devices, and offers products or services through a Web-based interface. An organization that has a large number of telecommuters must take steps to protect its proprietary information that could potentially reside on personally owned computers outside company control. An organization that has widely dispersed and disparate systems will have more risk potential than a centrally located one that utilizes uniform systems.

- **Threats Based on the Business Itself**

Are there any specific threats for your particular business? Have high-level executives been accused of inappropriate activities whereby stockholders or employees would have incentive to attack the business? Are there any individuals who have a vendetta against the company for real or imagined slights or accidents? Does the community have a history of antagonism against the organization? A risk management or security team should be asking these questions on a regular

basis to evaluate the risks in real time. This part of the security process is often overlooked due to the focus on daily workload.

- **Global Threats**

Businesses are often so narrowly focused on their local sphere of influence that they forget that by having a network connected to the Internet, they are now connected to the rest of the world. If a piece of malware identified on the other side of the globe targets the identical software used in your organization, you can be sure that you will eventually be impacted by this malware. Additionally, if extremist groups in other countries are targeting your specific industry, you will also be targeted.

Once threats and risks are identified, you can take one of four steps:

- Ignore the risk. This is never an acceptable response. This is simply burying your head in the sand and hoping the problem will go away—the business equivalent of not wearing a helmet when riding a motorcycle.
- Accept the risk. When the cost to remove the risk is greater than the risk itself, an organization will often decide to simply accept the risk. This is a viable option as long as the organization has spent the time required to evaluate the risk.
- Transfer the risk. Organizations with limited staff or other resources could decide to transfer the risk. One method of transferring the risk is to purchase specialized insurance targeted at a specific risk.
- Mitigate the risk. Most organizations mitigate risk by applying the appropriate resources to minimize the risks posed to their network.

For organizations that would like to identify and quantify the risks to their network and information assets, CERT provides a free suite of tools to assist with the project. Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) provides risk-based assessment for security assessments and planning. There are three versions of OCTAVE: the original

OCTAVE, designed for large organizations (more than 300 employees); OCTAVE-S (100 people or fewer); and OCTAVE-Allegro, which is a streamlined version of the tools and is focused specifically on information assets.

Another risk assessment tool that might prove helpful is the Risk Management Framework developed by Educause/Internet. Targeted at institutions of higher learning, the approach could be applied to other industries.

Tracking specific threats to specific operating systems, products, and applications can be time consuming. Visiting the National Vulnerability Database and manually searching for specific issues would not necessarily be an effective use of time. Fortunately, the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University has a tool called Cassandra that can be configured to notify you of specific threats to your particular products and applications.

- **Threats Based on Industry**

Businesses belonging to particular industries are targeted more frequently and with more dedication than those in other industries. Financial institutions and online retailers are targeted because “that’s where the money is.” Pharmaceutical manufacturers could be targeted to steal intellectual property, but they also could be targeted by special interest groups, such as those that do not believe in testing drugs on live animals. Identifying some of these threats requires active involvement in industry-specific trade groups in which businesses share information regarding recent attacks or threats they have identified.

1.2.2 Beware of Common Misconceptions

In addressing the security needs of an organization, it is common for professionals to succumb to some very common misconceptions. Perhaps the most common misconception is that the business is obscure, unsophisticated, or boring—simply not a target for malicious activity. Businesses must understand that any network that is connected to the Internet is a potential

target, regardless of the type of business. Attackers will attempt to gain access to a network and its systems for several reasons. The first is to look around to see what they can find. Regardless of the type of business, personnel information will more than likely be stored on one of the systems. This includes Social Security numbers and other personal information. This type of information is a target—always.

Another possibility is that the attacker will modify the information he or she finds or simply reconfigure the systems to behave abnormally. This type of attacker is not interested in financial gain; he is simply the technology version of teenagers who soap windows, egg cars, and cover property with toilet paper. He attacks because he finds it entertaining to do so. Additionally, these attackers could use the systems to store stolen "property" such as child pornography or credit-card numbers. If a system is not secure, attackers can store these types of materials on your system and gain access to them at their leisure.

The final possibility is that an attacker will use the hacked systems to mount attacks on other unprotected networks and systems. Computers can be used to mount denial-of-service (DoS) attacks, relay spam, or spread malicious software. To put it simply, no computer or network is immune from attack. Another common misconception is that an organization is immune from problems caused by employees, essentially saying, "We trust all our employees, so we don't have to focus our energies on protecting our assets from them." Though this is common for small businesses in which the owners know everyone, it also occurs in larger organizations where companies believe that they only hire "professionals." It is important to remember that no matter how well job candidates present themselves, a business can never know everything about an employee's past. For this reason it is important for businesses to conduct pre-employment background checks of all employees. Furthermore, it is important to conduct these background checks properly and completely.

Many employers trust this task to an online solution that promises to conduct a complete background check on an individual for a minimal fee. Many of these sites play on individuals'

lack of understanding of how some of these online databases are generated. These sites might not have access to the records of all jurisdictions, since many jurisdictions either do not make their records available online or do not provide them to these databases. In addition, many of the records are entered by minimum wage data-entry clerks whose accuracy is not always 100 percent.

Background checks should be conducted by organizations that have the resources at their disposal to get court records directly from the courthouses where the records are generated and stored. Some firms have a team of "runners" who visit the courthouses daily to pull records; others have a network of contacts who can visit the courts for them. Look for organizations that are active members of the National Association of Professional Background Screeners. Members of this organization are committed to providing accurate and professional results. And perhaps more important, they can provide counseling regarding the proper approach to take as well as interpreting the results of a background check.

If your organization does not conduct background checks, there are several firms that might be of assistance: Accurate Background, Inc., of Lake Forest, California; Credential Check, Inc., of Troy, Michigan; and Validity Screening Solutions in Overland Park, Kansas. The Web sites of these companies all provide informational resources to guide you in the process. (Note: For businesses outside the United States or for U.S. businesses with locations overseas, the process might be more difficult because privacy laws could prevent conducting a complete background check. The firms we've mentioned should be able to provide guidance regarding international privacy laws.)

Another misconception is that a pre-employment background check is all that is needed. Some erroneously believe that once a person is employed, he or she is "safe" and can no longer pose a threat. However, people's lives and fortunes can change during the course of employment. Financial pressures can cause otherwise law-abiding citizens to take risks they never would have thought possible. Drug and alcohol dependency can alter people's behavior as well. For these

and other reasons it is a good idea to do an additional background check when an employee is promoted to a position of higher responsibility and trust. If this new position involves handling financial responsibilities, the background check should also include a credit check.

Though these steps might sound intrusive, which is sometimes a reason cited not to conduct these types of checks, they can also be very beneficial to the employee as well as the employer. If a problem is identified during the check, the employer can often offer assistance to help the employee get through a tough time. Financial counseling and substance abuse counseling can often turn a potentially problematic employee into a very loyal and dedicated one.

Yet another common misconception involves information technology professionals. Many businesses pay their IT staff fairly high salaries because they understand that having a properly functioning technical infrastructure is important for the continued success of the company. Since the staff is adept at setting up and maintaining systems and networks, there is a general assumption that they know everything there is to know about computers. It is important to recognize that although an individual might be very knowledgeable and technologically sophisticated, no one knows everything about computers. Because management does not understand technology, they are not in a very good position to judge a person's depth of knowledge and experience in the field. Decisions are often based on the certifications a person has achieved during his or her career. Though certifications can be used to determine a person's level of competency, too much weight is given to them. Many certifications require nothing more than some time and dedication to study and pass a certification test. Some training companies also offer boot camps that guarantee a person will pass the certification test. It is possible for people to become certified without having any real-world experience with the operating systems, applications, or hardware addressed by the certification. When judging a person's competency, look at his or her experience level and background first, and if the person has achieved certifications in addition to having significant real-world experience, the certification is probably a reflection of the employee's true capabilities.

The IT staff does a great deal to perpetuate the image that they know everything about computers. One of the reasons people get involved with the IT field in the first place is because they have an opportunity to try new things and overcome new challenges. This is why when an IT professional is asked if she knows how to do something, she will always respond "Yes." But in reality the real answer should be, "No, but I'll figure it out." Though they frequently can figure things out, when it comes to security we must keep in mind that it is a specialized area, and implementing a strong security posture requires significant training and experience.

1.2.3 Provide Security Training for IT Staff—Now and Forever

Just as implementing a robust, secure environment is a dynamic process, creating a highly skilled staff of security professionals is also a dynamic process. It is important to keep in mind that even though an organization's technical infrastructure might not change that frequently, new vulnerabilities are being discovered and new attacks are being launched on a regular basis. In addition, very few organizations have a stagnant infrastructure; employees are constantly requesting new software, and more technologies are added in an effort to improve efficiencies. Each new addition likely adds additional security vulnerabilities.

It is important for the IT staff to be prepared to identify and respond to new threats and vulnerabilities. It is recommended that those interested in gaining a deep security understanding start with a vendor-neutral program. A vendor-neutral program is one that focuses on concepts rather than specific products. The SANS (SysAdmin, Audit, Network, Security) Institute offers two introductory programs: Intro to Information Security (Security 301), a five-day class designed for people just starting out in the security field, and the SANS Security Essentials Bootcamp (Security 401), a six-day class designed for people with some security experience. Each class is also available as a self-study program, and each can be used to prepare for a specific certification.

Another option is start with a program that follows the CompTia Security þ certification requirements, such as the Global Knowledge Essentials of Information Security. Some colleges offer similar programs.

Once a person has a good fundamental background in security, he should then undergo vendor-specific training to apply the concepts learned to specific applications and security devices.

A great resource for keeping up with current trends in security is to become actively involved in a security-related trade organization. The key concept here is actively involved. Many professionals join organizations so that they can add an item to the "professional affiliations" section of their re´sume´. Becoming actively involved means attending meetings on a regular basis and serving on a committee or in a position on the executive board. Though this seems like a daunting time commitment, the benefit is that the professional develops a network of resources that can be available to provide insight, serve as a sounding board, or provide assistance when a problem arises. Participating in these associations is a very cost-effective way to get up to speed with current security trends and issues. Here are some organizations that can prove helpful:

- ASIS International, the largest security-related organization in the world, focuses primarily on physical security but has more recently started addressing computer security as well.
- ISACA, formerly the Information Systems Audit and Control Association.
- High Technology Crime Investigation Association (HTCIA).
- Information Systems Security Association (ISSA).
- InfraGard, a joint public and private organization sponsored by the Federal Bureau of Investigation (FBI).

In addition to monthly meetings, many local chapters of these organizations sponsor regional conferences that are usually very reasonably priced and attract nationally recognized experts. Arguably one of the best ways to determine whether an employee has a strong grasp of information security concepts is if she can achieve the Certified Information Systems Security

Professional (CISSP) certification. Candidates for this certification are tested on their understanding of the following 10 knowledge domains:

- Access control
- Application security
- Business continuity and disaster recovery planning
- Cryptography
- Information security and risk management
- Legal, regulations, compliance, and investigations
- Operations security
- Physical (environmental) security
- Security architecture and design
- Telecommunications and network security

What makes this certification so valuable is that the candidate must have a minimum of five years of professional experience in the information security field or four years of experience and a college degree. To maintain certification, a certified individual is required to attend 120 hours of continuing professional education during the three-year certification cycle. This ensures that those holding the CISSP credential are staying up to date with current trends in security. The CISSP certification is maintained by (ISC).

1.2.4 Think "Outside the Box"

For most businesses, the threat to their intellectual assets and technical infrastructure comes from the "bad guys" sitting outside their organizations, trying to break in. These organizations establish strong perimeter defenses, essentially "boxing in" their assets. However, internal employees have access to proprietary information to do their jobs, and they often disseminate this information to areas where it is no longer under the control of the employer. This dissemination of data is generally not performed with any malicious intent, simply for employees to have access to data so that they can perform their job responsibilities more efficiently. This

also becomes a problem when an employee leaves (or when a person still-employed loses something like a laptop with proprietary information stored on it) and the organization and takes no steps to collect or control their proprietary information in the possession of their now ex-employee.

One of the most overlooked threats to intellectual property is the innocuous and now ubiquitous USB Flash drive. These devices, the size of a tube of lipstick, are the modern-day floppy disk in terms of portable data storage. They are a very convenient way to transfer data between computers. But the difference between these devices and a floppy disk is that USB Flash drives can store a very large amount of data. A 16 GB USB Flash drive has the same storage capacity as more than 10,000 floppy disks! As of this writing, a 16 GB USB Flash drive can be purchased for as little as $30. Businesses should keep in mind that as time goes by, the capacity of these devices will increase and the price will decrease, making them very attractive to employees.

These devices are not the only threat to data. Because other devices can be connected to the computer through the USB port, digital cameras, MP3 players, and external hard drives can now be used to remove data from a computer and the network to which it is connected. Most people would recognize that external hard drives pose a threat, but they would not recognize other devices as a threat. Cameras and music players are designed to store images and music, but to a computer they are simply additional mass storage devices. It is difficult for people to understand that an iPod can carry word processing documents, databases, and spreadsheets as well as music.

Fortunately, Microsoft Windows tracks the devices that are connected to a system in a Registry key, HKEY_Local_Machine\System\ControlSet00x\Enum\USBStor. It might prove interesting to look in this key on your own computer to see what types of devices have been connected.Figure 1.1 shows a wide array of devices that have been connected to a system that includes USB Flash drives, a digital camera, and several external hard drives. Windows Vista has an additional key that tracks connected devices: HKEY_Local_Machine\ Software\Microsoft\Windows Portable Devices\Devices. (Note: Analyzing the Registry is a great

way to investigate the activities of computer users. For many, however, the Registry is tough to navigate and interpret. If you are interested in understanding more about the Registry, you might want to download and play with Harlan Carvey's RegRipper)

Another threat to information that carries data outside the walls of the organization is the plethora of handheld devices currently in use. Many of these devices have the ability to send and receive email as well as create, store, and transmit word processing, spreadsheet, and PDF files. Though most employers will not purchase these devices for their employees, they are more than happy to allow their employees to sync their personally owned devices with their corporate computers. Client contact information, business plans, and other materials can easily be copied from a system. Some businesses feel that they have this threat under control because they provide their employees with corporate-owned devices and they can collect these devices when employees leave their employment.

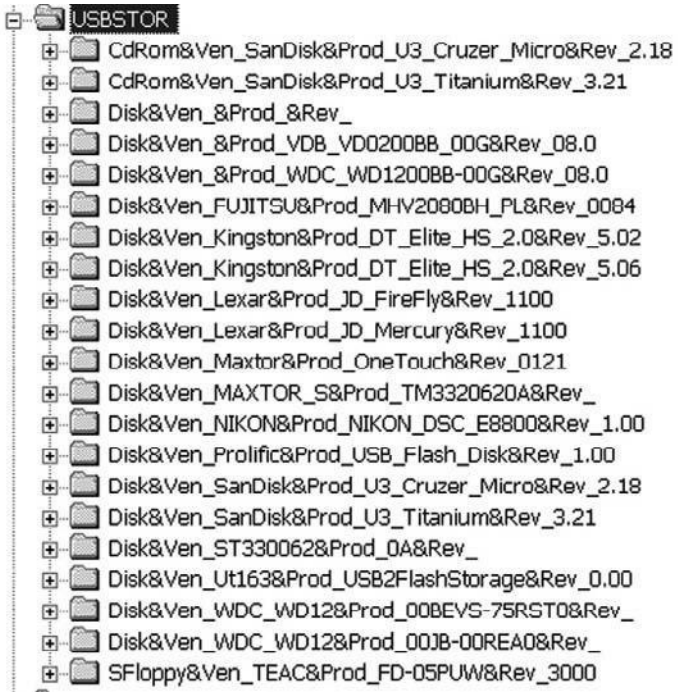


Figure 1.1: Identifying connected USB devices in the USBStor Registry key

The only problem with this attitude is that employees can easily copy data from the devices to their home computers before the devices are returned.

Because of the threat of portable data storage devices and handheld devices, it is important for an organization to establish policies outlining the acceptable use of these devices as well as implementing an enterprise-grade solution to control how, when, or if data can be copied to them. Filling all USB ports with epoxy is an inexpensive solution, but it is not really effective. Fortunately there are several products that can protect against this type of data leak. DeviceWall from Centennial Software and Mobile Security Enterprise Edition from Bluefire Security Technologies are two popular ones.

Another way that data leaves control of an organization is through the use of online data storage sites. These sites provide the ability to transfer data from a computer to an Internet-accessible location. Many of these sites provide 5 GB or more of free storage. Though it is certainly possible to blacklist these sites, there are so many, and more are being developed on a regular basis, that it is difficult if not impossible to block access to all of them. One such popular storage location is the storage space provided with a Gmail account. Gmail provides a large amount of storage space with its free accounts (7260 MB as of this writing, and growing). To access this storage space, users must use the Firefox browser with the Gspace plugin installed. Once logged in, users can transfer files simply by highlighting the file and clicking an arrow. Figure 1.2 shows the Gspace interface.

Another tool that will allow users to access the storage space in their Gmail account is the Gmail Drive shell extension. This shell extension places a drive icon in Windows Explorer, allowing users to copy files to the online storage location as though it were a normal mapped drive. Figure 1.3 shows the Gmail Drive icon in Windows Explorer.

Apple has a similar capability for those users with a MobileMe account. This drive is called iDisk and appears in the Finder. People who utilize iDisk can access the files from anywhere using a Web browser, but they can also upload files using the browser. Once uploaded, the files

are available right on the user's desktop, and they can be accessed like any other file. Figures 1.4 and 1.5 show iDisk features.

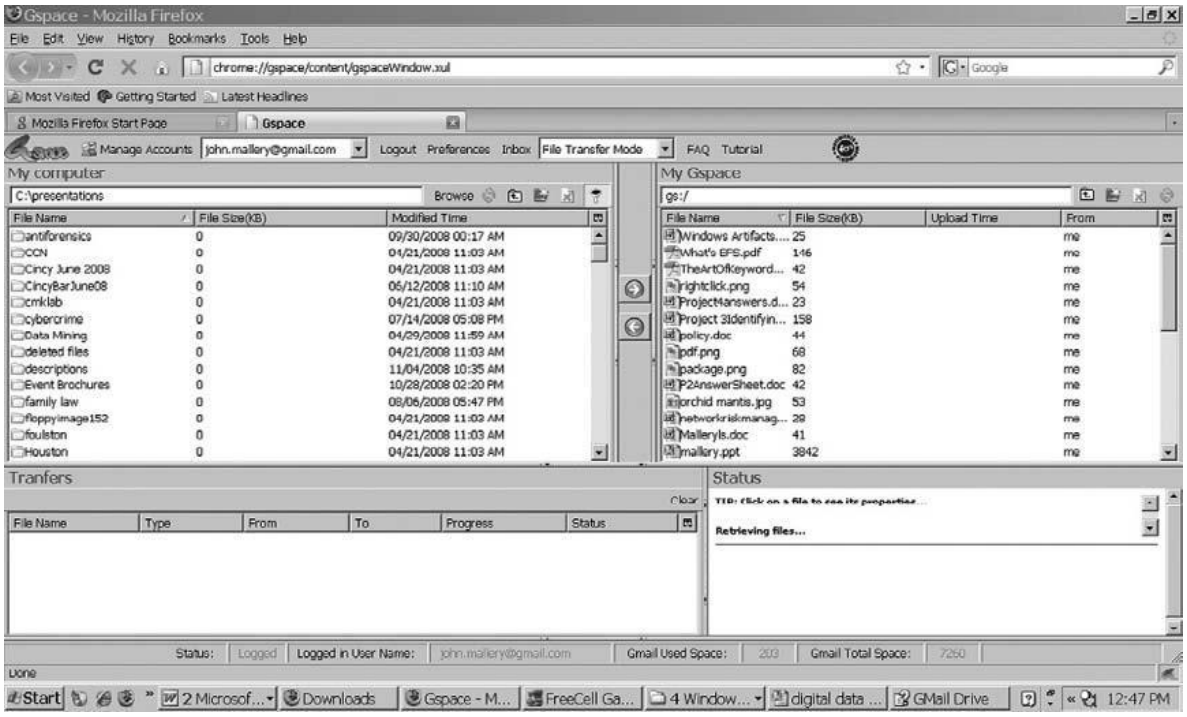


Figure 1.2: Accessing Gspace using the Firefox browser

Figure 1.3: Gmail Drive in Windows Explorer

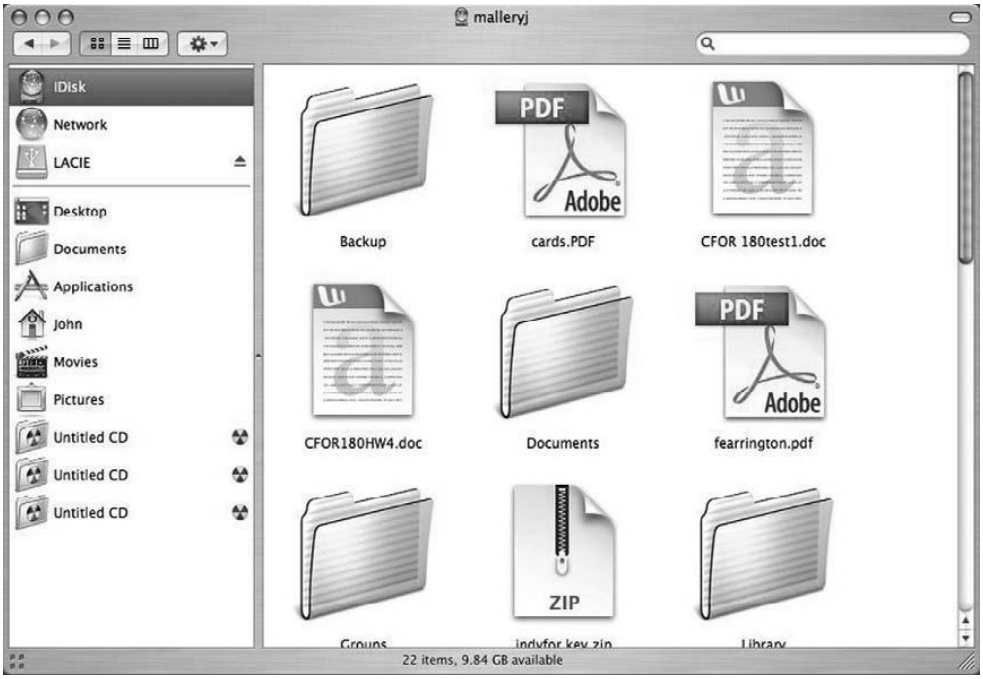


Figure 1.4: Accessing files in iDisk

1.2.5 Train Employees: Develop a Culture of Security

One of the greatest security assets is a business's own employees, but only if they have been properly trained to comply with security policies and to identify potential security problems. Many employees don't understand the significance of various security policies and implementations. As mentioned previously, they consider these policies nothing more than an inconvenience. Gaining the support and allegiance of employees takes time, but it is time well spent. Begin by carefully explaining the reasons behind any security implementation. One of the reasons could be ensuring employee productivity, but focus primarily on the security issues. File sharing using LimeWire and eMule might keep employees away from work, but they can also open up holes in a firewall. Downloading and installing unapproved software can install malicious software that can infect user systems, causing their computers to function slowly or not at all.

Perhaps the most direct way to gain employee support is to let employees know that the money needed to respond to attacks and fix problems initiated by users is money that is then not available for raises and promotions. Letting employees know that they now have some "skin in the game" is one way to get them involved in security efforts. If a budget is set aside for responding to security problems and employees help stay well within the budget, the difference between the money spent and the actual budget could be divided among employees as a bonus. Not only would employees be more likely to speak up if they notice network or system slowdowns, they would probably be more likely to confront strangers wandering through the facility.

Another mechanism that can be used to gain security allies is to provide advice regarding the proper security mechanisms for securing home computers. Though some might not see this as directly benefiting the company, keep in mind that many employees have corporate data on their home computers. This advice can come from periodic, live presentations (offer refreshments and attendance will be higher) or from a periodic newsletter that is either mailed or emailed to employees' personal addresses.

The goal of these activities is to encourage employees to approach management or the security team voluntarily. When this begins to happen on a regular basis, you will have expanded the capabilities of your security team and created a much more secure organization.

The security expert Roberta Bragg used to tell a story of one of her clients who took this concept to a high level. The client provided the company mail clerk with a WiFi hotspot detector and promised him a free steak dinner for every unauthorized wireless access point he could find on the premises. The mail clerk was very happy to have the opportunity to earn three free steak dinners.

1.2.6 Identify and Utilize Built-In Security Features of the Operating System and Applications

Many organizations and systems administrators state that they cannot create a secure organization because they have limited resources and simply do not have the funds to purchase robust security tools. This is a ridiculous approach to security because all operating systems and many applications include security mechanisms that require no organizational resources other than time to identify and configure these tools. For Microsoft Windows operating systems, a terrific resource is the online Microsoft TechNet Library. Under the Solutions Accelerators link you can find security guides for all recent Microsoft Windows operating systems. Figure 1.6 shows the table of contents for Windows 2008 Server.

Figure 1.6: Windows Server 2008 Security Guide Table of Contents

TechNet is a great resource and can provide insight into managing numerous security issues, from Microsoft Office 2007 to security risk management. These documents can assist in implementing the built-in security features of Microsoft Windows products. Assistance is needed in identifying many of these capabilities because they are often hidden from view and turned off by default.

One of the biggest concerns in an organization today is data leaks, which are ways that confidential information can leave an organization despite robust perimeter security. As mentioned previously, USB Flash drives are one cause of data leaks; another is the recovery of data found in the unallocated clusters of a computer's hard drive. Unallocated clusters, or free space, as it is commonly called, is the area of a hard drive where the operating system and applications dump their artifacts or residual data. Though this data is not viewable through a user interface, the data can easily be identified (and sometimes recovered) using a hex editor such as WinHex. Figure 1.7 shows the contents of a deleted file stored on a floppy disk being displayed by WinHex.

A computer be stolen or donated, it is very possible that someone could access the data located in unallocated clusters. For this reason, many people struggle to find an appropriate "disk-scrubbing" utility. Many such commercial utilities exist, but there is one built into Microsoft Windows operating systems. The command-line program cipher.exe is designed to display or alter the encryption of directories (files) stored on NTFS partitions.

Implementing a strong security posture often begins by making the login process more robust. This includes increasing the complexity of the login password. All passwords can be cracked, given enough time and resources, but the more difficult you make cracking a password, the greater the possibility the asset the password protects will stay protected.

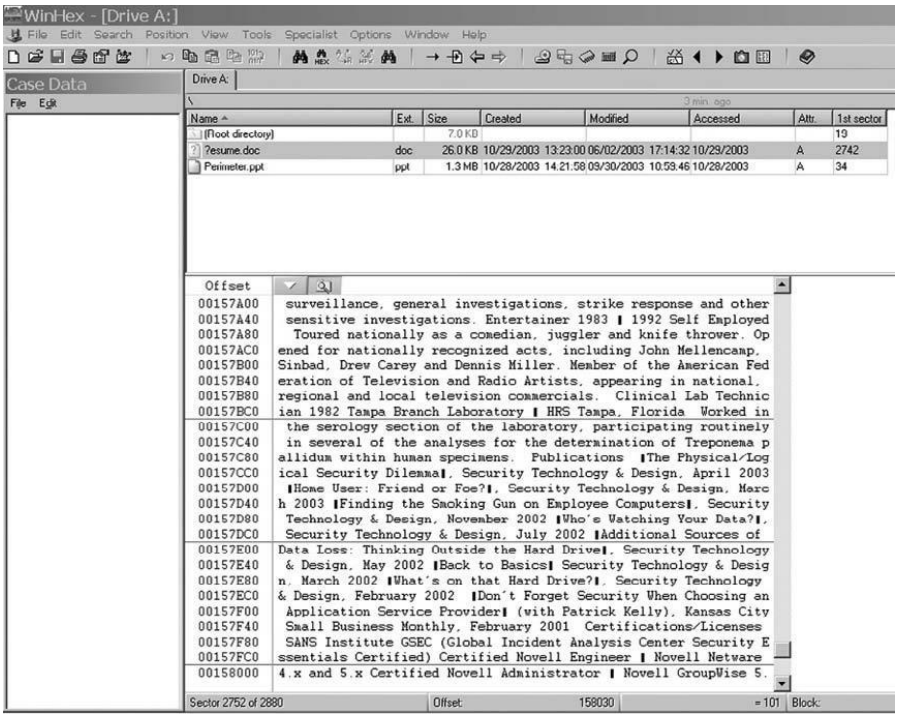


Figure 1.7: WinHex displaying the contents of a deleted Word document

All operating systems have some mechanism to increase the complexity of passwords. In Microsoft Windows XP Professional, this can be accomplished by clicking Start Control Panel Administrative Tools Local Security Policy. Under Security Settings, expand Account Policies and then highlight Password Policy. In the right-hand panel you can enable password complexity. Once this is enabled, passwords must contain at least three of the four following password groups:

• English uppercase characters (A through Z)
• English lowercase characters (a through z)
• Numerals (0 through 9)
• Nonalphabetic characters (such as !, $, #, %)

Few people even know about this command; even fewer are familiar with the /w switch. Here is a description of the switch from the program's Help file:

Removes data from available unused disk space on the entire volume. If this option is chosen, all other options are ignored. The directory specified can be anywhere in a local volume. If it is a mount point or points to a directory in another volume, the data on that volume will be removed. To use Cipher, click Start Run and type cmd. When the cmd.exe window opens, type cipher /w:folder, where folder is any folder in the volume that you want to clean, and then press Enter. Figure 1.8 shows Cipher wiping a folder.

For more on secure file deletion issues, see the author's white paper in the SANS reading room, "Secure file deletion: Fact or fiction?" Another source of data leaks is the personal and editing information that can be associated with Microsoft Office files. In Microsoft Word 2003 you can configure the application to remove personal information on save and to warn you when you are about to print, share, or send a document containing tracked changes or comments.

To access this feature, within Word click Tools Options and then click the Security tab. toward the bottom of the security window you will notice the two options described previously. Simply select the options you want to use. Figure 1.9 shows these options. Microsoft Office 2007 made this tool more robust and more accessible. A separate tool called Document Inspector can be accessed by clicking the Microsoft Office button, pointing to Prepare Document, and then clicking Inspect Document. Then select the items you want to remove. It is important to recognize that all operating systems have embedded tools to assist with security. They often require a little research to find, but the time spent in identifying them is less than the money spent on purchasing additional security products or recovering from a security breach. Though not yet used by many corporations, Mac OS X has some very robust security features, including File Vault, which creates an encrypted home folder and the ability to encrypt virtual memory. Figure 1.10 shows the security options for Mac OS X.

Figure 1.8: Cipher wiping a folder called Secretstuff

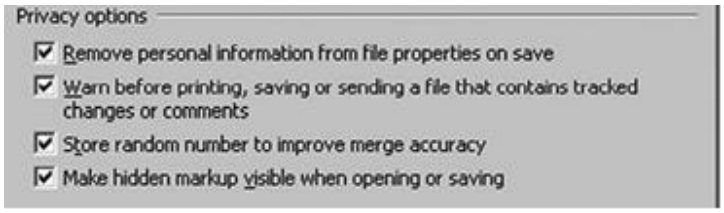


Figure 1.9: Security options for Microsoft Word 2003

Figure 1.10: Security options for Mac OS X

1.2.7 Monitor Systems

Even with the most robust security tools in place, it is important to monitor your systems. All security products are manmade and can fail or be compromised. As with any other aspect of technology, one should never rely on simply one product or tool. Enabling logging on your systems is one way to put your organization in a position to identify problem areas. The problem is, what should be logged? There are some security standards that can help with this determination. One of these standards is the Payment Card Industry Data Security Standard (PCI DSS). Requirement 10 of the PCI DSS states that organizations must "Track and monitor access to network resources and cardholder data." If you simply substitute confidential information for the phrase cardholder data, this requirement is an excellent approach to a log management program. Requirement 10 is reproduced here.

1. Establish a process for linking all access to system components (especially access done with administrative privileges such as root) to each individual user.
2. Implement automated audit trails for all system components to reconstruct the following events:
    - All individual user accesses to cardholder data
    - All actions taken by any individual with root or administrative privileges
    - Access to all audit trails
    - Invalid logical access attempts
    - Use of identification and authentication mechanisms
    - Initialization of the audit logs
    - Creation and deletion of system-level objects
3. Record at least the following audit trail entries for all system components for each event:
    - User identification
    - Type of event
    - Date and time

- Success or failure indication
- Origination of event
- Identity or name of affected data, system component, or resource

4. Synchronize all critical system clocks and times.

5. Secure audit trails so they cannot be altered:

- Limit viewing of audit trails to those with a job-related need.
- Protect audit trail files from unauthorized modifications.
- Promptly back up audit trail files to a centralized log server or media that is difficult to alter.
- Copy logs for wireless networks onto a log server on the internal LAN.
- Use file integrity monitoring and change detection software on logs to ensure that existing log data cannot be changed without generating alerts (although new data being added should not cause an alert).

6. Review logs for all system components at least daily. Log reviews must include those servers that perform security functions like intrusion detection system (IDS) and authentication, authorization, and accounting protocol (AAA) servers (e.g. RADIUS). Note: Log harvesting, parsing, and alerting tools may be used to achieve compliance.

7. Retain audit trail history for at least one year, with a minimum of three months online availability. Requirement 6 looks a little overwhelming, since few organizations have the time to manually review log files. Fortunately, there are tools that will collect and parse log files from a variety of sources. All these tools have the ability to notify individuals of a particular event. One simple tool is the Kiwi Syslog Daemon for Microsoft Windows. Figure 1.11 shows the configuration screen for setting up email alerts in Kiwi.

Additional log parsing tools include Microsoft's Log Parser and, for Unix, Swatch. Commercial tools include Cisco Security Monitoring, Analysis, and Response System (MARS) and GFI EventsManager.

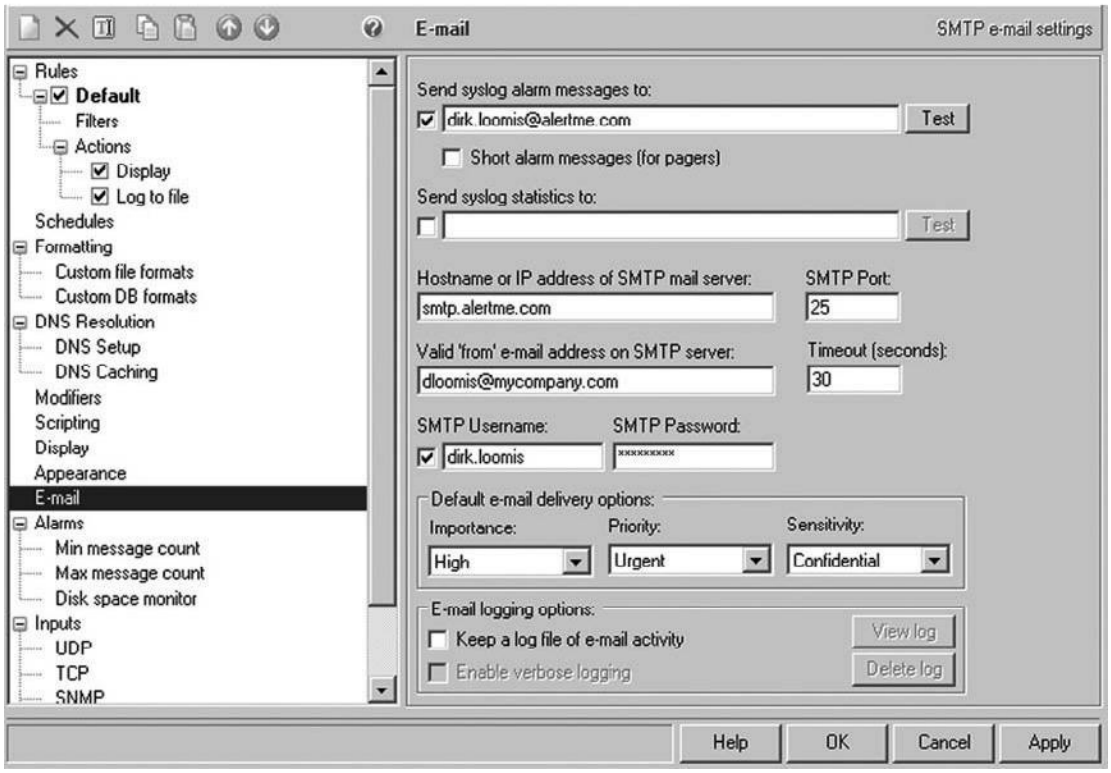


Figure 1.11: Kiwi Syslog Daemon Email Alert Configuration screen

An even more detailed approach to monitoring your systems is to install a packet-capturing tool on your network so you can analyze and capture traffic in real time. One tool that can be very helpful is Wireshark, which is "an award-winning network protocol analyzer developed by an international team of networking experts." Wireshark is based on the original packet capture tool, Ethereal. Analyzing network traffic is not a trivial task and requires some training, but it is

the perhaps the most accurate way to determine what is happening on your network. Figure 1.12 shows Wireshark monitoring the traffic on a wireless interface.

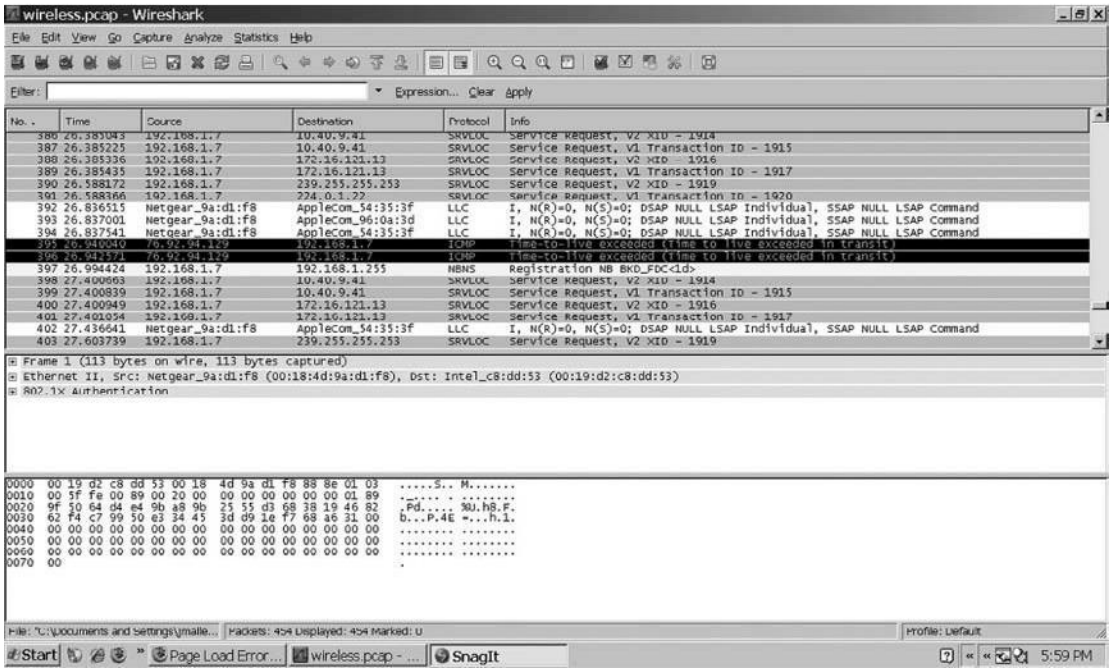


Figure 1.12: The protocol analyzer Wireshark monitoring a wireless interface

1.2.8 Hire a Third Party to Audit Security

Regardless of how talented your staff is, there is always the possibility that they overlooked something or inadvertently misconfigured a device or setting. For this reason it is very important to bring in an extra set of "eyes, ears, and hands" to review your organization's security posture.

Though some IT professionals will become paranoid having a third party review their work, intelligent staff members will recognize that a security review by outsiders can be a great learning opportunity. The advantage of having a third party review your systems is that the outsiders have experience reviewing a wide range of systems, applications, and devices in a variety of industries. They will know what works well and what might work but cause problems

in the future. They are also more likely to be up to speed on new vulnerabilities and the latest product updates. Why? Because this is all they do. They are not encumbered by administrative duties, internal politics, and help desk requests. They will be more objective than in-house staff, and they will be in a position to make recommendations after their analysis.

The third-party analysis should involve a two-pronged approach: They should identify how the network appears to attackers and how secure the system is, should attackers make it past the perimeter defenses. You don't want to have "Tootsie Pop security"—a hard crunchy shell with a soft center. The external review, often called a penetration test, can be accomplished in several ways; the first is a no knowledge approach, whereby the consultants are provided with absolutely no information regarding the network and systems prior to their analysis.

Though this is a very realistic approach, it can be time consuming and very expensive. Using this approach, consultants must use publicly available information to start enumerating systems for testing. This is a realistic approach, but a partial knowledge analysis is more efficient and less expensive. If provided with a network topology diagram and a list of registered IP addresses, the third-party reviewers can complete the review faster and the results can be addressed in a much more timely fashion. Once the penetration test is complete, a review of the internal network can be initiated. The audit of the internal network will identify open shares, unpatched systems, open ports, weak passwords, rogue systems, and many other issues.

1.2.9 Don't Forget the Basics

Many organizations spend a great deal of time and money addressing perimeter defenses and overlook some fundamental security mechanisms, as described here.

- **Change Default Account Passwords**

Nearly all network devices come preconfigured with a password/username combination. This combination is included with the setup materials and is documented in numerous locations. Very often these devices are the gateways to the Internet or other internal networks. If these default

passwords are not changed upon configuration, it becomes a trivial matter for an attacker to get into these systems. Hackers can find password lists on the Internet and vendors include default passwords in their online manuals. For example, Figure 1.13 shows the default username and password for a Netgear router.

- **Use Robust Passwords**

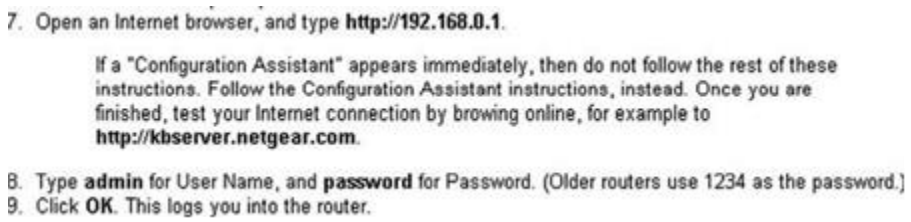
With the increased processing power of our computers and password-cracking software such as the Passware products and AccessData's Password Recovery Toolkit cracking passwords is fairly simple and straightforward. For this reason it is extremely important to create robust passwords. Complex passwords are hard for users to remember, though, so it is a challenge to create passwords that can be remembered without writing them down. One solution is to use the first letter of each word in a phrase, such as "I like to eat imported cheese from Holland." This becomes IlteicfH, which is an eight-character password using upper- and lowercase letters. This can be made even more complex by substituting an exclamation point for the letter I and substituting the number 3 for the letter e, so that the password becomes !lt3icfH. This is a fairly robust password that can be remembered easily.

- **Close Unnecessary Ports**

Ports on a computer are logical access points for communication over a network. Knowing what ports are open on your computers will allow you to understand the types of access points that exist. The well-known port numbers are 0 through 1023. Some easily recognized ports and what they are used for are listed here:

- Port 21: FTP
- Port 23: Telnet
- Port 25: SMTP
- Port 53: DNS
- Port 80: HTTP

- Port 110: POP
- Port 119: NNTP


7. Open an Internet browser, and type **http://192.168.0.1**.

   If a "Configuration Assistant" appears immediately, then do not follow the rest of these instructions. Follow the Configuration Assistant instructions, instead. Once you are finished, test your Internet connection by browing online, for example to **http://kbserver.netgear.com**.

8. Type **admin** for User Name, and **password** for Password. (Older routers use 1234 as the password.)
9. Click **OK**. This logs you into the router.

Figure 1.13: Default username and password for Netgear router

1.2.10 Patch, Patch, Patch

Nearly all operating systems have a mechanism for automatically checking for updates. This notification system should be turned on. Though there is some debate as to whether updates should be installed automatically, systems administrators should at least be notified of updates. They might not want to have them installed automatically, since patches and updates have been known to cause more problems than they solve. However, administrators should not wait too long before installing updates, because this can unnecessarily expose systems to attack. A simple tool that can help keep track of system updates is the Microsoft Baseline Security Analyzer which also will examine other fundamental security configurations.

- **Use Administrator Accounts for Administrative Tasks**

A common security vulnerability is created when systems administrators conduct administrative or personal tasks while logged into their computers with administrator rights. Tasks such as checking email, surfing the Internet, and testing questionable software can expose the computer to malicious software. This means that the malicious software can run with administrator privileges, which can create serious problems. Administrators should log into their systems using a standard user account to prevent malicious software from gaining control of their computers.

1.3 References

[1] www.ncsl.org/programs/lis/cip/priv/breachlaws.htm (October 2, 2008).

[2] Pop quiz: What was the first personal computer? www.blinkenlights.com/pc.shtml (October 26, 2008).

[3] http://www.sixapart.com (March 24, 2009).

[4] www.dropsend.com (October 26, 2008).

[5] www.filesanywhere.com (October 26, 2008).

[6] www.swivel.com (October 26, 2008).

[7] www.getgspace.com (October 27, 2008).

[8] Report: Cybercrime groups starting to operate like the Mafia, published July 16 2008 http://arstechnica.com/news.ars/post/20080716-report-cybercrime-groups-starting-to-operate-like-the-mafia.html (October 27,2008).

[9]www.gcio.nsw.gov.au/library/guidelines/resolveuid/87c81d4c6afbc1ae163024bd38aac9bd (October 29,
2008).

[10] www.csds.uidaho.edu/deb/costbenefit.pdf (October 29, 2008).

[11] Allen J, Pollak W. Why leaders should care about security. Podcast October 17, 2006. www.cert.org/
podcast/show/20061017allena.html (November 2, 2008).

[12] http://nvd.nist.gov/home.cfm (October 29, 2008).

[13] Allen J, Pollak W. Why leaders should care about security. Podcast October 17, 2006. www.cert.org/
podcast/show/20061017allena.html (November 2, 2008).

[14] www.cert.org/archive/pdf/05tn023.pdf.

[15] OCTAVE. www.cert.org/octave/ (November 2, 2008).

[16] Risk Management Framework. https://wiki.internet2.edu/confluence/display/secguide/RiskþManagementþFramework.

[17] Cassandra. https://cassandra.cerias.purdue.edu/main/index.html.

[18] National Association of Professional Background Screeners. www.napbs.com.

[19] www.accuratebackground.com.

[20] www.credentialcheck.com.

[21] www.validityscreening.com.

[22] SANS Intro to Computer Security. www.sans.org.

[23] SANS Security Essentials Bootcamp. www.sans.org.

[24]www.globalknowledge.com/training/course.asp?pageid¼9&courseid¼10242&catid=191&country=United+States.

[25] ASIS International. www.asisonline.org; ISACA, www.isaca.org; HTCIA, www.htcia.org; ISSA, www.issa.org; InfraGard, www.infragard.net.

[26] www.isc2.org.

[27]http://windowsir.blogspot.com/2008/06/portable-devices-on-vista.html (November 8, 2008).

[28] RegRipper. www.regripper.net.

[29] DeviceWall. www.devicewall.com.

[30] Bluefire Security Technologies, 1010 Hull St., Ste. 210, Baltimore, Md. 21230.

[31] Gspace. www.getgspace.com.

[32] Gmail Drive. www.viksoe.dk/code/gmail.htm.

[33] Microsoft TechNet Library. http://technet.microsoft.com/en-us/library/default.aspx.

[34] WinHex. www.x-ways.net/winhex/index-m.html.

[35] Secure file deletion: Fact or fiction? www.sans.org/reading_room/whitepapers/incident/631.php November 8, 2008).

[36] Users receive a password complexity requirements message that does not specify character group requirements for a password. http://support.microsoft.com/kb/821425 (November 8, 2008).

[37] PCI DSS. www.pcisecuritystandards.org/.

[38] Kiwi Syslog Daemon. www.kiwisyslog.com.

[39] Log Parser 2.2. www.microsoft.com/downloads/details.aspx?FamilyID¼890cd06b-abf8-4c25-91b2-
f8d975cf8c07&displaylang¼en.
[40] Swatch. http://sourceforge.net/projects/swatch/.
[41] Cisco MARS. www.cisco.com/en/US/products/ps6241/.
[42] GFI EventsManager. www.gfi.com/eventsmanager/.
[43] Wireshark. www.wireshark.org.
[44] www.phenoelit-us.org/dpl/dpl.html.
[45] Passware. www.lostpassword.com.
[46] Password Recovery Toolkit. www.accessdata.com/decryptionTool.html.
[47] ActivePorts. www.softpile.com.
[48] Fport. www.foundstone.com/us/resources/proddesc/fport.htm.
[49] Microsoft Baseline Security Analyzer. http://technet.microsoft.com/en-us/security/cc184923.aspx.
[50] 3 accused in theft of Coke secrets. Washington Post July 26, 2006 www.washingtonpost.com/wp-dyn/content/article/2006/07/05/AR2006070501717.html (November 8, 2008).

CHAPTER 2

BOTNET

2.1 Introduction[1-20]

A botnet is a collection of compromised Internet computers being controlled remotely by attackers for malicious and illegal purposes. The term comes from these programs being called robots, or bots for short, due to their automated behavior.

Bot software is highly evolved Internet malware, incorporating components of viruses, worms, spyware, and other malicious software. The person controlling a botnet is known as the botmaster or bot-herder, and he seeks to preserve his anonymity at all costs. Unlike previous malware such as viruses and worms, the motivation for operating a botnet is financial. Botnets are extremely profitable, earning their operators hundreds of dollars per day. Botmasters can either rent botnet processing time to others or make direct profits by sending spam, distributing spyware to aid in identity theft, and even extorting money from companies via the threat of a distributed denial-of-service (DDoS) attack. It is no surprise that many network security researchers believe that botnets are one of the most pressing security threats on the Internet today.

You sit down at your computer in the morning, still squinting from sleep. Your computer seems a little slower than usual, but you don't think much of it. After checking the news, you try to sign into eBay to check on your auctions. Oddly enough, your password doesn't seem to work. You try a few more times, thinking maybe you changed it recently—but without success.

Figuring you'll look into it later, you sign into online banking to pay some of those bills that have been piling up. Luckily, your favorite password still works there—so it must be a

temporary problem with eBay. Unfortunately, you are in for more bad news: The $0.00 balance on your checking and savings accounts isn't just a "temporary problem." Frantically clicking through the pages, you see that your accounts have been completely cleaned out with wire transfers to several foreign countries.

You check your email, hoping to find some explanation of what is happening. Instead of answers, you have dozens of messages from "network operations centers" around the world, informing you in no uncertain terms that your computer has been scanning, spamming, and sending out massive amounts of traffic over the past 12 hours or so. Shortly afterward, your Internet connection stops working altogether, and you receive a phone call from your service provider. They are very sorry, they explain, but due to something called "botnet activity" on your computer, they have temporarily disabled your account. Near panic now, you demand an explanation from the network technician on the other end. "What exactly is a botnet? How could it cause so much damage overnight?"

Though this scenario might sound far-fetched, it is entirely possible; similar things have happened to thousands of people over the last few years. Once a single bot program is installed on a victim computer, the possibilities are nearly endless. For example, the attacker can get your online passwords, drain your bank accounts, and use your computer as a remote-controlled "zombie" to scan for other victims, send out spam emails, and even launch DDoS attacks.

This chapter describes the botnet threat and the countermeasures available to network security professionals. First, it provides an overview of botnets, including their origins, structure, and underlying motivation. Next, the chapter describes existing methods for defending computers and networks against botnets. Finally, it addresses the most important aspect of the botnet problem: how to identify and track the botmaster in order to eliminate the root cause of the botnet problem.

2.1.1 Botnet Overview

Bots and botnets are the latest trend in the evolution of Internet malware. Their black-hat developers have built on the experience gathered from decades of viruses, worms, Trojan horses, and other malware to create highly sophisticated software that is difficult to detect and remove. Typical botnets have several hundred to several thousand members, though some botnets have been detected with over 1.5 million members. As of January 2007, Google's Vinton Cerf estimated that up to 150 million computers (about 25% of all Internet hosts) could be infected with bot software.

Before botnets, the main motivation for Internet attacks was fame and notoriety. By design, these attacks were noisy and easily detected. High-profile examples are the Melissa email worm (1999), ILOVEYOU (2000), Code Red (2001), Slammer (2003), and Sasser (2004). Though the impact of these viruses and worms was severe, the damage was relatively short-lived and consisted mainly of the cost of the outage plus man-hours required for cleanup. Once the infected files had been removed from the victim computers and the vulnerability patched, the attackers no longer had any control.

By contrast, botnets are built on the very premise of extending the attacker's control over his victims. To achieve long-term control, a bot must be stealthy during every part of its lifecycle, unlike its predecessors. As a result, most bots have a relatively small network footprint and do not create much traffic during typical operation. Once a bot is in place, the only required traffic consists of incoming commands and outgoing responses, constituting the botnet's command and control (C&C) channel. Therefore, the scenario at the beginning of the chapter is not typical of all botnets. Such an obvious attack points to either a brazen or inexperienced botmaster, and there are plenty of them.

The concept of a remote-controlled computer bot originates from Internet Relay Chat (IRC), where benevolent bots were first introduced to help with repetitive administrative tasks such as channel and nickname management. One of the first implementations of such an IRC bot was Eggdrop, originally developed in 1993 and still one of the most popular IRC bots. Over time, attackers realized that IRC was in many ways a perfect medium for large-scale botnet C&C. It provides an instantaneous one-to-many communications channel and can support very large numbers of concurrent users.

2.1.2 Botnet Topologies and Protocols

In addition to the traditional IRC-based botnets, several other protocols and topologies have emerged recently. The two main botnet topologies are centralized and peer-to-peer (P2P). Among centralized botnets, IRC is still the predominant protocol, but this trend is decreasing and several recent bots have used HTTP for their C&C channels. Among P2P botnets, many different protocols exist, but the general idea is to use a decentralized collection of peers and thus eliminate the single point of failure found in centralized botnets. P2P is becoming the most popular botnet topology because it has many advantages over centralized botnets.

- **Centralized**

Centralized botnets use a single entity (a host or a small collection of hosts) to manage all bot members. The advantage of a centralized topology is that it is fairly easy to implement and produces little overhead. A major disadvantage is that the entire botnet becomes useless if the central entity is removed, since bots will attempt to connect to nonexistent servers. To provide redundancy against this problem, many modern botnets rely on dynamic DNS services and/or fast-flux DNS techniques. In a fast-flux configuration, hundreds or thousands of compromised hosts are used as proxies to hide the identities of the true C&C servers. These hosts constantly alternate in a round-robin DNS configuration to resolve one hostname to many different IP addresses (none of which are the true IPs of C&C servers). Only the proxies know the true C&C servers, forwarding all traffic from the bots to these servers.

As we've described, the IRC protocol is an ideal candidate for centralized botnet control, and it remains the most popular among in-the-wild botmasters, although it appears that will not be true much longer. Popular examples of IRC bots are Agobot, Spybot, and Sdbot. Variants of these three families make up most active botnets today. By its nature, IRC is centralized and allows nearly instant communication among large botnets. One of the major disadvantages is that IRC traffic is not very common on the Internet, especially in an enterprise setting. As a result, standard IRC traffic can be easily detected, filtered, or blocked. For this reason, some botmasters run their IRC servers on nonstandard ports. Some even use customized IRC implementations, replacing easily recognized commands such as JOIN and PRIVMSG with other text. Despite these countermeasures, IRC still tends to stick out from the regular Web and email traffic due to uncommon port numbers.

Recently, botmasters have started using HTTP to manage their centralized botnets. The advantage of using regular Web traffic for C&C is that it must be allowed to pass through virtually all firewalls, since HTTP comprises a majority of Internet traffic. Even closed firewalls that only provide Web access (via a proxy service, for example) will allow HTTP traffic to pass. It is possible to inspect the content and attempt to filter out malicious C&C traffic, but this is not feasible due to the large number of existing bots and variants.

If botmasters use HTTPS (HTTP encrypted using SSL/TLS), then even content inspection becomes useless and all traffic must be allowed to pass through the firewall. However, a disadvantage of HTTP is that it does not provide the instant communication and built-in, scale-up properties of IRC: Bots must manually poll the central server at specific intervals. With large botnets, these intervals must be large enough and distributed well to avoid overloading the server with simultaneous requests. Examples of HTTP bots are Bobax and Rustock, with Rustock using a custom encryption scheme on top of HTTP to conceal its C&C traffic.

- **Peer-to-Peer**

As defenses against centralized botnets have become more effective, more and more botmasters are exploring ways to avoid the pitfalls of relying on a centralized architecture and therefore a single point of failure. Symantec reports a “steady decrease” in centralized IRC botnets and predicts that botmasters are now “accelerating their shift to newer, stealthier control methods, using protocols such as peer-to-peer”. In the P2P model, no centralized server exists, and all member nodes are equally responsible for passing on traffic. “If done properly, [P2P] makes it near impossible to shut down the botnet as a whole. It also provides anonymity to the [botmaster], because they can appear as just another node in the network,” says security researcher Joe Stewart of Lurhq. There are many protocols available for P2P networks, each differing in the way nodes first join the network and the role they later play in passing traffic along. Some popular protocols are BitTorrent, WASTE, and Kademlia. Many of these protocols were first developed for benign uses, such as P2P file sharing.

One of the first malicious P2P bots was Sinit, released in September 2003. It uses random scanning to find peers rather than relying on one of the established P2P bootstrap protocols. As a result, Sinit often has trouble finding peers, which results in overall poor connectivity. Due to the large amount of scanning traffic, this bot is easily detected by intrusion detection systems (IDSs). Another advanced bot using the P2P approach is Nugache, released in April 2006. It initially connects to a list of predefined peers to join the P2P network and then downloads a list of active peer nodes from there. This implies that if the “seed” hosts can be shut down, no new bots will be able to join the network, but existing nodes can still function. Nugache encrypts all communications, making it harder for IDSs to detect and increasing the difficulty of manual analysis by researchers. Nugache is seen as one of the first more sophisticated P2P bots, paving the way for future enhancements by botnet designers.

The most famous P2P bot so far is Peacomm, more commonly known as the Storm Worm. It started spreading in January 2007 and continues to have a strong presence. To communicate with

peers, it uses the Overnet protocol, based on the Kademlia P2P protocol. For bootstrapping, it uses a fixed list of peers (146 in one observed instance) distributed along with the bot. Once the bot has joined Overnet, the botmaster can easily update the binary and add components to extend its functionality. Often the bot is configured to automatically retrieve updates and additional components, such as an SMTP server for spamming, an email address harvesting tool, and a DoS module. Like Nugache, all of Peacomm's communications are encrypted, making it extremely hard to observe C&C traffic or inject commands appearing to come from the botmaster. Unlike centralized botnets relying on a dynamic DNS provider, Peacomm uses its own P2P network as a distributed DNS system that has no single point of failure. The fixed list of peers is a potential weakness, although it would be challenging to take all these nodes offline. Additionally, the attackers can always set up new nodes and include an updated peer list with the bot, resulting in an "arms race" to shut down malicious nodes.

2.3 Typical Bot Life Cycle [5, 20]

Regardless of the topology being used, the typical life cycle of a bot is similar:

1. Creation. First, the botmaster develops his bot software, often reusing existing code and adding custom features. He might use a test network to perform dry runs before deploying the bot in the wild.

2. Infection. There are many possibilities for infecting victim computers, including the following four. Once a victim machine becomes infected with a bot, it is known as a zombie.

• **Software vulnerabilities.** The attacker exploits a vulnerability in a running service to automatically gain access and install his software without any user interaction. This was the method used by most worms, including the infamous Code Red and Sasser worms.

• **Drive-by download.** The attacker hosts his file on a Web server and entices people to visit the site. When the user loads a certain page, the software is automatically installed without user

interaction, usually by exploiting browser bugs, misconfigurations, or unsecured ActiveX controls.

• **Trojan horse**. The attacker bundles his malicious software with seemingly benign and useful software, such as screen savers, antivirus scanners, or games. The user is fully aware of the installation process, but he does not know about the hidden bot functionality.

• **Email attachment:** Although this method has become less popular lately due to rising user awareness, it is still around. The attacker sends an attachment that will automatically install the bot software when the user opens it, usually without any interaction. This was the primary infection vector of the ILOVEYOU email worm from 2000. The recent Storm Worm successfully used enticing email messages with executable attachments to lure its victims.

3. **Rallying.** After infection, the bot starts up for the first time and attempts to contact its C&C server(s) in a process known as rallying. In a centralized botnet, this could be an IRC or HTTP server, for example. In a P2P botnet, the bots perform the bootstrapping protocol required to locate other peers and join the network. Most bots are very fault-tolerant, having multiple lists of backup servers to attempt if the primary ones become unavailable. Some C&C servers are configured to immediately send some initial commands to the bot (without botmaster intervention). In an IRC botnet, this is typically done by including the commands in the C&C channel's topic.

4. Waiting. Having joined the C&C network, the bot waits for commands from the botmaster. During this time, very little (if any) traffic passes between the victim and the C&C servers. In an IRC botnet, this traffic would mainly consist of periodic keep-alive messages from the server.

5. **Executing.** Once the bot receives a command from the botmaster, it executes it and returns any results to the botmaster via the C&C network. The supported commands are only limited by the botmaster's imagination and technical skills. Common commands are in line with the major uses of botnets: scanning for new victims, sending spam, sending DoS floods, setting up traffic

redirection, and many more. Following execution of a command, the bot returns to the waiting state to await further instructions. If the victim computer is rebooted or loses its connection to the C&C network, the bot resumes in the rallying state. Assuming it can reach its C&C network, it will then continue in the waiting state until further commands arrive.

Figure 2.1 shows the detailed infection sequence in a typical IRC-based botnet.

- An existing botnet member computer launches a scan, then discovers and exploits a vulnerable host.
- Following the exploit, the vulnerable host is made to download and install a copy of the bot software, constituting an infection.
- When the bot starts up on the vulnerable host, it enters the rallying state: It performs a DNS lookup to determine the current IP of its C&C server.
- The new bot joins the botnet's IRC channel on the C&C server for the first time, now in the waiting state.
- The botmaster sends his commands to the C&C server on the botnet's IRC channel.
- The C&C server forwards the commands to all bots, which now enter the executing state.

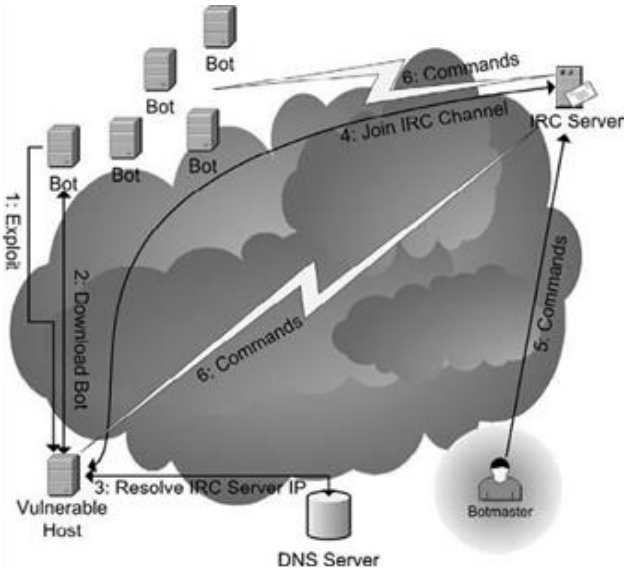


Figure 2.1: Infection sequence of a typical centralized IRC-based botnet

2.4 The Botnet Business Model [2,4,21-32]

Unlike the viruses and worms of the past, botnets are motivated by financial profit. Organized crime groups often use them as a source of income, either by hiring "freelance" botmasters or by having their own members create botnets. As a result, network security professionals are up against motivated, well-financed organizations that can often hire some of the best minds in computers and network security. This is especially true in countries such as Russia, Romania, and other Eastern European nations where there is an abundance of IT talent at the high school and university level but legitimate IT job prospects are very limited. In such an environment, criminal organizations easily recruit recent graduates by offering far better opportunities than the legitimate job market. One infamous example of such a crime organization is the Russian Business Network (RBN), a Russian Internet service provider (ISP) that openly supports criminal activity. They are responsible for the Storm Worm (Peacomm), the March 2007 DDoS attacks on Estonia and a high-profile attack on the Bank of India in August 2007, along with many other attacks.

It might not be immediately obvious how a collection of computers can be used to cause havoc and produce large profits. The main point is that botnets provide anonymous and distributed access to the Internet. The anonymity makes the attackers untraceable, and a botnet's distributed nature makes it extremely hard to shut down. As a result, botnets are perfect vehicles for criminal activities on the Internet. Some of the main profit-producing methods are explained here, but criminals are always devising new and creative ways to profit from botnets:

- **Spam.** Spammers send millions of emails advertising phony or overpriced products, phishing for financial data and login information, or running advance-fee schemes such as the Nigerian 419 scam. Even if only a small percentage of recipients respond to this spam, the payoff is considerable for the spammer. It is estimated that up to 90% of all spam originates from botnets.

- **DDoS and extortion.** Having amassed a large number of bots, the attacker contacts an organization and threatens to launch a massive DDoS attack, shutting down its Web site for several hours or even days. Another variation on this method is to find vulnerabilities, use them steal financial or confidential data, and then demand money for the "safe return" of the data and to keep it from being circulated in the underground economy. Often, companies would rather pay off the attacker to avoid costly downtime, lost sales, and the lasting damage to its reputation that would result from a DDoS attack or data breach.

- **Identity theft.** Once a bot has a foothold on a victim's machine, it usually has complete control. For example, the attacker can install keyloggers to record login and password information, search the hard drive for valuable data, or alter the DNS configuration to redirect victims to look-alike Web sites and collect personal information, known as pharming. Using the harvested personal information, the attacker can make fraudulent credit card charges, clean out the victim's bank account, and apply for credit in the victim's name, among many other things.

- **Click fraud.** In this scenario, bots are used to repeatedly click Web advertising links, generating per-click revenue for the attacker. This represents fraud because only the clicks of human users with a legitimate interest are valuable to advertisers. The bots will not buy the product or service as a result of clicking the advertisement.

- These illegal activities are extremely profitable. For example, a 2006 study by the Germany Honeynet Project estimated that a botmaster can make about $430 per day just from per-install advertising software. A 20-year-old California botmaster indicted in February 2006 earned $100,000 in advertising revenue from his botnet operations.

However, both of these cases pale in comparison to the estimated $20 million worth of damage caused by an international ring of computer criminals known as the A-Team.

- Due to these very profitable uses of botnets, many botmasters make money simply by creating botnets and then renting out processing power and bandwidth to spammers, extortionists, and identity thieves. Despite a recent string of high-profile botnet arrests, these are merely a drop in the bucket. Overall, botmasters still have a fairly low chance of getting caught due to a lack of effective traceback techniques. The relatively low risk combined with high yield makes the botnet business very appealing as a fundraising method for criminal enterprises, especially in countries with weak computer crime enforcement.

2.5 Botnet Defense [33-44]

- When botnets emerged, the response was similar to previous Internet malware: Antivirus vendors created signatures and removal techniques for each new instance of the bot. This approach initially worked well at the host level, but researchers soon started exploring more advanced methods for eliminating more than one bot at a time. After all, a botnet with tens of thousands of members would be very tedious to combat one bot at a time.
- This section describes the current defenses against centralized botnets, moving from the host level to the network level, then to the C&C server, and finally to the botmaster himself.

2.5.1 Detecting and Removing Individual Bots

- Removing individual bots does not usually have a noticeable impact on the overall botnet, but it is a crucial first step in botnet defense. The basic antivirus approach using signature-based detection is still effective with many bots, but some are starting to use

polymorphism, which creates unique instances of the bot code and evades signature-based detection.

- For example, Agobot is known to have thousands of variants, and it includes built-in support for polymorphism to change its signature at will.
- To deal with these more sophisticated bots and all other polymorphic malware, detection must be done using behavioral analysis and heuristics. Researchers Stinson and Mitchell have developed a taint-based approach called BotSwat that marks all data originating from the network. If this data is used as input for a system call, there is a high probability that it is bot-related behavior, since user input typically comes from the keyboard or mouse on most end-user systems.

2.5.2 Detecting C&C Traffic

To mitigate the botnet problem on a larger scale, researchers turned their attention to network-based detection of the botnet's C&C traffic. This method allows organizations or even ISPs to detect the presence of bots on their entire network, rather than having to check each machine individually. One approach is to examine network traffic for certain known patterns that occur in botnet C&C traffic. This is, in effect, a network-deployed version of signature-based detection, where signatures have to be collected for each bot before detection is possible. Researchers Goebel and Holz implemented this method in their Rishi tool, which evaluates IRC nicknames for likely botnet membership based on a list of known botnet naming schemes. As with all signature-based approaches, it often leads to an "arms race" where the attackers frequently change their malware and the network security community tries to keep up by creating signatures for each new instance.

Rather than relying on a limited set of signatures, it is also possible to use the IDS technique of anomaly detection to identify unencrypted IRC botnet traffic. This method was successfully

implemented by researchers Binkley and Singh at Portland State University, and as a result they reported a significant increase in bot detection on the university network.

Another IDS-based detection technique called BotHunter was proposed by Gu et al. in 2007. Their approach is based on IDS dialog correlation techniques: It deploys three separate network monitors at the network perimeter, each detecting a specific stage of bot infection. By correlating these events, BotHunter can reconstruct the traffic dialog between the infected machine and the outside Internet. From this dialog, the engine determines whether a bot infection has taken place with a high accuracy rate. Moving beyond the scope of a single network/organization, traffic from centralized botnets can be detected at the ISP level based only on transport layer flow statistics. This approach was developed by Karasaridis et al. and solves many of the problems of packet-level inspection. It is passive, highly scalable, and only uses flow summary data (limiting privacy issues). Additionally, it can determine the size of a botnet without joining and can even detect botnets using encrypted C&C. The approach exploits the underlying principle of centralized botnets: Each bot has to contact the C&C server, producing detectable patterns in network traffic flows.

Beyond the ISP level, a heuristic method for Internet-wide bot detection was proposed by Ramachandran et al. in 2006. In this scheme, query patterns of DNS black-hole lists (DNSBLs) are used to create a list of possible bot-infected IP addresses. It relies on the fact that botmasters need to periodically check whether their spam-sending bots have been added to a DNSBL and have therefore become useless. The query patterns of botmasters to a DNSBL are very different from those of legitimate mail servers, allowing detection. One major limitation is that this approach focuses mainly on the sending of spam. It would most likely not detect bots engaged in other illegal activities, such as DDoS attacks or click fraud, since these do not require DNSBL lookups.

2.5.3 Detecting and Neutralizing the C&C Servers

Though detecting C&C traffic and eliminating all bots on a given local network is a step in the right direction, it still doesn't allow the takedown of an entire botnet at once.

To achieve this goal in a centralized botnet, access to the C&C servers must be removed. This approach assumes that the C&C servers consist of only a few hosts that are accessed directly. If hundreds or thousands of hosts are used in a fast-flux proxy configuration, it becomes extremely challenging to locate and neutralize the true C&C servers.

In work similar to BotHunter, researchers Gu et al. developed BotSniffer in 2008. This approach represents several improvements, notably that BotSniffer can handle encrypted traffic, since it no longer relies only on content inspection to correlate messages. A major advantage of this approach is that it requires no advance knowledge of the bot's signature or the identity of C&C servers. By analyzing network traces, BotSniffer detects the spatial-temporal correlation among C&C traffic belonging to the same botnet. It can therefore detect both the bot members and the C&C server(s) with a low false positive rate.

Most of the approaches mentioned under "Detecting C&C Traffic" can also be used to detect the C&C servers, with the exception of the DNSBL approach. However, their focus is mainly on detection and removal of individual bots. None of these approaches mentions targeting the C&C servers to eliminate an entire botnet.

One of the few projects that has explored the feasibility of C&C server takedown is the work of Freiling et al. in 2005. Although their focus is on DDoS prevention, they describe the method that is generally used in the wild to remove C&C servers when they are detected. First, the bot binary is either reverse-engineered or run in a sandbox to observe its behavior, specifically the hostnames of the C&C servers. Using this information, the proper dynamic DNS providers can be notified to remove the DNS entries for the C&C servers, preventing any bots from contacting them and thus severing contact between the botmaster and his botnet. Dagon et al. used a similar

approach in 2006 to obtain experiment data for modeling botnet propagation, redirecting the victim's connections from the true C&C server to their sinkhole host. Even though effective, the manual analysis and contact with the DNS operator is a slow process. It can take up to several days until all C&C servers are located and neutralized. However, this process is essentially the best available approach for shutting down entire botnets in the wild. As we mentioned, this technique becomes much harder when fast-flux proxies are used to conceal the real C&C servers or a P2P topology is in place.

2.5.4 Attacking Encrypted C&C Channels

Though some of the approaches can detect encrypted C&C traffic, the presence of encryption makes botnet research and analysis much harder. The first step in dealing with these advanced botnets is to penetrate the encryption that protects the C&C channels. A popular approach for adding encryption to an existing protocol is to run it on top of SSL/ TLS; to secure HTTP traffic, ecommerce Web sites run HTTP over SSL/TLS, known as HTTPS. Many encryption schemes that support key exchange (including SSL/TLS) are susceptible to man-in-the-middle (MITM) attacks, whereby a third party can impersonate the other two parties to each other. Such an attack is possible only when no authentication takes place prior to the key exchange, but this is a surprisingly common occurrence due to poor configuration.

The premise of an MITM attack is that the client does not verify that it's talking to the real server, and vice versa. When the MITM receives a connection from the client, it immediately creates a separate connection to the server (under a different encryption key) and passes on the client's request. When the server responds, the MITM decrypts the response, logs and possibly alters the content, then passes it on to the client re-encrypted with the proper key. Neither the client nor the server notice that anything is wrong, because they are communicating with each other over an encrypted connection, as expected.

The important difference is that unknown to either party, the traffic is being decrypted and re-encrypted by the MITM in transit, allowing him to observe and alter the traffic. In the context of bots, two main attacks on encrypted C&C channels are possible:

(1) “gray-box” analysis, whereby the bot communicates with a local machine impersonating the C&C server, and a full MITM attack, in which the bot communicates with the true C&C server. Figure 2.2 shows a possible setup for both attacks, using the Delegate proxy for the conversion to and from SSL/TLS.

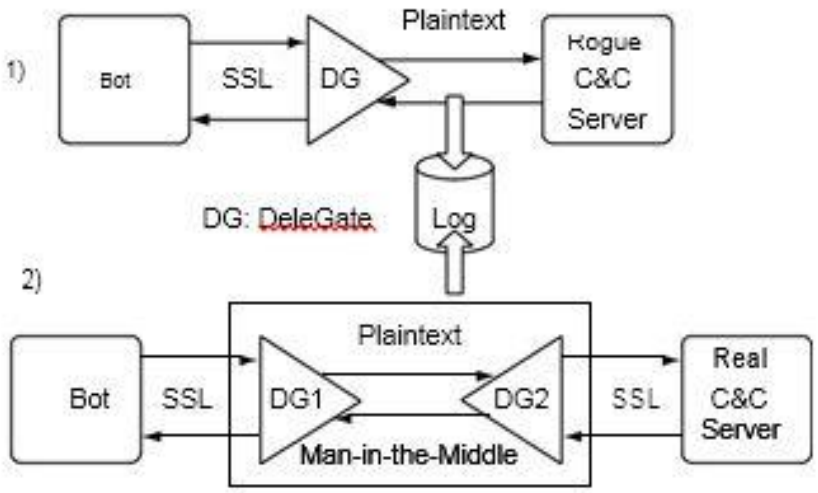


Figure 2.2: Setups for man-in-the-middle attacks on encrypted C&C channels

The first attack is valuable to determine the authentication information required to join the live botnet: the address of the C&C server, the IRC channel name (if applicable), plus any required passwords. However, it does not allow the observer to see the interaction with the larger botnet, specifically the botmaster. The second attack reveals the full interaction with the botnet, including all botmaster commands, the botmaster password used to control the bots, and possibly the IP addresses of other bot members (depending on the configuration of the C&C server). Figures 2.3–2.5 show the screenshots of the full MITM attack on a copy of Agobot configured to connect to its C&C server via SSL/TLS. Specifically, Figure 2.3 shows the boaster’s IRC

window, with his commands and the bot's responses. Figure 2.4 shows the encrypted SSL/TLS trace, and Figure 2.5 shows the decrypted plaintext that was observed at the Delegate proxy. The botmaster password botmaster PASS is clearly visible, along with the required username, botmaster.

Armed with the botmaster username and password, the observer could literally take over the botnet. He could log in as the botmaster and then issue a command such as Agobot's .bot. remove, causing all bots to disconnect from the botnet and permanently remove themselves from the infected computers. Unfortunately, there are legal issues with this approach because it constitutes unauthorized access to all the botnet computers, despite the fact that it is in fact a benign command to remove the bot software.

2.5.5 Locating and Identifying the Botmaster

Shutting down an entire botnet at once is a significant achievement, especially when the botnet numbers in the tens of thousands of members. However, there is nothing stopping the botmaster from simply deploying new bots to infect the millions of vulnerable hosts on the Internet, creating a new botnet in a matter of hours. In fact, most of the machines belonging to the shut-down botnet are likely to become infected again because the vulnerabilities and any attacker-installed backdoors often remain active, despite the elimination of the C&C servers.

Botnet-hunting expert Gadi Evron agrees: "When we disable a command-and-control server, the botnet is immediately recreated on another host. We're not hurting them anymore," he said in a 2006 interview. The only permanent solution of the botnet problem is to go after the root cause: the botmasters. Unfortunately, most botmasters are very good at concealing their identities and

locations, since their livelihood depends on it. Tracking the botmaster to her true physical location is a complex problem that is described in detail in the next section. So far, there is no published work that would allow automated botmaster traceback on the Internet, and it remains an open problem.

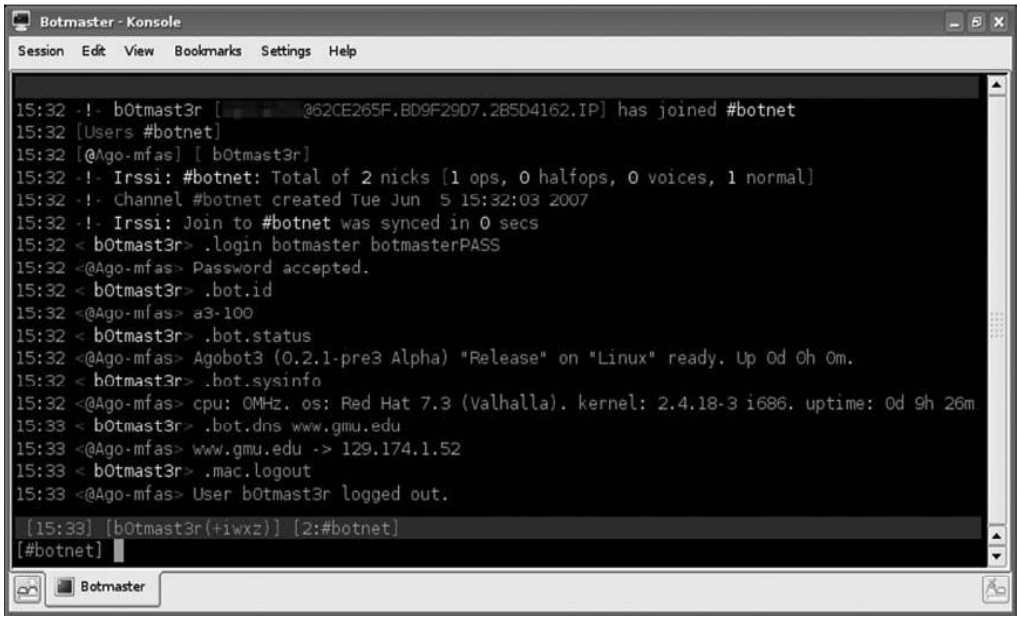


Figure 2.3: Screenshot showing the botmaster's IRC window

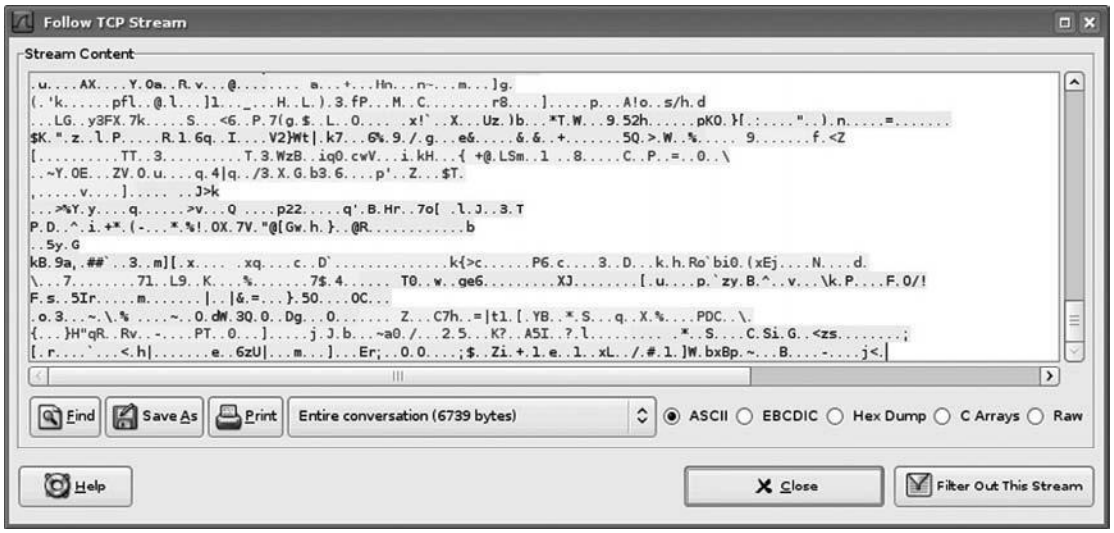


Figure 2.4: Screenshot showing the SSL/TLS-encrypted network traffic

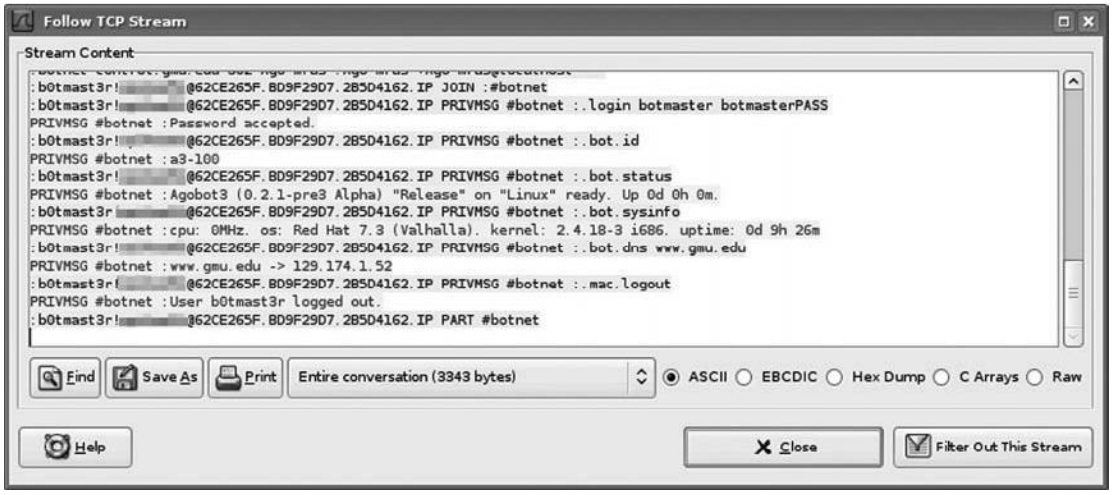


Figure 2.5: Screenshot showing decrypted plaintext from the Delegate proxy

2.6 Botmaster Traceback [22, 31-32, 45-56]

The botnet field is full of challenging problems: obfuscated binaries, encrypted C&C channels, fast-flux proxies protecting central C&C servers, customized communication protocols, and many more (see Figure 2.6). Arguably the most challenging task is locating the botmaster. Most botmasters take precautions on multiple levels to ensure that their connections cannot be traced to their true locations.

The reason for the botmaster's extreme caution is that a successful trace would have disastrous consequences. He could be arrested, his computer equipment could be seized and scrutinized in detail, and he could be sentenced to an extended prison term. Additionally, authorities would likely learn the identities of his associates, either from questioning him or by searching his computers. As a result, he would never again be able to operate in the Internet underground and could even face violent revenge from his former associates when he is released.

In the United States, authorities have recently started to actively pursue botmasters, resulting in several arrests and convictions. In November 2005, 20-year-old Jeanson James Ancheta of

California was charged with botnet-related computer offenses. He pleaded guilty in January 2006 and could face up to 25 years in prison. In a similar case, 20-year-old

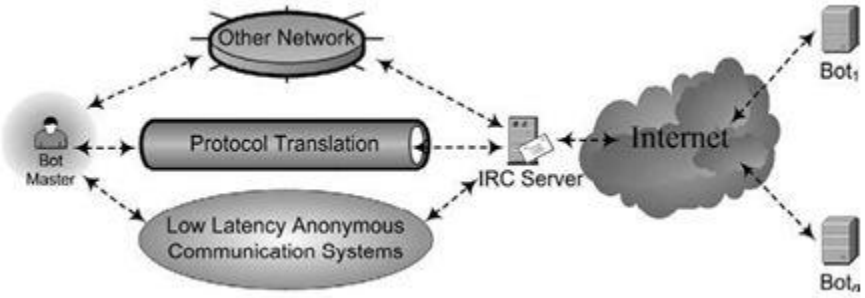


Figure 2.6: Botnet C&C traffic laundering

Christopher Maxwell was indicted on federal computer charges. He is accused of using his botnet to attack computers at several universities and a Seattle hospital, where bot infections severely disrupted operations.

In particular, the FBI's Operation Bot Roast has resulted in several high-profile arrests, both in the United States and abroad. The biggest success was the arrest of 18-year-old New Zealand native Owen Thor Walker, who was a member of a large international computer crime ring known as the A-Team. This group is reported to have infected up to 1.3 million computers with bot software and caused about $20 million in economic damage. Despite this success, Walker was only a minor player, and the criminals in control of the A-Team are still at large. Unfortunately, botmaster arrests are not very common. The cases described here represent only several individuals; thousands of botmasters around the world are still operating with impunity. They use sophisticated techniques to hide their true identities and locations, and they often operate in countries with weak computer crime enforcement. The lack of international coordination, both on the Internet and in law enforcement, makes it hard to trace botmasters and even harder to hold them accountable to the law.

2.6.1 Traceback Challenges

One defining characteristic of the botmaster is that he originates the botnet C&C traffic. Therefore, one way to find the botmaster is to track the botnet C&C traffic. To hide himself, the botmaster wants to disguise his link to the C&C traffic via various traffic-laundering techniques that make tracking C&C traffic more difficult. For example, a botmaster can route his C&C traffic through a number of intermediate hosts, various protocols, and low-latency anonymous networks to make it extremely difficult to trace. To further conceal his activities, a botmaster can also encrypt his traffic to and from the C&C servers. Finally, a botmaster only needs to be online briefly and send small amounts of traffic to interact with his botnet, reducing the chances of live traceback. Figure 2.6 illustrates some of the C&C traffic-laundering techniques a botmaster can use.

The intermediate hosts used for traffic laundering are known as stepping stones. The attacker sets them up in a chain, leading from the botmaster's true location to the C&C server. Stepping stones can be SSH servers, proxies (such as SOCKS), IRC bouncers (BNCs), virtual private network (VPN) servers, or any number of network redirection services. They usually run on compromised hosts, which are under the attacker's control and lack audit/logging mechanisms to trace traffic. As a result, manual traceback is tedious and time-consuming, requiring the cooperation of dozens of organizations whose networks might be involved in the trace.

The major challenge posed by stepping stones is that all routing information from the previous hop (IP headers, TCP headers, and the like) is stripped from the data before it is sent out on a new, separate connection. Only the content of the packet (the application layer data) is preserved, which renders many existing tracing schemes useless. An example of a technique that relies on routing header information is probabilistic packet marking. This approach was introduced by Savage et al. in 2000, embedding tracing information in an unused IP header field. Two years later, Goodrich expanded this approach, introducing "randomize-and-link" for better scalability. Another technique for IP-level traceback is the log/hash-based scheme introduced by Snoeren et

al. and enhanced by Li et al. These techniques were very useful in combating the fast-spreading worms of the early 2000s, which did not use stepping stones. However, these approaches do not work when stepping stones are present, since IP header information is lost.

- **Multiple Protocols**

Another effective and efficient method to disguise the botmaster is to launder the botnet C&C traffic across other protocols. Such protocol laundering can be achieved by either protocol tunneling or protocol translation. For example, a sophisticated botmaster could route its command and control traffic through SSH (or even HTTP) tunnels to reach the command and control center. The botmaster could also use some intermediate host X as a stepping stone, use some real-time communication protocols other than IRC between the botmaster host and host X, and use IRC between the host X and the IRC server. In this case, host X performs the protocol translation at the application layer and serves as a conduit of the botnet C&C channel. One protocol that is particularly suitable for laundering the botnet command and control is instant messaging (IM), which supports real-time text-based communication between two or more people.

- **Low-Latency Anonymous Network**

Besides laundering the botnet C&C across stepping stones and different protocols, a sophisticated botmaster could anonymize its C&C traffic by routing it through some low-latency anonymous communication systems. For example, Tor—the second generation of onion routing—uses an overlay network of onion routers to provide anonymous outgoing connections and anonymous hidden services. The botmaster could use Tor as a virtual tunnel to anonymize his TCP-based C&C traffic to the IRC server of the botnet. At the same time, the IRC server of the botnet could utilize Tor's hidden services to anonymize the IRC server of the botnet in such a way that its network location is unknown to the bots and yet it could communicate with all the bots.

- **Encryption**

All or part of the stepping stone chain can be encrypted to protect it against content inspection, which could reveal information about the botnet and botmaster. This can be done using a number of methods, including SSH tunneling, SSL/TLS-enabled BNCs, and IPsec tunneling. Using encryption defeats all content-based tracing approaches, so the tracer must rely on other network flow characteristics, such as packet size or timing, to correlate flows to each other.

- **Low-Traffic Volume**

Since the botmaster only has to connect briefly to issue commands and retrieve results from his botnet, a low volume of traffic flows from any given bot to the botmaster. During a typical session, only a few dozen packets from each bot can be sent to the botmaster. Tracing approaches that rely on analysis of packet size or timing will most likely be ineffective because they typically require a large amount of traffic (several hundred packets) to correlate flows with high statistical confidence. Examples of such tracing approaches all use timing information to embed a traceable watermark. These approaches can handle stepping stones, encryption, and even low-latency anonymizing network, but they cannot be directly used for botmaster traceback due to the low traffic volume.

2.6.2 Traceback Beyond the Internet

Even if all three technical challenges can be solved and even if all Internet-connected organizations worldwide cooperate to monitor traffic, there are additional traceback challenges beyond the reach of the Internet (see Figure 2.7). Any IP-based traceback method assumes that the true source IP belongs to the computer the attacker is using and that this machine can be physically located. However, in many scenarios this is not true—for example, (1) Internet-connected mobile phone networks, (2) open wireless (Wi-Fi) networks, and (3) public computers, such as those at libraries and Internet cafe´s.

Most modern cell phones support text-messaging services such as Short Message Service (SMS), and many smart phones also have full-featured IM software. As a result, the botmaster can use a mobile device to control her botnet from any location with cell phone reception. To enable her cell phone to communicate with the C&C server, a botmaster needs to use a protocol translation service or a special IRC client for mobile phones. She can run the translation service on a compromised host, an additional stepping stone. For an IRC botnet, such a service would receive the incoming SMS or IM message, then repackage it as an IRC message and send it on to the C&C server (possibly via more stepping stones), as shown in Figure 2.7 . To eliminate the need for protocol translation, the botmaster can run a native IRC client on a smart phone with Internet access. Examples of such clients are the Java-based WLIrc and jmIrc open source projects. In Figure 2.8, a Nokia smartphone is shown running MSN Messenger, controlling an Agobot zombie via MSN-IRC protocol translation. On the screen, a new bot has just been infected and has joined the IRC channel following the botmaster's .scan.dcom command.

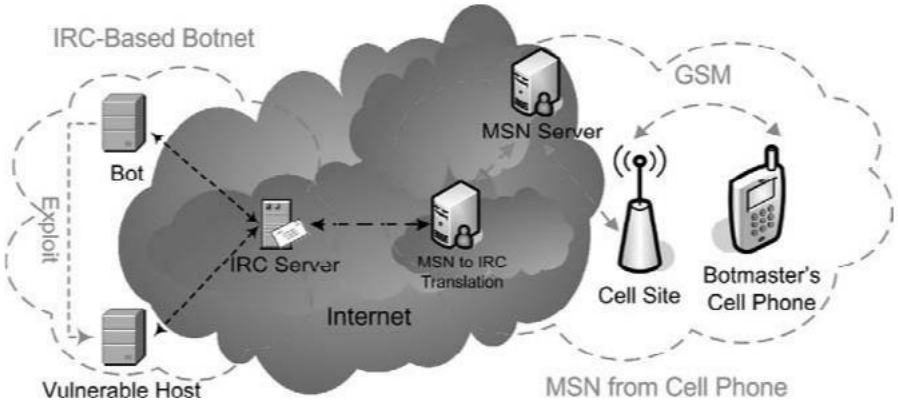


Figure 2.7: Using a cell phone to evade Internet-based traceback

When a botnet is being controlled from a mobile device, even a perfect IP traceback solution would only reach as far as the gateway host that bridges the Internet and the carrier's mobile network. From there, the tracer can ask the carrier to complete the trace and disclose the name and even the current location of the cell phone's owner. However, there are several problems

with this approach. First, this part of the trace again requires lots of manual work and cooperation of yet another organization, introducing further delays and making a real-time trace unlikely. Second, the carrier won't be able to determine the name of the subscriber if he is using a prepaid cell phone. Third, the tracer could obtain an approximate physical location based on cell site triangulation. Even if he can do this in real time, it might not be very useful if the botmaster is in a crowded public place. Short of detaining all people in the area and checking their cell phones, police won't be able to pinpoint the botmaster.

A similar situation arises when the botmaster uses an unsecured Wi-Fi connection. This could either be a public access point or a poorly configured one that is intended to be private. With a strong antenna, the botmaster can be located up to several thousand feet away. In a typical downtown area, such a radius can contain thousands of people and just as many computers. Again, short of searching everyone in the vicinity, the police will be unable to find the botmaster.

Finally, many places provide public Internet access without any logging of the users' identities. Prime examples are public libraries, Internet cafe´s, and even the business centers at most hotels. In this scenario, a real-time trace would actually find the botmaster, since he would be sitting at the machine in question. However, even if the police are late by only several minutes, there might no longer be any record of who last used the computer. Physical evidence such as fingerprints, hair, and skin cells would be of little use, since many people use these computers each day. Unless a camera system is in place and it captured a clear picture of the suspect on his way to/from the computer, the police again will have no leads.

This section illustrates a few common scenarios where even a perfect IP traceback solution would fail to locate the botmaster. Clearly, much work remains on developing automated, integrated traceback solutions that work across various types of networks and protocols.

Figure 2.8: Using a Nokia smartphone to control an Agobot-based botnet.

2.7 Summary

Botnets are one of the biggest threats to the Internet today, and they are linked to most forms of Internet crime. Most spam, DDoS attacks, spyware, click fraud, and other attacks originate from botnets and the shadowy organizations behind them. Running a botnet is immensely profitable, as several recent high-profile arrests have shown. Currently, many botnets still rely on a centralized IRC C&C structure, but more and more botmasters are using P2P protocols to provide resilience and avoid a single point of failure. A recent large-scale example of a P2P botnet is the Storm Worm, widely covered in the media.

A number of botnet countermeasures exist, but most are focused on bot detection and removal at the host and network level. Some approaches exist for Internet-wide detection and disruption of entire botnets, but we still lack effective techniques for combating the root of the problem: the botmasters who conceal their identities and locations behind chains of stepping-stone proxies.

The three biggest challenges in botmaster traceback are stepping stones, encryption, and the low traffic volume. Even if these problems can be solved with a technical solution, the trace must be

able to continue beyond the reach of the Internet. Mobile phone networks, open wireless access points, and public computers all provide an additional layer of anonymity for the botmasters.

Short of a perfect solution, even a partial traceback technique could serve as a very effective deterrent for botmasters. With each botmaster that is located and arrested, many botnets will be eliminated at once. Additionally, other botmasters could decide that the risks outweigh the benefits when they see more and more of their colleagues getting caught. Currently, the economic equation is very simple: Botnets can generate large profits with relatively low risk of getting caught. A botmaster traceback solution, even if imperfect, would drastically change this equation and convince more botmasters that it simply is not worth the risk of spending the next 10–20 years in prison.

2.8 References

[1] Holz T. A short visit to the bot zoo. IEEE Security and Privacy 2005;3(3):76–9.

[2] Berinato S. Attack of the bots, WIRED. Issue 14.11, November 2006, www.wired.com/wired/archive/14.11/ botnet.html.

[3] Evers J. 'Bot herders' may have controlled 1.5 million PCs. http://news.cnet.com/Bot-herders-may-have-controlled-1.5-million-PCs/2100-7350_3-5906896.html.

[4] Greenberg A. Spam crackdown 'a drop in the bucket'. Forbes June 14, 2007, www.forbes.com/security/2007/06/14/spam-arrest-fbi-tech-security cx_ag_0614spam.html.

[5] Wikipedia contributors. Timeline of notable computer viruses and worms. http://en.wikipedia.org/w/index.php?title¼Timeline_of_notable_computer_viruses_and_worms &oldid=207972502 (accessed May 3, 2008).

[6] Barford P, Yegneswaran V. "An inside look at botnets," Special Workshop on Malware Detection, Advances in Information Security. Springer Verlag, 2006.

[7] Wikipedia contributors. Eggdrop. http://en.wikipedia.org/w/index.php?title=Eggdrop&oldid¼207430332 (accessed May 3, 2008).

[8] Cooke E, Jahanian F, McPherson D. The zombie roundup: Understanding, detecting, and disturbing botnets, In: Proc. 1st Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI), Cambridge; July 7, 2005. p. 39–44.

[9] Ianelli N, Hackworth A. Botnets as a vehicle for online crime. In: Proc. 18th Annual Forum of Incident Response and Security Teams (FIRST), Baltimore; June 25–30, 2006.

[10] Rajab M, Zarfoss J, Monrose F, Terzis A. A multifaceted approach to understanding the botnet phenomenon. In: Proc. of the 6th ACM SIGCOM Internet Measurement Conference, Brazil: Rio de Janeiro; October 2006.

[11] Trend Micro. Taxonomy of botnet threats. Trend Micro Enterprise Security Library; November 2006.

[12] Symantec. Symantec internet security threat report, trends for July–December 2007. Volume XIII, April 2008.

[13] Grizzard J, Sharma V, Nunnery C, Kang B, Dagon D. Peer-to-peer botnets: Overview and case study. In: Proc. First Workshop on Hot Topics in Understanding Botnets (HotBots), Cambridge, April 2007.

[14] Stewart J. Bobax Trojan analysis, SecureWorks May 17, 2004, http://secureworks.com/research/threats/ bobax.

[15] Chiang K, Lloyd L. A case study of the Rustock Rootkit and Spam Bot. In: Proc. First Workshop on Hot Topics in Understanding Botnets (HotBots), Cambridge, April 10, 2007.

[16] Lemos R. Bot software looks to improve peerage, SecurityFocus. May 2, 2006, www.securityfocus.com/ news/11390/.

[17] Wang P, Sparks S, Zou C. An advanced hybrid peer-to-peer botnet. In: Proc. First Workshop on Hot Topics in Understanding Botnets (HotBots), Cambridge, April 10, 2007.

[18] Stewart J. Sinit P2P Trojan analysis. SecureWorks. December 8, 2004, www.secureworks.com/research/ threats/sinit/.

[19] Schoof R, Koning R. Detecting peer-to-peer botnets. unpublished paper, University of Amsterdam, February 4, 2007, http://staff.science.uva.nl/ delaat/sne-2006-2007/p17/report.pdf.

[20] Wikipedia contributors. Storm worm. http://en.wikipedia.org/w/index.php?title¼Storm_Worm&oldid= 207916428 (accessed May 4, 2008).

[21] Bizeul D. Russian business network study. unpublished paper, November 20, 2007, www.bizeul.org/files/ RBN_study.pdf.

[22] Cha AE. Internet dreams turn to crime, Washington Post May 18, 2003, www.washingtonpost.com/ac2/wp-dyn/A2619-2003May17.

[23] Koerner BI. From Russia with l pht, Legal Affairs May–June 2002, http://legalaffairs.org/issues/May-June-2002/feature_koerner_mayjun2002.msp.

[24] Delio M. Inside Russia's hacking culture. WIRED. March 12, 2001, www.wired.com/culture/lifestyle/news/2001/03/42346.

[25] Wikipedia contributors. Russian business network. http://en.wikipedia.org/w/index.php?title=Russian_ Business_Network&oldid=209665215 (accessed May 3, 2008).

[26] Tung L. Infamous Russian ISP behind Bank of India hack. ZDNet. September 4, 2007, http://news.zdnet.co. uk/security/0,1000000189,39289057,00.htm? r=2.

[27] Ba¨cher P, Holz T, Ko¨tter M, Wicherski G. Know your enemy: Tracking botnets. March 13, 2005, see www. honeynet.org/papers/bots/.

[28] Wikipedia contributors. E-mail spam, http://en.wikipedia.org/w/index.php?title=E-mail_spam&oldid= 209902571 (accessed May 3, 2008).

[29] Wikipedia contributors. Pharming. http://en.wikipedia.org/w/index.php?title¼Pharming&oldid=196469141accessed May 3, 2008.

[30] Naraine R. Money bots: Hackers cash in on hijacked PCs. eWeek. September 8, 2006, www.eweek.com/article2/0,1759,2013924,00.asp.

[31] Roberts PF. DOJ indicts hacker for hospital botnet attack. eWeek. February 10, 2006, www.eweek.com/article2/0,1759,1925456,00.asp.

[32] Claburn T. New Zealander 'AKILL' pleads guilty to botnet charges. Information Week April 3, 2008, www.informationweek.com/news/security/cybercrime/showArticle.jhtml?articleID=207001573.

[33] Wikipedia contributors. Agobot (computer worm). http://en.wikipedia.org/w/index.php?title=Agobot_%28computer_worm%29&oldid=201957526 (accessed May 3, 2008).

[34] Stinson E, Mitchell J. Characterizing bots' remote control behavior. In: Proc. 4th International Conference on Detection of Intrusions & Malware and Vulnerability Assessment (DIMVA), Lucerne, Switzerland, July 12–13, 2007.

[35] Goebel J, Holz T. Rishi: Identify bot contaminated hosts by IRC nickname evaluation. In: Proc. First
Workshop on Hot Topics in Understanding Botnets (HotBots), Cambridge, April 10, 2007.

[36] Binkley J, Singh S. An algorithm for anomaly-based botnet detection, In: Proc. 2nd Workshop on Steps to
Reducing Unwanted Traffic on the Internet (SRUTI), San Jose, July 7, 2006. p. 43–8.

[37] Gu G, Porras P, Yegneswaran V, Fong M, Lee W. BotHunter: Detecting malware infection through IDSdriven dialog correlation. In: Proc. 16th USENIX Security Symposium, Boston; August, 2007.

[38] Karasaridis A, Rexroad B, Hoeflin D. Wide-scale botnet detection and characterization, In: Proc. First
Workshop on Hot Topics in Understanding Botnets (HotBots), Cambridge, MA; April 10, 2007.

[39] Ramachandran A, Feamster N, Dagon D. Revealing botnet membership using DNSBL counter-intelligence, In: Proc. 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI), San Jose, CA; July 7, 2006. p. 49–54.

[40] Gu G, Zhang J, Lee W. BotSniffer: Detecting botnet command and control channels in network traffic, In: Proc. 15th Network and Distributed System Security Symposium (NDSS), San Diego, February 2008.

[41] Freiling F, Holz T, Wicherski G. Botnet tracking: Exploring a root-cause methodology to prevent denial-ofservice attacks. In: Proc. 10th European Symposium on Research in Computer Security (ESORICS), Milan, Italy, September 12–14, 2005.

[42] Dagon D, Zou C, Lee W. Modeling botnet propagation using time zones, In: Proc. 13th Network and Distributed System Security Symposium (NDSS), February 2006.

[43] DeleGate multi-purpose application gateway. www.delegate.org/delegate/ (accessed May 4, 2008).

[44] Naraine R. Is the botnet battle already lost? eWeek. October 16, 2006, www.eweek.com/article2/0,1895,2029720,00.asp.

[45] Roberts PF. California man charged with botnet offenses. eWeek. November 3, 2005, www.eweek.com/article2/0,1759,1881621,00.asp.

[46] Roberts PF. Botnet operator pleads guilty. eWeek. January 24, 2006, www.eweek.com/article2/0,1759,1914833,00.asp.

[47] Nichols S. FBI 'bot roast' scores string of arrests. vnunet.com. December 3, 2007, www.vnunet.com/vnunet/news/2204829/bot-roast-scores-string-arrests.

[48] Savage S, Wetherall D, Karlin A, Anderson T. Practical network support for IP traceback, In: Proc. ACM SIGCOMM 2000, Sept. 2000. p. 295–306.

[49] Goodrich MT. Efficient packet marking for large-scale IP traceback, In: Proc. 9th ACM Conference on Computer and Communications Security (CCS 2002), October 2002. p. 117–26.

[50] Snoeren A, Patridge C, Sanchez LA, Jones CE, Tchakountio F, Kent ST, et al. Hash-based IP traceback. In: Proc. ACM SIGCOMM 2001, September 2001. p. 3–14.

CHAPTER 3

UNDERSTANDING CLOUD COMPUTING

3.1 Origins and Influences [1]

- **A Brief History**

The idea of computing in a "cloud" traces back to the origins of utility computing, a concept that computer scientist John McCarthy publicly proposed in 1961:

> *"If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility. ... The computer utility could become the basis of a new and important industry."*

In 1969, Leonard Kleinrock, a chief scientist of the Advanced Research Projects Agency Network or ARPANET project that seeded the Internet, stated:

> *"As of now, computer networks are still in their infancy, but as they grow up and become sophisticated, we will probably see the spread of 'computer utilities' ...".*

The general public has been leveraging forms of Internet-based computer utilities since the mid-1990s through various incarnations of search engines (Yahoo!, Google), e-mail services (Hotmail, Gmail), open publishing platforms (MySpace, Facebook, YouTube), and other types of social media (Twitter, LinkedIn). Though consumer-centric, these services popularized and validated core concepts that form the basis of modern-day cloud computing.

In the late 1990s, Salesforce.com pioneered the notion of bringing remotely provisioned services into the enterprise. In 2002, Amazon.com launched the Amazon Web Services (AWS) platform,

a suite of enterprise-oriented services that provide remotely provisioned storage, computing resources, and business functionality.

A slightly different evocation of the term "Network Cloud" or "Cloud" was introduced in the early 1990s throughout the networking industry. It referred to an abstraction layer derived in the delivery methods of data across heterogeneous public and semi-public networks that were primarily packet-switched, although cellular networks used the "Cloud" term as well. The networking method at this point supported the transmission of data from one end-point (local network) to the "Cloud" (wide area network) and then further decomposed to another intended end-point. This is relevant, as the networking industry still references the use of this term, and is considered an early adopter of the concepts that underlie utility computing.

It wasn't until 2006 that the term "cloud computing" emerged in the commercial arena. It was during this time that Amazon launched its Elastic Compute Cloud (EC2) services that enabled organizations to "lease" computing capacity and processing power to run their enterprise applications. Google Apps also began providing browser-based enterprise applications in the same year, and three years later, the Google App Engine became another historic milestone.

- **Definitions**

A Gartner report listing cloud computing at the top of its strategic technology areas further reaffirmed its prominence as an industry trend by announcing its formal definition as:

"…a style of computing in which scalable and elastic IT-enabled capabilities are delivered as a service to external customers using Internet technologies."

This is a slight revision of Gartner's original definition from 2008, in which "massively scalable" was used instead of "scalable and elastic." This acknowledges the importance of scalability in relation to the ability to scale vertically and not just to enormous proportions.

Forrester Research provided its own definition of cloud computing as:

"…a standardized IT capability (services, software, or infrastructure) delivered via Internet technologies in a pay-per-use, self-service way."

The definition that received industry-wide acceptance was composed by the National Institute of Standards and Technology (NIST). NIST published its original definition back in 2009, followed by a revised version after further review and industry input that was published in September of 2011:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models."

A more concise definition:

"Cloud computing is a specialized form of distributed computing that introduces utilization models for remotely provisioning scalable and measured resources."

This simplified definition is in line with all of the preceding definition variations that were put forth by other organizations within the cloud computing industry. The characteristics, service models, and deployment models referenced in the NIST definition.

- **Business Drivers**

Before delving into the layers of technologies that underlie clouds, the motivations that led to their creation by industry leaders must first be understood. Several of the primary business drivers that fostered modern cloud-based technology are presented in this section.

The origins and inspirations of many of the characteristics, models, and mechanisms covered throughout subsequent chapters can be traced back to the upcoming business drivers. It is important to note that these influences shaped clouds and the overall cloud computing market

from both ends. They have motivated organizations to adopt cloud computing in support of their business automation requirements. They have correspondingly motivated other organizations to become providers of cloud environments and cloud technology vendors in order to create and meet the demand to fulfill consumer needs.

- **Capacity Planning**

Capacity planning is the process of determining and fulfilling future demands of an organization's IT resources, products, and services. Within this context, *capacity* rep-resents the maximum amount of work that an IT resource is capable of delivering in a given period of time. A discrepancy between the capacity of an IT resource and its demand can result in a system becoming either inefficient (over-provisioning) or unable to fulfill user needs (under-provisioning). Capacity planning is focused on minimizing this discrepancy to achieve predictable efficiency and performance.

Different capacity planning strategies exist:

- *Lead Strategy* – adding capacity to an IT resource in anticipation of demand
- *Lag Strategy* – adding capacity when the IT resource reaches its full capacity
- *Match Strategy* – adding IT resource capacity in small increments, as demand increases

Planning for capacity can be challenging because it requires estimating usage load fluctuations. There is a constant need to balance peak usage requirements without unnecessary over-expenditure on infrastructure. An example is outfitting IT infrastructure to accommodate maximum usage loads which can impose unreasonable financial investments. In such cases, moderating investments can result in under-provisioning, leading to transaction losses and other usage limitations from lowered usage thresholds.

- **Cost Reduction**

A direct alignment between IT costs and business performance can be difficult to main-tain. The growth of IT environments often corresponds to the assessment of their maxi-mum usage

requirements. This can make the support of new and expanded business automations an ever-increasing investment. Much of this required investment is fun-neled into infrastructure expansion because the usage potential of a given automation solution will always be limited by the processing power of its underlying infrastructure.

Two costs need to be accounted for: the cost of acquiring new infrastructure, and the cost of its ongoing ownership. Operational overhead represents a considerable share of IT budgets, often exceeding up-front investment costs.

Common forms of infrastructure-related operating overhead include the following:

- technical personnel required to keep the environment operational
- upgrades and patches that introduce additional testing and deployment cycles
- utility bills and capital expense investments for power and cooling
- security and access control measures that need to be maintained and enforced to protect infrastructure resources
- administrative and accounts staff that may be required to keep track of licenses and support arrangements

The on-going ownership of internal technology infrastructure can encompass burdensome responsibilities that impose compound impacts on corporate budgets. An IT department can consequently become a significant and at times overwhelming drain on the business, potentially inhibiting its responsiveness, profitability, and overall evolution.

- **Organizational Agility**

Businesses need the ability to adapt and evolve to successfully face change caused by both internal and external factors. Organizational agility is the measure of an organization's responsiveness to change.

An IT enterprise often needs to respond to business change by scaling its IT resources beyond the scope of what was previously predicted or planned for. For example, infra-structure may be

subject to limitations that prevent the organization from responding to usage fluctuations—even when anticipated—if previous capacity planning efforts were restricted by inadequate budgets.

In other cases, changing business needs and priorities may require IT resources to be more available and reliable than before. Even if sufficient infrastructure is in place for an organization to support anticipated usage volumes, the nature of the usage may generate runtime exceptions that bring down hosting servers. Due to a lack of reliability controls within the infrastructure, responsiveness to consumer or customer requirements may be reduced to a point whereby a business' overall continuity is threatened.

On a broader scale, the up-front investments and infrastructure ownership costs that are required to enable new or expanded business automation solutions may themselves be prohibitive enough for a business to settle for IT infrastructure of less-than-ideal quality, thereby decreasing its ability to meet real-world requirements.

Worse yet, the business may decide against proceeding with an automation solution altogether upon review of its infrastructure budget, because it simply cannot afford to. This form of inability to respond can inhibit an organization from keeping up with market demands, competitive pressures, and its own strategic business goals.

- **Technology Innovations**

Established technologies are often used as inspiration and, at times, the actual foundations upon which new technology innovations are derived and built. This section briefly describes the pre-existing technologies considered to be the primary influences on cloud computing.

- **Clustering**

A cluster is a group of independent IT resources that are interconnected and work as a single system. System failure rates are reduced while availability and reliability are increased, since redundancy and failover features are inherent to the cluster.

A general prerequisite of hardware clustering is that its component systems have reasonably identical hardware and operating systems to provide similar performance levels when one failed component is to be replaced by another. Component devices that form a cluster are kept in synchronization through dedicated, high-speed communication links.

The basic concept of built-in redundancy and failover is core to cloud platforms. Clustering technology is explored further in Chapter 8 as part of the *Resource Cluster* mechanism description.

- **Grid Computing**

A computing grid (or "computational grid") provides a platform in which computing resources are organized into one or more logical pools. These pools are collectively coordinated to provide a high performance distributed grid, sometimes referred to as a "super virtual computer." Grid computing differs from clustering in that grid systems are much more loosely coupled and distributed. As a result, grid computing systems can involve computing resources that are heterogeneous and geographically dispersed, which is generally not possible with cluster computing-based systems.

Grid computing has been an on-going research area in computing science since the early 1990s. The technological advancements achieved by grid computing projects have influenced various aspects of cloud computing platforms and mechanisms, specifically in relation to common feature-sets such as networked access, resource pooling, and scalability and resiliency. These types of features can be established by both grid computing and cloud computing, in their own distinctive approaches.

For example, grid computing is based on a middleware layer that is deployed on computing resources. These IT resources participate in a grid pool that implements a series of workload distribution and coordination functions. This middle tier can contain load balancing logic, failover controls, and autonomic configuration management, each having previously inspired

similar and several more sophisticated cloud computing technologies. It is for this reason that some classify cloud computing as a descendant of earlier grid computing initiatives.

- **Virtualization**

Virtualization represents a technology platform used for the creation of virtual instances of IT resources. A layer of virtualization software allows physical IT resources to pro-vide multiple virtual images of themselves so that their underlying processing capabilities can be shared by multiple users.

Prior to the advent of virtualization technologies, software was limited to residing on and being coupled with static hardware environments. The virtualization process severs this software-hardware dependency, as hardware requirements can be simulated by emulation software running in virtualized environments.

Established virtualization technologies can be traced to several cloud characteristics and cloud computing mechanisms, having inspired many of their core features. As cloud computing evolved, a generation of *modern* virtualization technologies emerged to overcome the performance, reliability, and scalability limitations of traditional virtualization platforms.

Technology Innovations vs. Enabling Technologies

It is essential to highlight several other areas of technology that continue to contribute to modern-day cloud-based platforms. These are distinguished as *cloud-enabling technologies*:

- Broadband Networks and Internet Architecture
- Data Center Technology
- (Modern) Virtualization Technology
- Web Technology
- Multitenant Technology
- Service Technology

Each of these cloud-enabling technologies existed in some form prior to the formal advent of cloud computing. Some were refined further, and on occasion even redefined, as a result of the subsequent evolution of cloud computing.

3.2 Basic Concepts and Terminology[1]

This section establishes a set of basic terms that represent the fundamental concepts and aspects pertaining to the notion of a cloud and its most primitive artifacts.

- **Cloud**

A *cloud* refers to a distinct IT environment that is designed for the purpose of remotely provisioning scalable and measured IT resources. The term originated as a metaphor for the Internet which is, in essence, a network of networks providing remote access to a set of decentralized IT resources. Prior to cloud computing becoming its own formal-ized IT industry segment, the symbol of a cloud was commonly used to represent the Internet in a variety of specifications and mainstream documentation of Web-based architectures. This same symbol is now used to specifically represent the boundary of a cloud environment, as shown in Figure 3.1.

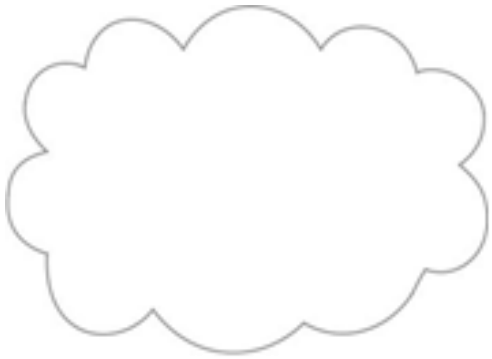


Figure 3.1 The symbol used to denote the boundary of a cloud environment

It is important to distinguish the term “cloud” and the cloud symbol from the Inter-net. As a specific environment used to remotely provision IT resources, a cloud has a finite boundary.

There are many individual clouds that are accessible via the Internet. Whereas the Internet provides open access to many Web-based IT resources, a cloud is typically privately owned and offers access to IT resources that is metered.

Much of the Internet is dedicated to the access of content-based IT resources published via the World Wide Web. IT resources provided by cloud environments, on the other hand, are dedicated to supplying back-end processing capabilities and user-based access to these capabilities. Another key distinction is that it is not necessary for clouds to be Web-based even if they are commonly based on Internet protocols and technologies. Protocols refer to standards and methods that allow computers to communicate with each other in a pre-defined and structured manner. A cloud can be based on the use of any protocols that allow for the remote access to its IT resources.

- **IT Resource**

An *IT resource* is a physical or virtual IT-related artifact that can be either software-based, such as a virtual server or a custom software program, or hardware-based, such as a physical server or a network device (Figure 3.2).

Figure 3.2 Examples of common IT resources and their corresponding symbols

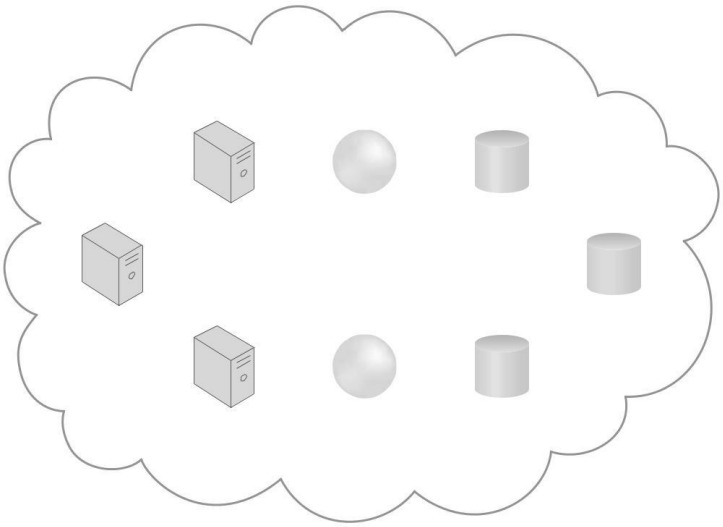


Figure 3.3 A cloud is hosting eight IT resources: three virtual servers, two cloud services, and three storage devices

Technology architectures and various interaction scenarios involving IT resources are illustrated in diagrams like the one shown in Figure 3.3. It is important to note the following points when studying and working with these diagrams:

- The IT resources shown within the boundary of a given cloud symbol usually do not represent all of the available IT resources hosted by that cloud. Subsets of IT resources are generally highlighted to demonstrate a particular topic.
- Focusing on the relevant aspects of a topic requires many of these diagrams to intentionally provide abstracted views of the underlying technology architectures. This means that only a portion of the actual technical details are shown.

Furthermore, some diagrams will display IT resources outside of the cloud symbol. This convention is used to indicate IT resources that are not cloud-based.

- **On-Premise**

As a distinct and remotely accessible environment, a cloud represents an option for the deployment of IT resources. An IT resource that is hosted in a conventional IT enterprise within an organizational boundary (that does not specifically represent a cloud) is considered to be located on the premises of the IT enterprise, or *on premise* for short. In other words, the term "on premise" is another way of stating "on the premises of a controlled IT environment that is not cloud-based." This term is used to qualify an IT resource as an alternative to "cloud-based." An IT resource that is on premise cannot be cloud-based, and vice-versa.

Note the following key points:

- An on premise IT resource can access and interact with a cloud-based IT resource.
- An on premise IT resource can be moved to a cloud, thereby changing it to a cloud-based IT resource.
- Redundant deployments of an IT resource can exist in both on premise and cloud-based environments.

If the distinction between on premise and cloud-based IT resources is confusing in relation to private clouds (described in the *Cloud Deployment Models* section of Chapter 4), then an alternative qualifier can be used.

- **Cloud Consumers and Cloud Providers**

The party that provides cloud-based IT resources is the *cloud provider*. The party that uses cloud-based IT resources is the *cloud consumer*. These terms represent roles usually assumed by organizations in relation to clouds and corresponding cloud provisioning contracts. These roles are formally defined in Chapter 4, as part of the *Roles and Boundaries* section.

- **Scaling**

Scaling, from an IT resource perspective, represents the ability of the IT resource to handle increased or decreased usage demands.

The following are types of scaling:

- *Horizontal Scaling* – scaling out and scaling in
- *Vertical Scaling* – scaling up and scaling down

The next two sections briefly describe each.

- ***Horizontal Scaling***

The allocating or releasing of IT resources that are of the same type is referred to as *horizontal scaling* (Figure 3.4). The horizontal allocation of resources is referred to as *scaling out* and the horizontal releasing of resources is referred to as *scaling in*. Horizontal scaling is a common form of scaling within cloud environments.

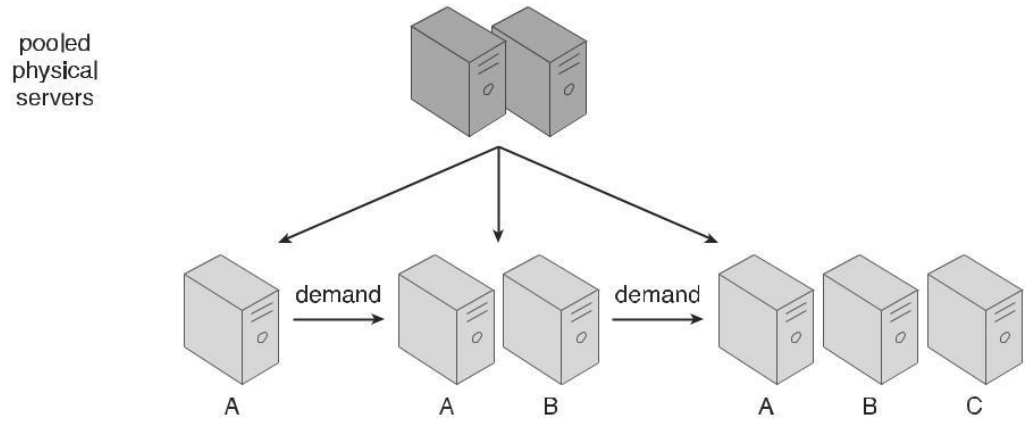


Figure 3.4 An IT resource (Virtual Server A) is scaled out by adding more of the same IT resources (Virtual Servers B and C).

- ***Vertical Scaling***

When an existing IT resource is replaced by another with higher or lower capacity, *vertical scaling* is considered to have occurred (Figure 3.5). Specifically, the replacing of an IT resource with another that has a higher capacity is referred to as *scaling up* and the replacing an IT resource with another that has a lower capacity is considered *scaling down*. Vertical scaling is

less common in cloud environments due to the downtime required while the replacement is taking place.

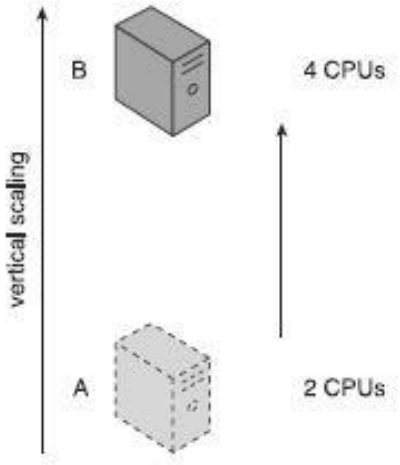


Figure 3.5 An IT resource (a virtual server with two CPUs) is scaled up by replacing it with a more powerful IT resource with increased capacity for data storage (a physical server with four CPUs)

Table 3.1 provides a brief overview of common pros and cons associated with horizontal and vertical scaling.

Table 3.1 A comparison of horizontal and vertical scaling

Horizontal Scaling	Vertical Scaling
less expensive	more expensive
(through commodity hardware components)	(specialized servers)
IT resources instantly available	IT resources normally instantly available
resource replication and automated scaling	additional setup is normally needed
additional IT resources needed	no additional IT resources needed
not limited by hardware capacity	limited by maximum hardware capacity

- **Cloud Service**

  Although a cloud is a remotely accessible environment, not all IT resources residing within a cloud can be made available for remote access. For example, a database or a physical server deployed within a cloud may only be accessible by other IT resources that are within the same cloud. A software program with a published API may be deployed specifically to enable access by remote clients.

A *cloud service* is any IT resource that is made remotely accessible via a cloud. Unlike other IT fields that fall under the service technology umbrella—such as service-oriented architecture—the term "service" within the context of cloud computing is especially broad. A cloud service can exist as a simple Web-based software program with a technical interface invoked via the use of a messaging protocol, or as a remote access point for administrative tools or larger environments and other IT resources.

In Figure 3.6, the yellow circle symbol is used to represent the cloud service as a simple Web-based software program. A different IT resource symbol may be used in the latter case, depending on the nature of the access that is provided by the cloud service.

A cloud service that exists as a virtual server is also being accessed from outside of the cloud's boundary (right). The cloud service on the left is likely being invoked by a consumer program that was designed to access the cloud service's published technical interface. The cloud service on the right may be accessed by a human user that has remotely logged on to the virtual server.

The driving motivation behind cloud computing is to provide IT resources as services that encapsulate other IT resources, while offering functions for clients to use and lever-age remotely. A multitude of models for generic types of cloud services have emerged, most of which are labeled with the "as-a-service" suffix.

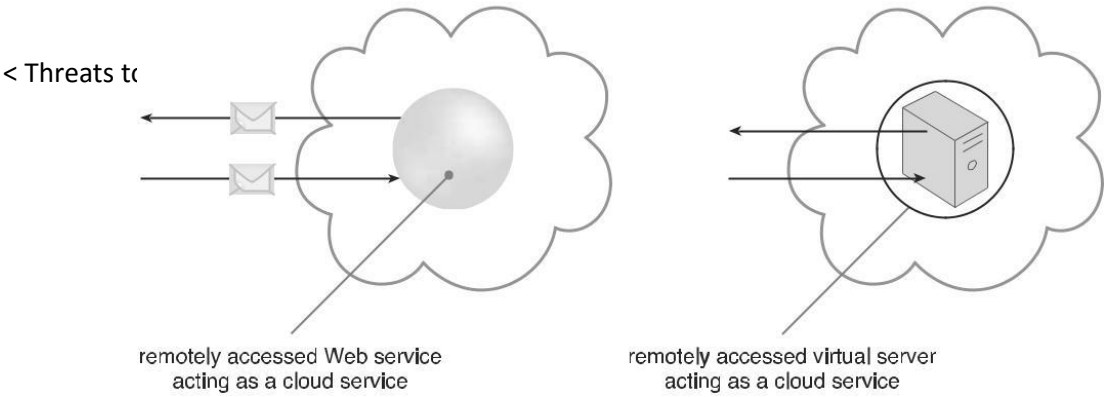


Figure 3.6 A cloud service with a published technical interface is being accessed by a consumer outside of the cloud (left).

- **Cloud Service Consumer**

The *cloud service consumer* is a temporary runtime role assumed by a software program when it accesses a cloud service.

As shown in Figure 3.7, common types of cloud service consumers can include software programs and services capable of remotely accessing cloud services with published service contracts, as well as workstations, laptops and mobile devices running software capable of remotely accessing other IT resources positioned as cloud services.

Figure 3.7 Examples of cloud service consumers. Depending on the nature of a given diagram, an artifact labeled as a cloud service consumer may be a software program or a hardware device (in which case it is implied that it is running a software program capable of acting as a cloud service consumer)

3.3 Goals and Benefits [1]

The common benefits associated with adopting cloud computing are explained in this section.

3.3.1 Reduced Investments and Proportional Costs

Similar to a product wholesaler that purchases goods in bulk for lower price points, pub-lic cloud providers base their business model on the mass-acquisition of IT resources that are then made available to cloud consumers via attractively priced leasing pack-ages. This opens the door for organizations to gain access to powerful infrastructure without having to purchase it themselves.

The most common economic rationale for investing in cloud-based IT resources is in the reduction or outright elimination of up-front IT investments, namely hardware and software purchases and ownership costs. A cloud's Measured Usage characteristic rep-resents a feature-set that allows measured operational expenditures (directly related to business performance) to replace anticipated capital expenditures. This is also referred to as *proportional costs*.

This elimination or minimization of up-front financial commitments allows enterprises to start small and accordingly increase IT resource allocation as required. Moreover, the reduction of up-front capital expenses allows for the capital to be redirected to the core business investment. In its most basic form, opportunities to decrease costs are derived from the deployment and operation of large-scale data centers by major cloud providers. Such data centers are commonly located in destinations where real estate, IT professionals, and network bandwidth can be obtained at lower costs, resulting in both capital and operational savings.

The same rationale applies to operating systems, middleware or platform software, and application software. Pooled IT resources are made available to and shared by multiple cloud consumers, resulting in increased or even maximum possible utilization. Operational costs and inefficiencies can be further reduced by applying proven practices and patterns for optimizing cloud architectures, their management, and their governance.

Common measurable benefits to cloud consumers include:

- On-demand access to pay-as-you-go computing resources on a short-term basis (such as processors by the hour), and the ability to release these computing resources when they are no longer needed.
- The perception of having unlimited computing resources that are available on demand, thereby reducing the need to prepare for provisioning.

The ability to add or remove IT resources at a fine-grained level, such as modifying available storage disk space by single gigabyte increments.

- Abstraction of the infrastructure so applications are not locked into devices or locations and can be easily moved if needed.

For example, a company with sizable batch-centric tasks can complete them as quickly as their application software can scale. Using 100 servers for one hour costs the same as using one server for 100 hours. This “elasticity” of IT resources, achieved without requiring steep initial investments to create a large-scale computing infrastructure, can be extremely compelling.

Despite the ease with which many identify the financial benefits of cloud computing, the actual economics can be complex to calculate and assess. The decision to proceed with a cloud computing adoption strategy will involve much more than a simple comparison between the cost of leasing and the cost of purchasing. For example, the financial benefits of dynamic scaling and the risk transference of both over-provisioning (under-utilization) and under-provisioning (over-utilization) must also be accounted for.

3.3.2 Increased Scalability

By providing pools of IT resources, along with tools and technologies designed to lever-age them collectively, clouds can instantly and dynamically allocate IT resources to cloud consumers, on-demand or via the cloud consumer’s direct configuration. This empowers cloud consumers to scale their cloud-based IT resources to accommodate processing fluctuations and

peaks automatically or manually. Similarly, cloud-based IT resources can be released (automatically or manually) as processing demands decrease.

A simple example of usage demand fluctuations throughout a 24 hour period is pro-vided in Figure 3.8

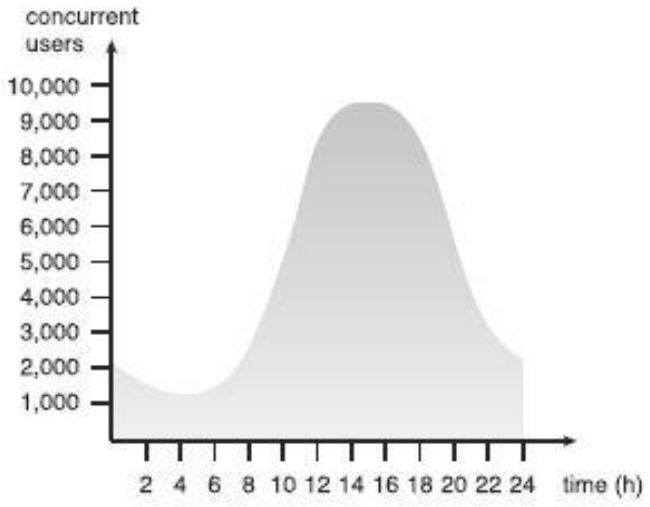


Figure 3.8 An example of an organization's changing demand for an IT resource over the course of a day

The inherent, built-in feature of clouds to provide flexible levels of scalability to IT resources is directly related to the aforementioned proportional costs benefit. Besides the evident financial gain to the automated reduction of scaling, the ability of IT resources to always meet and fulfill unpredictable usage demands avoids potential loss of business that can occur when usage thresholds are met.

3.3.3 Increased Availability and Reliability

The availability and reliability of IT resources are directly associated with tangible business benefits. Outages limit the time an IT resource can be "open for business" for its customers, thereby limiting its usage and revenue generating potential. Runtime failures that are not

immediately corrected can have a more significant impact during high-volume usage periods. Not only is the IT resource unable to respond to customer requests, its unexpected failure can decrease overall customer confidence.

A hallmark of the typical cloud environment is its intrinsic ability to provide extensive support for increasing the availability of a cloud-based IT resource to minimize or even eliminate outages, and for increasing its reliability so as to minimize the impact of run-time failure conditions.

Specifically:

- An IT resource with increased availability is accessible for longer periods of time (for example, 22 hours out of a 24 hour day). Cloud providers generally offer "resilient" IT resources for which they are able to guarantee high levels of availability.
- An IT resource with increased reliability is able to better avoid and recover from exception conditions. The modular architecture of cloud environments provides extensive failover support that increases reliability.

It is important that organizations carefully examine the SLAs offered by cloud providers when considering the leasing of cloud-based services and IT resources. Although many cloud environments are capable of offering remarkably high levels of availability and reliability, it comes down to the guarantees made in the SLA that typically represent their actual contractual obligations.

3.4 Risks and Challenges [1]

Several of the most critical cloud computing challenges pertaining mostly to cloud consumers that use IT resources located in public clouds are presented and examined.

3.4.1 Increased Security Vulnerabilities

The moving of business data to the cloud means that the responsibility over data security becomes shared with the cloud provider. The remote usage of IT resources requires an expansion of trust boundaries by the cloud consumer to include the external cloud. It can be difficult to establish a security architecture that spans such a trust boundary without introducing vulnerabilities, unless cloud consumers and cloud providers hap-pen to support the same or compatible security frameworks—which is unlikely with public clouds.

Another consequence of overlapping trust boundaries relates to the cloud provider's privileged access to cloud consumer data. The extent to which the data is secure is now limited to the security controls and policies applied by both the cloud consumer and cloud provider. Furthermore, there can be overlapping trust boundaries from different cloud consumers due to the fact that cloud-based IT resources are commonly shared.

The overlapping of trust boundaries and the increased exposure of data can provide malicious cloud consumers (human and automated) with greater opportunities to attack IT resources and steal or damage business data. Figure 3.9 illustrates a scenario whereby two organizations accessing the same cloud service are required to extend their respective trust boundaries to the cloud, resulting in overlapping trust boundaries. It can be challenging for the cloud provider to offer security mechanisms that accommodate the security requirements of both cloud service consumers.

3.4.2 Reduced Operational Governance Control

Cloud consumers are usually allotted a level of governance control that is lower than that over on premise IT resources. This can introduce risks associated with how the cloud provider operates its cloud, as well as the external connections that are required for communication between the cloud and the cloud consumer.

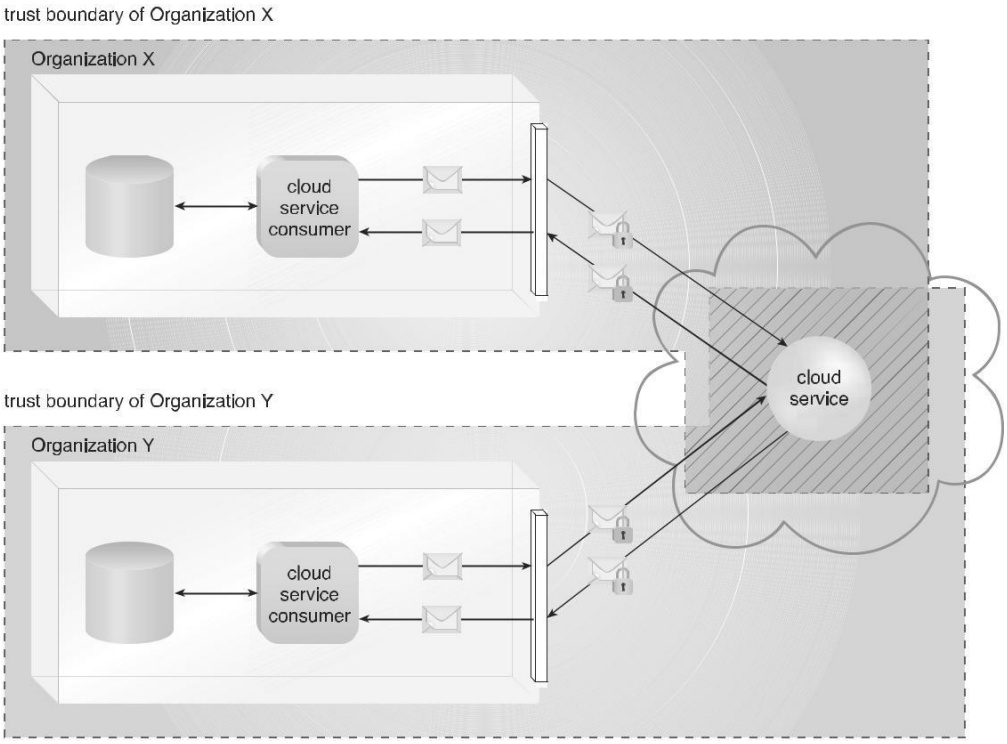


Figure 3.9 The shaded area with diagonal lines indicates the overlap of two organizations' trust boundaries

Consider the following examples:

- An unreliable cloud provider may not maintain the guarantees it makes in the SLAs that were published for its cloud services. This can jeopardize the quality of the cloud consumer solutions that rely on these cloud services.
- Longer geographic distances between the cloud consumer and cloud provider can require additional network hops that introduce fluctuating latency and potential bandwidth constraints.

The latter scenario is illustrated in Figure 3.10.

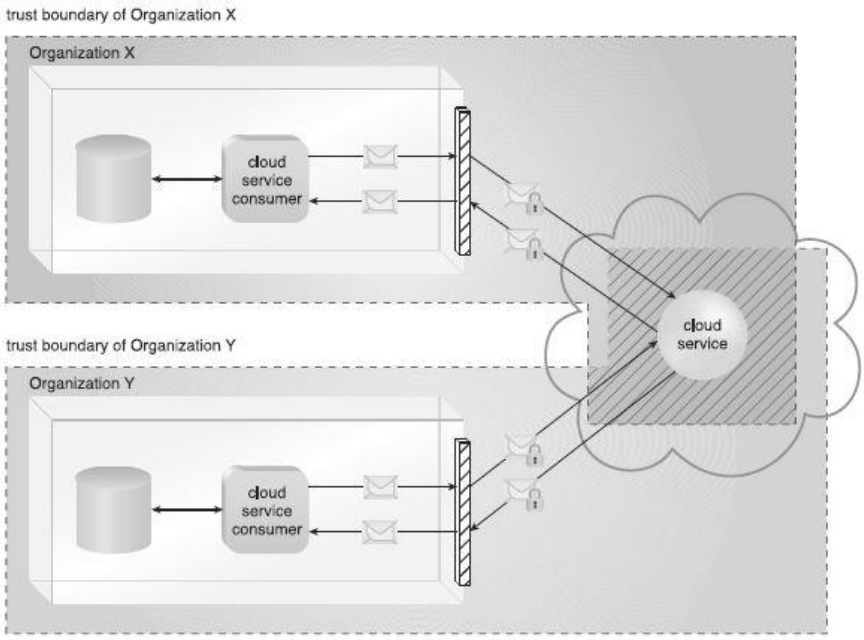


Figure 3.10 An unreliable network connection compromises the quality of communication between cloud consumer and cloud provider environments

Legal contracts, when combined with SLAs, technology inspections, and monitoring, can mitigate governance risks and issues. A cloud governance system is established through SLAs, given the "as-a-service" nature of cloud computing. A cloud consumer must keep track of the actual service level being offered and the other warranties that are made by the cloud provider.

3.4.3 Limited Portability between Cloud Providers

Due to a lack of established industry standards within the cloud computing industry, public clouds are commonly proprietary to various extents. For cloud consumers that have custom-built solutions with dependencies on these proprietary environments, it can be challenging to move from one cloud provider to another. Portability is a measure used to determine the impact of moving cloud consumer IT resources and data between clouds (Figure 3.11).

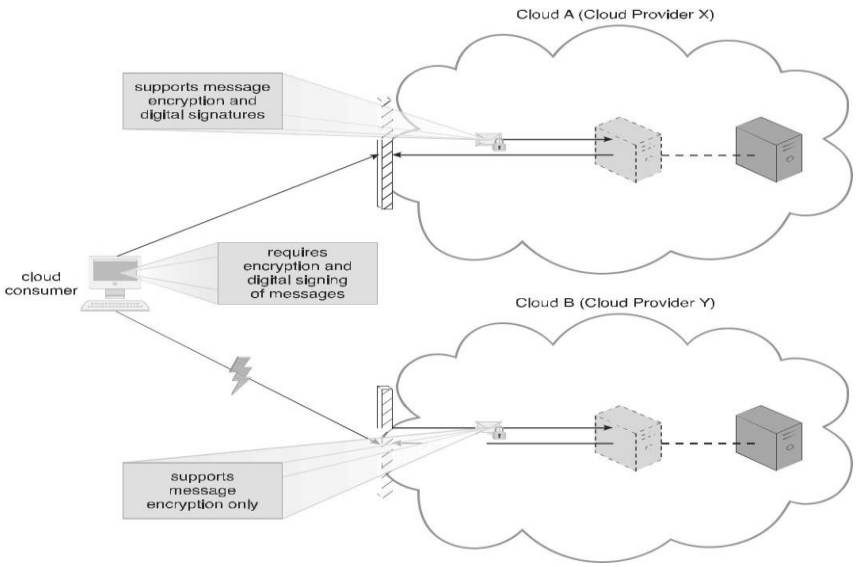


Figure 3.11 A cloud consumer's application has a decreased level of portability when assessing a potential migration from Cloud A to Cloud B, because the cloud provider of Cloud B does not support the same security technologies as Cloud A

3.4.4 Multi-Regional Compliance and Legal Issues

Third-party cloud providers will frequently establish data centers in affordable or convenient geographical locations. Cloud consumers will often not be aware of the physical location of their IT resources and data when hosted by public clouds. For some organizations, this can pose serious legal concerns pertaining to industry or government regulations that specify data privacy and storage policies. For example, some UK laws require personal data belonging to UK citizens to be kept within the United Kingdom.

Another potential legal issue pertains to the accessibility and disclosure of data. Countries have laws that require some types of data to be disclosed to certain government agencies or to the subject of the data. For example, a European cloud consumer's data that is located in the U.S.

can be more easily accessed by government agencies (due to the U.S. Patriot Act) when compared to data located in many European Union countries.

Most regulatory frameworks recognize that cloud consumer organizations are ultimately responsible for the security, integrity, and storage of their own data, even when it is held by an external cloud provider.

3.5 References

[1] Cloud Computing Concept,Technology & Architecture by Thomas Erl, Prentice Hall Publicaton.

CHAPTER 4

ZERO DAY ATTACK

4.1 Introduction [9-13, 15-16, 20, 25, 27, 29, 3 7-38]

A zero-day attack is a cyber-attack exploiting a vulnerability that has not been disclosed publicly. There is almost no defense against a zero-day attack: while the vulnerability remains unknown, the software acted cannot be patched and anti-virus products cannot detect the attack through signature-based scanning. For cyber criminals, unpatched vulnerabilities in popular software, such as Microsoft Office or Adobe Flash, represent a free pass to any target they might wish to attack, from Fortune 500 companies to mil-lions of consumer PCs around the world. For this reason, the market value of a new vulnerability ranges between $5,000{$250,000}. Examples of notable zero-day attacks include the 2010 Hydraq trojan, also known as the \Aurora" attack that aimed to steal information from several companies , the 2010 Stuxnet worm, which combined four zero-day vulnerabilities to target industrial control systems, and the 2011 attack against RSA. Unfortunately, very little is known about zero-day attacks because, in general, data is not available until after the attacks are discovered. Prior studies rely on indirect measurements (e.g., analyzing patches and exploits) or the post-mortem analysis of isolated incidents, and they do not shed light on the duration, prevalence and characteristics of zero-day attacks.

Zero-day vulnerabilities are believed to be used primarily for carrying out targeted attacks, based on the post-mortem analysis of the vulnerabilities that security analysts have connected to zero-day attacks. However, prior research has focused on the entire window of exposure to a vulnerability, which lasts until all vulnerable hosts are patched and which covers attacks initiated after the vulnerability was disclosed. For example, a study of three exploit archives showed that

15% of these exploits were created before the disclosure of the corresponding vulnerability. A follow-up study found that only 65% of vulnerabilities in software running on a typical Windows host have patches available at disclosure, which provides an opportunity for attackers to exploit the unpatched vulnerabilities on a larger scale. These studies do not discern the security vulnerabilities that are ultimately exploited in the wild, and they do not provide any information on the window of opportunity for stealth attacks, before the vulnerabilities exploited have been dis-closed publicly.

We conduct a systematic study of zero-day attacks between 2008 -2011. We develop a technique for identifying and analyzing zero-day attacks from the data available through the Worldwide Intelligence Network Environment (WINE), a platform for data intensive experiments in cyber security. WINE includes field data collected by Symantec on 11 million hosts around the world. These hosts do not represent honeypots or machines in an artificial lab environment; they are real computers that are targeted by cyber-attacks. For example, the binary reputation data set includes information on binary executables downloaded by users who opt in for Symantec's reputation-based security program (which assigns a reputation score to binaries that are not known to be either benign or malicious). The anti-virus telemetry data set includes reports about host-based threats (e.g., viruses, worms, trojans) detected by Symantec's anti-virus products.

The key idea behind our technique is to identify executable les that are linked to exploits of known vulnerabilities. We start from the public information about disclosed vulnerabilities (i.e., vulnerabilities that have been assigned a CVE identifier), available from vulnerability databases and vendor advisories. We use the public Threat Explorer web site to determine threats identified by Symantec that are linked to these vulnerabilities, and then we query the anti-virus telemetry data set in WINE for the hashes of all the distinct les (malware variants) that are detected by these signatures. Finally, we search the history of binary reputation submissions for these malicious les, which allows us to estimate when and where they appeared on the Internet. Correlating these independently-collected data sets allows us to study all the phases in the vulnerability life-cycle. For example, when we end records for the presence of a malicious

executable in the wild before the corresponding vulnerability was disclosed, we have identified a zero-day attack.

To the best of our knowledge, this represents the first attempt to measure the prevalence and duration of zero-day attacks, as well as the impact of vulnerability disclosure on the volume of attacks observed. We identify 18 vulnerabilities exploited in the real-world before disclosure. Out of these 18 vulnerabilities, 11 were not previously known to have been employed in zero-day attacks, which suggests that zero-day attacks are more common than previously thought. A typical zero-day attack lasts on average 312 days and hits multiple targets around the world; however, some of these attacks remain unknown for up to 2.5 years. After these vulnerabilities are disclosed, the volume of attacks exploiting them increases by up to 5 orders of magnitude.

These findings have important technology and policy implications. The challenges for identifying and analyzing elusive threats, such as zero-day attacks, emphasize that experiments and empirical studies in cyber security must be conducted at scale by taking advantage of the resources that are available for this purpose, such as the WINE platform. This will allow researchers and practitioners to investigate mitigation techniques for these threats based on empirical data rather than on anecdotes and back-of-the-envelope calculations. For example, the fact that zero-day attacks are rare events, but that the new exploits are re-used for multiple targeted attacks, suggests that techniques for assigning reputation based on the prevalence of files can reduce the effectiveness of the exploit. Furthermore, because we quantify the increase in the volume of attacks after vulnerability disclosures, we provide new data for assessing the overall benefit to society of the full disclosure policy, which calls for disclosing new vulnerabilities publicly, even if patches are not yet available.

Table 4.1 Summary of findings

Findings	Implications
Zero-day attacks are more frequent than previously thought: 11 out of 18 vulnerabilities identified were not known zero-day vulnerabilities.	Zero-day attacks are serious threats that may have a significant impact on the organizations affected.
Zero-day attacks last between 19 days and 30 months, with a median of 8 months and an average of approximately 10 months.	Zero-day attacks are not detected in a timely manner using current policies and technologies.
Most zero-day attacks affect few hosts, with the exception of a few high-profile attacks (e.g., Stuxnet).	Most zero-day vulnerabilities are employed in targeted attacks.
58% of the anti-virus signatures are still active at the time of writing.	Data covering 4 years is not sufficient for observing all the phases in the vulnerability lifecycle.
After zero-day vulnerabilities are disclosed, the number of malware variants exploiting them increases 183–85,000 times and the number of attacks increases 2–100,000 times.	The public disclosure of vulnerabilities is followed by an increase of up to five orders of magnitude in the volume of attacks.
Exploits for 42% of all vulnerabilities employed in host-based threats are detected in field data within 30 days after the disclosure date.	Cyber criminals watch closely the disclosure of new vulnerabilities, in order to start exploiting them.

We make three contributions here:

- We propose a method for identifying zero-day attacks from data collected on real hosts and made available to the research community via the WINE platform.
- We conduct a systematic study of the characteristics of zero-day attacks. Our findings are summarized in Table 4.1.
- We compare the impact of zero-day vulnerabilities be-fore and after their public disclosure, and we discuss the implications for the policy of full disclosure.

4.2 Problem Statement and Goals [10]

The Common Vulnerabilities and Exposures (CVE) consortium maintains a database with extensive information about vulnerabilities, including technical details and the disclosure dates, that is a widely accepted standard for academia, governmental organizations and the cyber security industry. For CVE, a vulnerability is a software mistake that allows attackers execute commands as other users, access data that has access restrictions, behave as another user or launch denial of service attack. In general, a zero-day attack is an attack that exploits vulnerabilities not yet disclosed to the public. This is only one phase in the lifecycle of these vulnerabilities (see Figure 4.1). A security vulnerability starts as a programming bug that evades

testing. Cyber criminals sometimes discover the vulnerably, exploit it, and package the exploit with a malicious payload to conduct zero-day attacks against selected targets. After the vulnerability or the exploits are discovered by the security community and described in a public advisory, the vendor of the affected software releases a patch for the vulnerability and security vendors update anti-virus signatures to detect the exploit or the specific attacks. However, the exploit is then reused, and in some cases additional exploits are created based on the patch, for attacks on a larger scale, targeting Internet hosts that have not yet applied the patch. The race between these attacks and the remediation measures introduced by the security community can continue for several years, until the vulnerability ceases to affect end-hosts.

The following events mark this lifecycle (Figure 4.2):

- **Vulnerability introduced.** A bug (e.g., programming mistake, memory mismanagement) is introduced in software that is later released and deployed on hosts around the world (time = t_v)
- **Exploit released in the wild.** Actors in the underground economy discover the vulnerability, create a working exploit and use it to conduct stealth attacks against selected targets (time =t_e).
- **Vulnerability discovered by the vendor.** The vendor learns about the vulnerability (either by dis-covering it through testing or from a third-party report), assesses its severity, assigns a priority for fixing it and starts working on a patch (time =t_d).
- **Vulnerability disclosed publicly.** The vulnerability is disclosed, either by the vendor or on public forums and mailing lists. A CVE identifier (e.g., CVE-2010-2568) is assigned to the vulnerability (time =t_0).
- **Anti-virus signatures released.** Once the vulnerability is disclosed, anti-virus vendors release new signatures for ongoing attacks and created heuristic detections for the exploit. After this point, the attacks can be detected on end-hosts with updated A/V signatures (time =t_s).

- **Patch released.** On the disclosure date, or shortly afterward, the software vendor releases a patch for the vulnerability. After this point, the hosts that have applied the patch are no longer susceptible to the exploit (time =t_p).
- **Patch deployment completed.** All vulnerable hosts worldwide are patched and the vulnerability ceases to have an impact (time =t_a).

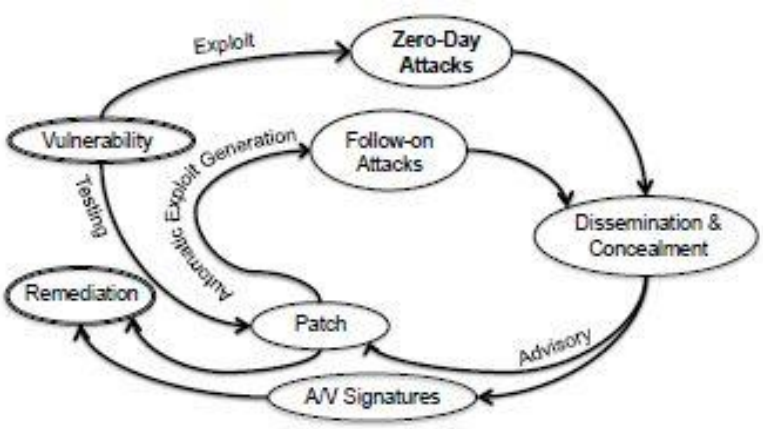


Figure 4.1: Lifecycle of zero-day vulnerabilities

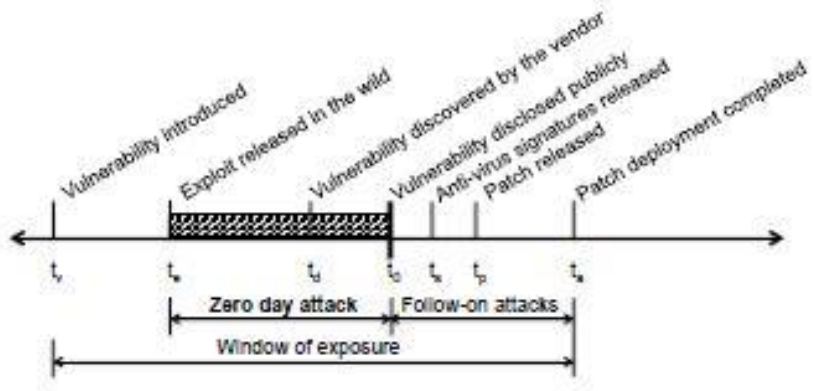


Figure 4.2: Attack timeline. These events do not always occur in this order, but $t_a > t_p \geq t_d > t_v$ and $t_0 \geq t_d$. The relation between t_d and t_e. cannot be determined in most cases. For a zero-day attack $t_0 > t_e$.

A zero-day attack is characterized by a vulnerability that is exploited in the wild before it is disclosed, i.e. $t_0 > t_e$. Similarly, a zero-day vulnerability is vulnerability employed in a zero-day attack. Our goals in to measure the prevalence and duration of zero-day attacks and to compare the impact of zero-day vulnerabilities before and after t_0.

Software vendors x bugs and patch vulnerabilities in all their product releases, and as a result some vulnerabilities are never exploited or disclosed. We only con-sider vulnerabilities that have been assigned a CVE identifier. Similarly, in some cases vendors learn about a vulnerability before it is exploited, but consider it low priority, and cyber criminals may also delay the release of exploits until they identify a suitable target, to prevent the discovery of the vulnerability. While the CVE database sometimes indicates when vulnerabilities were reported to the vendors, it is generally impossible to determine the exact date when the vendor or the cyber criminals discovered the vulnerability or even which discovery came first. We therefore consider the disclosure date of the vulnerability as \day zero," the end of the zero-day attack. Moreover, some exploits are not employed for malicious activities before the disclosure date and are disseminated as proofs-of-concept, to help the soft-ware vendor understand the vulnerability and the anti-virus vendors update their signatures. When disclosed vulnerabilities are left unpatched, this creates an opportunity for cyber criminals to create additional exploits and to conduct attacks on a larger scale; however, these attacks can usually be detected by an anti-virus program with up-to-date definitions. We consider only exploits that have been used in real-world attacks before the corresponding vulnerabilities were disclosed.

Non-goals. We do not aim to analyze the techniques used to exploit zero-day vulnerabilities or for packing the mal-ware to avoid detection. We therefore focus on data that highlights the presence and propagation rate of malware on the Internet, rather than on the static or behavioral analysis of the malware samples. The motivations behind zero-day attacks, the dynamics of the market for zero-day vulnerabilities and the attacks against disclosed vulnerabilities for which patches are not available are also outside the scope of this study.

4.3 Related Work [3, 5, 12-15,18,20,24,29,32,37]

Frei studied zero-day attacks by combining publicly avail-able information on vulnerabilities disclosed between 2000 - 2007 with a study of three exploit archives, popular in the hacker community. This study showed that, on the disclosure date, an exploit was available for 15% of vulnerabilities and a patch was available for 43% of vulnerabilities (these percentages are not directly comparable because they are computed over different bases all vulnerabilities that have known exploits and all vulnerabilities that have been patched, respectively). The study also found that 94% of exploits are created within 30 days after disclosure. However, the exploits included in public archives are proofs-of-concept that are not always used in real-world attacks. Shahzad et al. conduct a similar study, but on a larger data set. The authors analyze how the type and number of vulnerabilities change during the period of their analysis window. McQueen et al. analyze the lifespan of known zero-day vulnerabilities in order to be able to estimate the real number of zero-day vulnerabilities existed in the past. In contrast to this previous work, we analyze field data, collected on real hosts targeted by cyber-attacks, to understand the prevalence and duration of zero-day attacks before vulnerabilities are disclosed, and we conduct a real-world analysis rather than make statistical estimations.

Symantec analysts identified 8-15 zero-day vulnerabilities each year between 2006 -2011. For example, 9 vulnerabilities were used in zero-day attacks in 2008, 12 in 2009, 14 in 2010 and 8 in 2011. The 14 zero-day vulnerabilities dis-covered in 2010 affected the Windows operating system, as well as widely used applications such as Internet Explorer, Adobe Reader, and Adobe Flash Player. These vulnerabilities were employed in high-pro le attacks, such as Stuxnet and Hydraq. In 2009, Qualys analysts reported knowledge of 56 zero-day vulnerabilities. In contrast, to these reports, we propose a technique for identifying zero-day attacks automatically from field data available to the research community, and we conduct a systematic study of zero-day attacks in the real world. In particular, we identify 11 vulnerabilities, disclosed between 2008-2011 that were not known to have been used in a zero-day attack.

Most prior work has focused on the entire window of exposure to a vulnerability (see Figure 4.2), first defined by Schneier. Arbaugh et al. evaluated the number of intrusions observed during each phase of the vulnerability lifecycle and showed that a significant number of vulnerabilities continue to be exploited even after patches become available. Frei compared how fast Microsoft and Apple react to newly disclosed vulnerabilities and, while significant differences exist between the two vendors, both have some vulnerabilities with no patch available 180 days after disclosure. A Secunia study showed that 50% of Windows users were exposed to 297 vulnerabilities in a year and that patches for only 65% of these vulnerabilities were available at the time of their public disclosure. Moreover, even after patches become available, users often delay their deployment, partly because of the overhead of patch management and partly because of the general observation that the process of xing bugs tends to introduce additional software defects. For example, a typical Windows user must manage 14 update mechanisms to keep the host fully patched, while an empirical study suggested that over 10% of security patches have bugs of their own.

While the market for zero-day vulnerabilities has not been studied as thoroughly as other aspects of the underground economy, the development of exploits for such vulnerabilities is certainly a pro table activity. For example, several security firms run programs, such as HP's Zero Day Initiative and Verisign's iDefense Vulnerability Contributor Program that pay developers up to $10,000 for their exploits, with the purpose of developing intrusion-protection filters against these exploits. Between 2000 -2007, 10% of vulnerabilities have been disclosed through these programs. Similarly, software vendors often reward the discovery of new vulnerabilities in their products, offering prizes up to $60,000 for exploits against targets that are di cult to attack, such as Google's Chrome browser. Moreover, certain firms and developers specialize in selling exploits to confidential clients on the secretive, but legal, market for zero-day vulnerabilities. Industry sources suggest that the market value of such vulnerabilities can reach $250,000. In particular, the price of exploits against popular platforms, such as Windows, iOS or the major

web browsers, may exceed $100,000, depending on the complexity of the exploit and on how long the vulnerability remains undisclosed.

4.4 Identifying Zero-Day Attacks Automatically [1, 19, 21, 25, 37-39]

To identify zero-day attacks automatically, we analyze the historical information provided by multiple data sets. In this section, we describe our data sets and the ground truth for our analysis. We then introduce our method for identifying zero-day attacks and discuss the threats to the validity of our findings.

4.4.1 Data sets

We conduct our study on the Worldwide Intelligence Net-work Environment (WINE), a platform for data intensive experiments in cyber security. WINE was developed at Symantec Research Labs for sharing comprehensive field data with the research community. WINE samples and aggregates multiple terabyte-size data sets, which Symantec uses in its day-to-day operations, with the aim of supporting open-ended experiments at scale. The data included in WINE is collected on a representative subset of the hosts running Symantec products, such as the Norton Antivirus. These hosts do not represent honeypots or machines in an artificial lab environment; they are real computers, in active use around the world, that are targeted by cyber-attacks. WINE also enables the reproduction of prior experimental results, by archiving the reference data sets that researchers use and by recording information on the data collection process and on the experimental procedures employed.

We correlate the WINE data sets with information from three additional sources: the Open Source Vulnerability Database (OSVDB), Symantec's Threat Explorer, and a Symantec data set with dynamic analysis results for malware samples. While we process the data provided by OSVDB and Symantec Threat Explorer for forming the basis of our ground truth, we analyze WINE and the dynamic analysis results, in a further stage, to identify the zero-day attacks.

OSVDB is a public database that aggregates all the avail-able sources of information about vulnerabilities that have been disclosed since 1998. Because the Microsoft Windows platform has been the main target for cyber-attacks over the past decade, we focus on vulnerabilities in Windows or in software developed for Windows. The information we collected from OSVDB includes the discovery, disclosure and exploit release date of the vulnerabilities. To complete the picture of the vulnerability lifecycle, we collect the patch release dates from Microsoft and Adobe Security Bulletins.

Threat Explorer is a public web site with up-to-date in-formation about the latest threats, risks and vulnerabilities. In addition, it provides detailed historical information about most threats for which Symantec has generated anti-virus signatures. From these details, we are only interested in the malware class of the threat (e.g., Trojan, Virus, Worm), the signature generation date and associated CVE identifier(s), if the threat exploits known vulnerabilities. We build the ground truth of this study by combining information from OSVDB and Symantec Threat Explorer to prepare a list of threats along with the vulnerabilities they exploit.

We analyze two WINE data sets: anti-virus telemetry and binary reputation. The anti-virus telemetry data records detections of known threats for which Symantec generated a signature that was subsequently deployed in an anti-virus product. The anti-virus telemetry data in WINE was collected between December 2009 and August 2011, and it includes 225 million detections that occurred on 9 million hosts. From each record, we use the detection time, the associated threat label, the hash (MD5 and SHA2) of the malicious le, and the country where the machine resides. We use this data in two ways: first, to link the threat labels with malicious les, and second, to enrich our knowledge about the impact of zero-day vulnerabilities after they are publicly disclosed.

The binary reputation data, on the other hand, does not record threat detections. Instead, it reports all the binary executables whether benign or malicious that have been downloaded on end-hosts around the world. The binary reputation data in WINE was collected since February

2008, and it includes 32 billion reports about approximately 300 million distinct les, which were downloaded on 11 million hosts. Each report includes the download time, the hash (MD5 and SHA2) of the binary, and the URL from which it was downloaded these les may include malicious binaries that were not detected at the time of their download because the threat was unknown. We note that this data is collected only from the Symantec customers who gave their consent to share it. The binary reputation data allows us to look back in time to get more insights about what happened before signatures for malicious binaries were created. There-fore, analyzing this data set enables us to discover zero-day attacks conducted in the past.

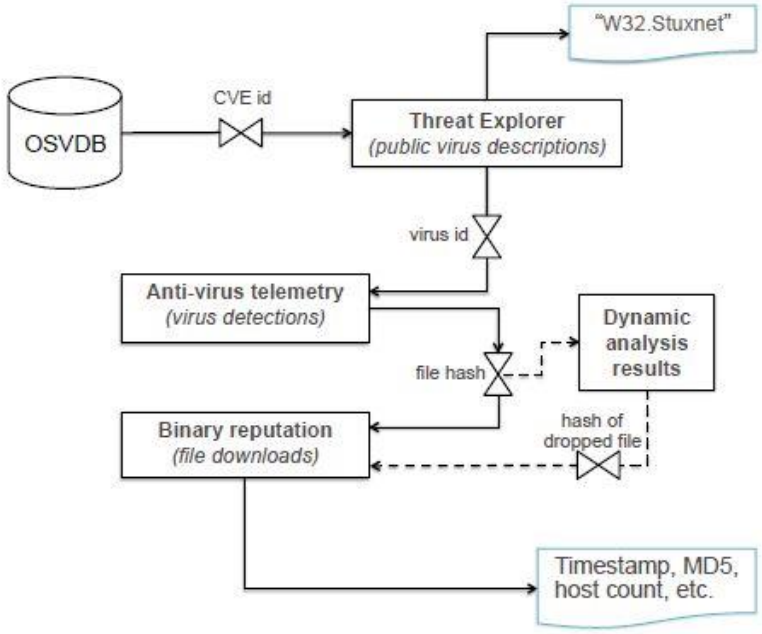


Figure 4.3: Overview of our method for identifying zero-day attacks systematically

In the recent years, most exploits are embedded in non-executable les such as *.pdf, *.doc, *.xlsx. Because the binary reputation data only reports executable les, it is not straightforward to find out whether a non-executable exploit was involved in a zero-day attack or not. To analyze non-executable exploits, we try to identify a customized malicious binary that was downloaded after a successful exploitation, and we then search the binary in the binary reputation data. To

this end, we search the dynamic analysis data set to create a list of binaries that are downloaded after the exploitation phase.

4.4.2 Method for identifying zero-day attacks

Figure 4.3 illustrates our analysis method, which has five steps: building the ground truth, identifying exploits in executables, identifying executables dropped after exploitation (optional phase), analyzing the presence of exploits on the Internet, and identifying zero-day attacks.

Building the ground truth. We first gather information about vulnerabilities in Windows and in software running on the Windows platform by querying OSVDB and other references about disclosed vulnerabilities (e.g., Microsoft Bulletins). For all the vulnerabilities that are identified by a CVE number we collect the discovery, disclosure, exploit re-lease date and patch release dates. We then search Symantec's Threat Explorer for these CVE numbers to identify the threats that exploit these vulnerabilities. Each threat has a name (e.g., W32.Stuxnet) and a numerical identifier, called virus_id. We manually filter out the virus_ids that correspond to generic virus detections (e.g., “Trojan Horse"), as identified by their Threat Explorer descriptions [38]. This step results in a mapping of threats to their corresponding CVE identifiers, Zi = {virus_id_i, cve_id_i}, which are our candidates for the zero-day attack study. Note that some virus ids use more than one vulnerability, therefore in Z_i it is possible to observe the same virus_id more than once.

- **Identifying exploits in executables**

In the second stage our aim is to identify the exploits that are detected by each virus_id in Z so that we can search for them in the binary reputation data. The anti-virus telemetry data set records the hashes of all the malicious files identified by Symantec's anti-virus products. We represent each file recorded in the system with an identifier (file_hash_id). Certain virus ids detect a large number of le hash ids because of the poly-morphism employed by malware authors

to evade detection. This step results in a mapping of threats to their variants, E_i = {virus_id_i; file hash_id_i}.

- **Identifying executables dropped after exploitation**

When exploits are embedded in non-executable les, we can find their file_ hash_id_s in the anti-virus telemetry data but not in the binary reputation data. To detect zero-day at-tacks that employ such exploits, we query the dynamic analysis data set for les that are downloaded after successful exploitations performed by the le hash ids identified in the previous step. This step also produces a mapping of threats to malicious les, but instead of listing the exploit les in E we add the dropped binary les. This may result in false positives because, even if we detect a dropped executable in the binary reputation data before the disclosure date of the corresponding vulnerability, we cannot be con dent that this executable was linked to a zero-day attack. In other words, the executable may have been downloaded using other infection techniques. Therefore, this step is optional in our method.

- **Analyzing the presence of exploits on the Internet**

Having identified which executables exploit known cve_id_s, we search for each executable in the binary reputation data to estimate when they rst appeared on the Internet. Be-cause the binary reputation data indicates the presence of these les, and not whether they were executed (or even if they could have executed on the platform where they were discovered), these reports indicate attacks rather than successful infections. As some virus ids match more than one variant, the first executable detected marks the start of the attack. After this step, for each virus id in Z we can approximate the time when the attack started in the real world.

- **Identifying zero-day attacks**

Finally, to find the virus_id_s involved in zero-day attacks we compare the start dates of each attack with the disclosure dates of the corresponding vulnerabilities. If at least one of the le hash ids of a threat Z_i = {virus_id_i, cve_id_i} was downloaded be-fore the disclosure date of cve_id_i,

we conclude that cve_id_i is a zero-day vulnerability and that virus idi performed a zero-day attack.

4.4.3 Threats to validity

The biggest threat to the validity of our results is selection bias. As WINE does not include data from hosts with-out Symantec's anti-virus products, our results may not be representative of the general population of platforms in the world. In particular, users who install anti-virus software might be more careful with the security of their computers and, therefore, might be less exposed to attacks. Although we cannot rule out the possibility of selection bias, the large size of the population in our study (11 million hosts and 300 million les) and the amount of zero-day vulnerabilities we identify using our automated method (18,which is on the same order of magnitude as reported as the 31 Symantec analysts during the same period) suggest that our results have a broad applicability.

Moreover, for the zero-day vulnerabilities detected toward the beginning of our data collection period, we may underestimate the duration of the attacks. We therefore caution the reader that our results for the duration of zero-day attack are best interpreted as lower bounds.

4.5. Analysis Results and Findings [27-28, 33-38]

In this section, we analyze the zero-day vulnerabilities to build our ground truth starting from a January 2012 copy of the OSVDB database. The binary reputation data we analyze was collected between February 2008 and March 2012. As this is the key component of our method, we can only identify zero-day attacks that occurred between 2008 and 2011.

Table 4.2: The 0-day vulnerabilities identified by automated method

0-day vulnerability	Anti-virus signatures	Disclosure Date	Public Exploit Release	Attack Start Date	Variants	Hosts targeted
CVE-2008-0015	Bloodhoud.Exploit.259	2009-07-06	Not known	2008-12-28	1	2
CVE-2008-2249	Bloodhound.Exploit.214	2008-12-09	Not known	2008-10-14	1	1
CVE-2008-4250	W32.Downadup W32.Downadup.B W32.Fujacks.CE W32.Neeris.C W32.Wapomi.B	2008-10-23	2008-10-23	2008-02-05	312	450 K
CVE-2009-0084	Bloodhound.Exploit.238	2009-04-14	Not known	2008-10-23	3	3
CVE-2009-0561	Bloodhound.Exploit.251	2009-06-09	Not known	2009-01-11	1	1
CVE-2009-0658	Trojan.Pidief	2009-02-20	Not known	2008-09-02	7	23
CVE-2009-1134	Bloodhound.Exploit.254	2009-06-09	Not known	2008-07-25	1	20 K
CVE-2009-2501	Bloodhoud.Exploit.277	2009-10-13	Not known	2009-01-07	6	12
CVE-2009-3126	Bloodhound.Exploit.278	2009-10-13	Not known	2009-01-27	6	16
CVE-2009-4324	Trojan.Pidief.H	2009-12-14	2009-12-15	2009-03-15	1	3
CVE-2010-0028	Bloodhound.Exploit.314	2010-02-10	Not known	2008-10-14	127	102
CVE-2010-0480	Bloodhound.Exploit.324	2010-04-14	Not known	2010-03-26	1	1
CVE-2010-1241	Bloodhound.Exploit.293	2010-04-11	Not known	2008-11-29	2	3
CVE-2010-2568	Bloodhound.Exploit.343 W32.Stuxnet W32.Changeup.C W32.Ramnit	2010-07-17	2010-07-18	2008-02-13	3597	1.5 M
CVE-2010-2862	Bloodhound.Exploit.353	2010-08-04	Not known	2009-03-05	4	18
CVE-2010-2883	Bloodhound.Exploit.357	2010-09-08	2010-09-07	2008-12-14	2	18
CVE-2011-0618	Bloodhound.Exploit.412	2011-05-13	Not known	2010-01-03	1	1
CVE-2011-1331	Trojan.Tarodrop.L	2011-06-16	Not known	2009-03-19	13	32

We first apply our method without the optional step that takes into account the dynamic analysis data. As shown in Table 4.2, we identify 18 zero-day vulnerabilities: 3 dis-closed in 2008, 7 in 2009, 6 in 2010 and 2 in 2011. The second column of the table lists the anti-virus signatures linked to these vulnerabilities; the signatures are described on the Threat Explorer web site. While the exploit les associated with most vulnerabilities were detected by only one anti-virus signature typically a heuristic detection for the exploit there are some vulnerabilities associated with several signatures. For example, CVE-2008-4250 was exploited 8 months before the disclosure date by Conficker (also known as W32.Downadup) and four other worms.

The third column of the table lists the disclosure date of these vulnerabilities, and the fifth column lists the earliest occurrence, observable in WINE, of a le exploiting them. For these vulnerabilities, exploits were active in the real world before disclosure, which indicates that they are zero-day vulnerabilities. For comparison, in the fourth column of Table 4.2 we also report the exploit release dates, as recorded in public vulnerability databases such as OSVDB. This information is available for only 4 out of the 18 vulnerabilities and in all these cases the exploit release date is within one day of the vulnerability disclosure, while working exploits existed in the wild 8-30 months before disclosure. This emphasizes the importance of analyzing field data when studying zero-day attacks.

To determine whether these vulnerabilities were already known to have been involved in zero-day attacks, we manually search all 18 vulnerabilities on Google. From the annual vulnerability trends reports produced by Symantec and the SANS Institute, as well as blog posts on the topic of zero-day vulnerabilities, we found out that 7 of our vulnerabilities are generally accepted to be zero-day vulnerabilities (see Table 4.3). For example, CVE-2010-2568 is one of the four zero-day vulnerabilities exploited by Stuxnet and it is known to have also been employed by another threat for more than 2 years before the disclosure date (17 July 2010). As shown in Table 4.3, most of these vulnerabilities affected Microsoft and Adobe products.

The zero-day attacks we identify lasted between 19 days (CVE-2010-0480) and 30 months (CVE-2010-2568), and the average duration of a zero-day attack is 312 days. Figure 4.4 also illustrates this distribution. The last column in Table 4.2 shows the number of hosts targeted before the zero-day at-tacks was detected. 15 of the zero-day vulnerabilities targeted fewer than 1,000 hosts, out of the 11 million hosts in our data set. On the other hand, 3 vulnerabilities were employed in attacks that infected thousands or even millions of Internet users. For example, Conficker exploiting the vulnerability CVE-2008-4250 managed to infect approximately 370 thousand machines without being detected over more than two months. This example illustrates the effectiveness of zero-day vulnerabilities for conducting stealth cyber-attacks.

Table 4.3: New 0-day vulnerabilities discovered and their description

0-day vulnerability	New 0-day	Description
CVE-2008-0015		Microsoft ATL Remote Code Executiong Vulnerabilitiy (RCEV)
CVE-2008-2249	Yes	Microsoft Windows GDI WMF Integer Overflow Vulnerability
CVE-2008-4250	Yes	Windows Server Service NetPathCanonicalize() Vulnerability
CVE-2009-0084	Yes	Microsoft DirectX DirectShow MJPEG Video Decompression RCEV
CVE-2009-0561	Yes	Microsoft Excel Malformed Record Object Integer Overflow
CVE-2009-0658		Adobe Acrobat and Reader PDF File Handling JBIG2 Image RCEV
CVE-2009-1134	Yes	Microsoft Office Excel QSIR Record Pointer Corruption Vulnerability
CVE-2009-2501		Microsoft GDI+ PNG File Processing RCEV
CVE-2009-3126	Yes	Microsoft GDI+ PNG File Integer Overflow RCEV
CVE-2009-4324		Adobe Reader and Acrobat newplayer() JavaScript Method RCEV
CVE-2010-0028	Yes	Microsoft Paint JPEG Image Processing Integer Overflow
CVE-2010-0480	Yes	Microsoft Windows MPEG Layer-3 Audio Decoder Buffer Overflow Vulnerability
CVE-2010-1241	Yes	NITRO Web Gallery 'PictureId' Parameter SQL Injection Vulnerability
CVE-2010-2568		Microsoft Windows Shortcut 'LNK/PIF' Files Automatic File Execution Vulnerability
CVE-2010-2862	Yes	Adobe Acrobat and Reader Font Parsing RCEV
CVE-2010-2883		Adobe Reader 'CoolType.dll' TTF Font RCEV
CVE-2011-0618	Yes	Adobe Flash Player ActionScript VM Remote Integer Overflow Vulnerability
CVE-2011-1331		JustSystems Ichitaro Remote Heap Buffer Overflow Vulnerability

We also ask the question whether the zero-day vulnerabilities continued to be exploited up until the end of our experimentation period. Figure 4.5 shows the distribution of the time that we continue to detect anti-virus signatures linked to these vulnerabilities, expressed as a percentage of the time between disclosure and the time of writing. The figure suggests that zero-day vulnerabilities do not loose their popularity after the disclosure date. While two vulnerabilities, CVE-2009-1134 and CVE-2009-2501, ceased to have an impact after being exploited over a year, 58% of the anti-virus signatures are still active at the time of writing. Because of this high fraction of vulnerabilities still in use, it would be meaningless to compute the half-life or the decay of the vulnerability usage. The only conclusion we can draw is that data covering 4 years is not sufficient for observing all the phases in the vulnerability lifecycle (Figure 4.1).

While linking exploits to dropped executables through the dynamic analysis of malware samples may produce false positives, we repeat our experiments taking this data set into account, to see if we can identify more zero-day vulnerabilities. We do not detect additional zero-day attacks in this manner, but this optional step allows us to confirm 2 of the zero-day vulnerabilities that we have already discovered.

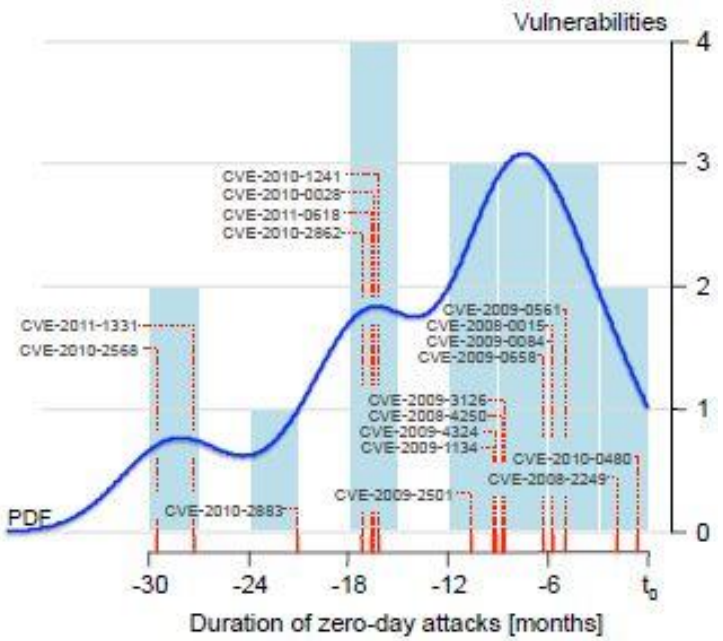


Figure 4.4: Duration of zero-day attacks. The histograms group attack durations in 3-month increments, before disclosure, and the red rug indicates the attack duration for each zero-day vulnerability

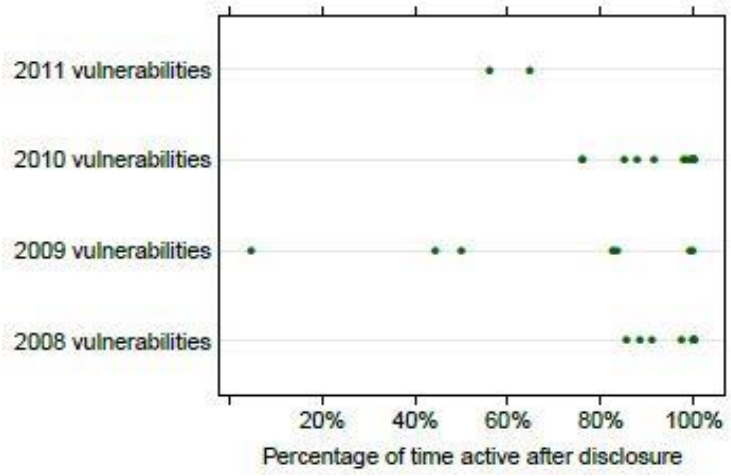
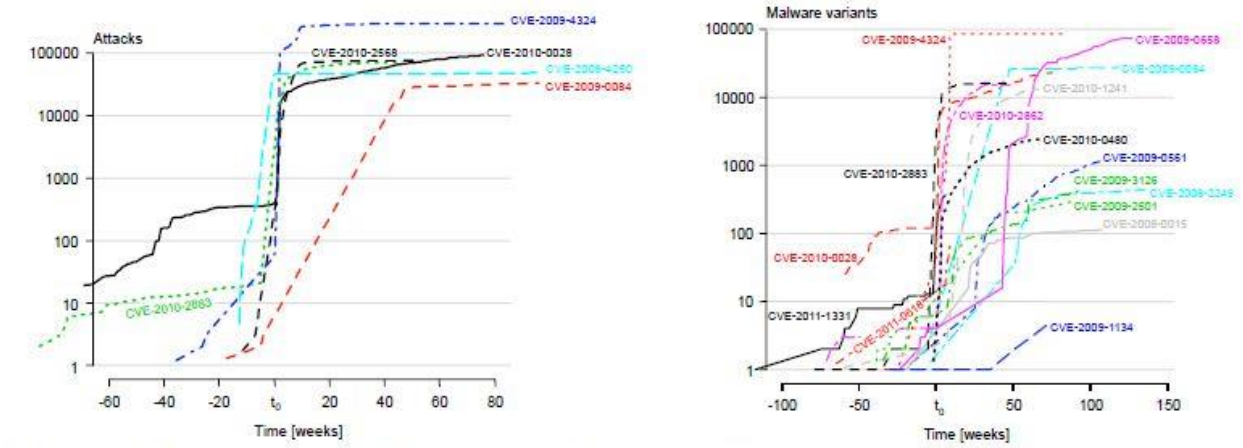


Figure 4.5: Percentage of the period after the dis-closure date that zero-day vulnerabilities are still exploited. Each dot corresponds to an antivirus sig-nature. 100% means that a vulnerability exploit was still in use at the time of writing

4.5.1 Zero-day vulnerabilities after disclosure

To learn what happens after the disclosure of zero-day vulnerabilities, we investigate the volume of attacks exploiting these vulnerabilities over time. Specifically, we analyze the variation of the number of malware variants, as they emerge in the wild, and of the number of times they are detected. Figure 4.6a shows how many downloads (before the disclosure date) and detections (after the disclosure date) of the exploits for the zero-day vulnerabilities were observed until the last exploitation attempt. The number of attacks in-creases 2-100,000 times after the disclosure dates of these vulnerabilities.



(a) Attacks exploiting zero-day vulnerabilities before and after the disclosure (time = t_0).

(b) Malware variants exploiting zero-day vulnerabilities before and after disclosure (time = t_0).

Figure 4.6: Impact of vulnerability disclosures on the volume of attacks. We utilize logarithmic scales to illustrate an increase of several orders of magnitude after disclosure

Figure 4.6b shows that the number of variants (les exploiting the vulnerability) exhibits the same abrupt increase after disclosure: 183-85,000 more variants are detected each day. One reason for observing large number of new different les that exploit the zero-day vulnerabilities might be that they are repacked versions of the same exploits. However, it is doubtful that repacking alone can account for an increase by up to 5 orders of magnitude. More likely, this increase is the result of the extensive reuse of field proven exploits in other malware.

Figure 4.7 shows the time elapsed until all the vulnerabilities disclosed between 2008 and 2011 started being exploited in the wild. Exploits for 42% of these vulnerabilities appear in the field data within 30 days after the disclosure date. This illustrates the fact that the cyber criminals watch closely the disclosure of new vulnerabilities, in order to start exploiting them, which causes a significant risk for end-users.

4.5.2 Other Zero-day Vulnerabilities

Every year, Symantec analysts prepare an “Internet Security Threats Report" (ISTR) in which new threats, vulnerabilities and malware trends are reported. This report includes information about the zero-day vulnerabilities identified during the previous year. These reports identify 31 between 2008 -2011: 9 in 2008, 12 in 2009, 14 in 2010 and 8 in 2011. For each year, our automated method discovers on average 3 zero-day vulnerabilities that were not known before and on average 2 zero-day vulnerabilities from the list reported by Symantec. However, we were not able to identify on average 8 known zero-day vulnerabilities per year.

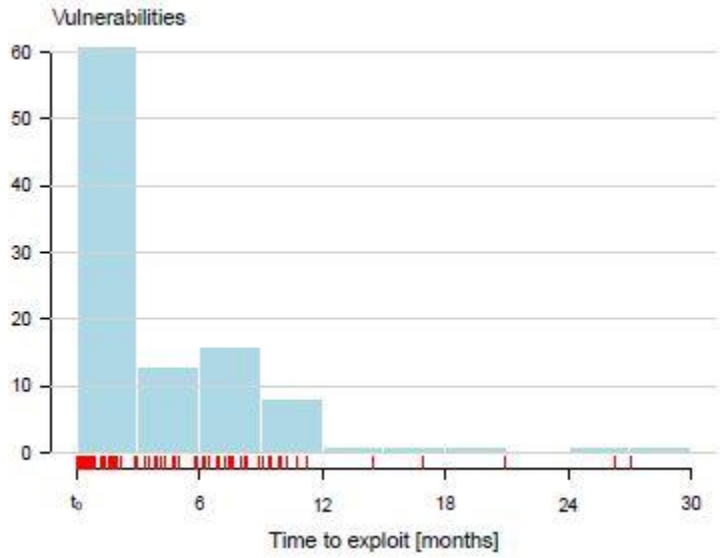


Figure 4.7: Time before vulnerabilities disclosed between 2008-2011 started being exploited in the field.

The histograms group the exploitation lag in 3-month increments, after disclosure, and the red rug indicates the lag for each exploited vulnerability. The zero-day attacks are excluded from the figure 4.7.

To understand the reasons why our method missed 24 zero-day vulnerabilities reported in ISTR, we performed a manual analysis on their characteristics, such as being employed in highly targeted attacks, applying polymorphism etc. This study highlights the limitations of our method:

Web Attacks: Anti-virus telemetry only records detections of host-based attacks. To detect web-based at-tacks, e.g. cross-site scripting attacks, we would need to analyze network based intrusion-detection data. While telemetry from Symantec's intrusion detection products is included in WINE, we did not consider this data set in our study because it is not straight forward to correlate it with the binary reputation data. Our method did not detect CVE-2011-2107, CVE-2011-3765, etc. because these vulnerabilities are exploited in web attacks.

Polymorphic malware: another limitation of our method is that, if the exploit les created for the zero-day vulnerabilities are polymorphic, the le hashes may be di erent in the anti-virus telemetry in binary reputation data. Most of the zero-day exploits that we could not identify were polymorphic, for example, CVE-2010-0806, CVE-2010-3654, CVE-2009-1537.

Non-executable exploits: In recent years, exploits tend to be embedded in non-executable les such as pdf, doc, xlsx. Symantec's anti-virus products pro-vide detections for such malware, and the anti-virus telemetry data contains records for detections of non-executable les. However, the binary reputation data only tracks binary les. Because we use the binary reputation data to approximate the start dates of at-tacks, we cannot detect zero-day vulnerabilities that are exploited by non-executable les. One workaround we considered was to link exploits in non-executable les with the executable dropped once the exploit is successful, by establishing correlations through dynamic analysis results. Unfortunately, results for non-executable les were available only starting in late 2011 (i.e., almost the end of the period covered in our study). Therefore, the dynamic analysis data set pro-vided limited benefits. A representative example for

vulnerabilities that we could not detect due to non-executable files problem is CVE-2011-0609, which was exploited in RSA attack.

Targeted attacks: zero-day vulnerabilities are usually exploited in targeted attacks. Because these attacks target a limited number of organizations, which hold sensitive information than can be stolen, most consumers are not exposed to these attacks. Even though we analyze binary reputation data collected on 11 million hosts, this may not be is not sufficient for identifying zero-day attacks that are highly targeted.

4.6. Discussion [2, 4, 6, 8, 9, 12, 13, 15, 17, 22, 26, 29, 30, 31, 40]

Zero-day attacks are di cult to prevent because they exploit unknown vulnerabilities, for which there are no patches and no anti-virus or intrusion-detection signatures. It seems that, as long as software will have bugs and the development of exploits for new vulnerabilities will be a pro table activity, we will be exposed to zero-day attacks. In fact, 60% of the zero-day vulnerabilities we identify in our study were not known before, which suggests that there are many more zero-day attacks than previously thought perhaps more than twice as many. However, reputation-based technologies, which assign a score to each le based on its prevalence in the wild and on a number of other inputs, single out rare events such as zero-day attacks and can reduce the effectiveness of the exploits.

The large fraction of new zero-day vulnerabilities we identify also emphasizes that zero-day attacks are di cult to detect through manual analysis, given the current volume of cyber-attacks. Automated methods for finding zero-day attacks in field data, such as the method that facilitate the systematic study of these threats. For example, our method allows us to measure the duration of zero-day attacks (Figure 4.4). While the average duration is approximately 10 months, the fact that all but one of the vulnerabilities disclosed after 2010 remained unknown for more than 16 months suggests that we may be underestimating the duration of zero-day attacks, as the data we analyze goes back only to February 2008. In the future, such automated techniques will allow analysts to detect zero-day attacks faster, e.g., when a new exploit is reused in multiple targeted

attacks. However, this will require establishing mechanisms for organizations to share information about suspected targeted attacks with the security community.

Our findings also provide new data for the debate on the benefits of the full disclosure policy. This policy is based on the premise that disclosing vulnerabilities to the public, rather than to the vendor, is the best way to x them be-cause this provides an incentive for vendors to patch faster, rather than to rely on security-through-obscurity. This debate is ongoing, but most participants agree that disclosing vulnerabilities causes an increase in the volume of attacks. Indeed, this is what the supporters of full disclosure are counting on, to provide a meaningful incentive for patching. However, the participants to the debate dis-agree about whether trading off a high volume of attacks for faster patching provides an overall benefit to the society. For example, Schneier initiated the debate by suggesting that, to mitigate the risk of disclosure, we should either patch all the vulnerable hosts as soon as the x becomes available, or we should limit the information available about the vulnerability. Ozmet et al. concluded that disclosing information about vulnerabilities improves system security, while Rescorla et al. could not find the same strong evidence on a more limited data set. Arora et al. and Cavusoglu et al. analyzed the impact of full disclosure using techniques inspired from game theory, and they reached opposite conclusions about whether patches would immediately follow the disclosure of vulnerabilities.

The root cause of these disagreements lies in the difficulty of quantifying the real-world impact of vulnerability disclosures and of patch releases without analyzing comprehensive field data. We take a first step toward this goal by showing that the disclosure of zero-day vulnerabilities causes a significant risk for end-users, as the volume of attacks in-creases by up to 5 orders of magnitude. However, vendors prioritize which vulnerabilities they patch, giving more urgency to vulnerabilities that are disclosed or about to be disclosed. For example, 80% of the 2007 vulnerabilities were discovered more than 30 days before the disclosure date. Moreover, even after patches become available users often delay their deployment, e.g., because a typical Windows user must manage 14 update mechanisms to keep the host fully patched. At the same time, anecdotal evidence suggests that attackers also adapt their strategies to the expected

disclosure of zero-day vulnerabilities. For example, the 2004 Witty worm was released less than 48 hours after the vulnerability it exploited was disclosed, which raised the suspicion that the attacker did not utilize a working exploit until the deployment of the patch was imminent; the exploit le used in the 2011 attack against RSA was sent to 15 different organizations in the two weeks leading to the vulnerability's disclosure, in an attempt to exploit is as much as possible before it was discovered and patched. This is because early disclosure reduces the value of zero-day vulnerabilities; for example, some fees for new exploits are paid in installments, with each subsequent payment depending on the lack of a patch. Additional research is needed for quantifying these aspects of the full disclosure trade-off, e.g., by measuring how quickly vulnerable hosts are patched in the field, following vulnerability disclosures. Like our study of zero-day attacks, answering these additional research questions will require empirical studies conducted at scale, using comprehensive field data.

4.7. Conclusion

Zero-day attacks have been discussed for decades, but no study has yet measured the duration and prevalence of these attacks in the real world, before the disclosure of the corresponding vulnerabilities. We take a first step in this direction by analyzing field data collected on 11 million Windows hosts over a period of 4 years. The key idea in our study is to identify executable les that are linked to exploits of known vulnerabilities. By searching for these les in a data set with historical records of les downloaded on end-hosts around the world, we systematically identify zero-day attacks and we analyze their evolution in time.

We identify 18 vulnerabilities exploited in the wild before their disclosure, of which 11 were not previously known to have been employed in zero-day attacks. Zero-day attacks last on average 312 days, and up to 30 months, and they typically affect few hosts. However, there are some exceptions for high pro le attacks such as Conficker and Stuxnet, which we respectively detected on hundreds of thousands and millions of the hosts in our study, before the vulnerability disclosure. After the disclosure of zero-day vulnerabilities, the volume of attacks exploiting them

increases by up to 5 orders of magnitude. These findings have important implications for future security technologies and for public policy.

4.8 References

[1] Adobe Systems Incorporated. Security bulletins and advisories. http://www.adobe.com/support/security/, 2012.

[2] R. Anderson and T. Moore. The economics of information security. In Science, vol. 314, no. 5799, 2006.

[3] W. A. Arbaugh, W. L. Fithen, and J. McHugh. Windows of vulnerability: A case study analysis. IEEE Computer, 33(12), December 2000.

[4] A. Arora, R. Krishnan, A. Nandkumar, R. Telang, and Y. Yang. Impact of vulnerability disclosure and patch availability - an empirical analysis. In Workshop on the Economics of Information Security (WEIS 2004), 2004.

[5] S. Beattie, S. Arnold, C. Cowan, P. Wagle, and C. Wright. Timing the application of security patches for optimal uptime. In Large Installation System Administration Conference, pages 233-242, Philadelphia, PA, Nov 2002.

[6] J. Bollinger. Economies of disclosure. In SIGCAS Comput. Soc., 2004.

[7] D. Brumley, P. Poosankam, D. X. Song, and J. Zheng. Automatic patch-based exploit generation is possible: Techniques and implications. In IEEE Symposium on Security and Privacy, pages 143-157, Oakland, CA, May 2008.

[8] H. C. H. Cavusoglu and S. Raghunathan. Emerging issues in responsible vulnerability disclosure. In Workshop on Information Technology and Systems, 2004.

[9] D. H. P. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos. Polonium : Tera-scale graph mining for malware detection. In SIAM International Conference on Data Mining (SDM), Mesa, AZ, April 2011.

[10] CVE. A dictionary of publicly known information security vulnerabilities and exposures. http://cve.mitre.org/, 2012.

[11] N. Falliere, L. O'Murchu, and E. Chien. W32.stuxnet dossier. http://www.symantec.com/content/en/us/ enterprise/media/security_response/ whitepapers/w32_stuxnet_dossier.pdf, February 2011.

[12] S. Frei. Security Econometrics: The Dynamics of (In) Security. PhD thesis, ETH Zurich,• 2009.

[13] S. Frei. End-Point Security Failures, Insight gained from Secunia PSI scans. Predict Workshop, February 2011.

[14] Google Inc. Pwnium: rewards for exploits, February 2012. http://blog.chromium.org/2012/02/ pwnium-rewards-for-exploits.html.

[15] A. Greenberg. Shopping for zero-days: A price list for hackers' secret software exploits. Forbes, 23 March 2012. http://www.forbes.com/sites/ andygreenberg/2012/03/23/ shopping-for-zero-days-an-price-list-for-hackers-secret-so.

[16] A. Lelli. The Trojan.Hydraq incident: Analysis of the Aurora 0-day exploit. http://www.symantec.com/connect/blogs/ trojanhydraq-incident-analysis-aurora-0-day-exploit, 25 January 2010.

[17] R. McMillan. RSA spearphish attack may have hit US defense organizations. PC World, 8 September 2011. http://www.pcworld.com/businesscenter/article/ 239728/rsa_spearphish_attack_may_have_hit_us_ defense_organizations.html.

[18] M. A. McQueen, T. A. McQueen, W. F. Boyer, and M. R. Chaffi n. Empirical estimates and observations of 0day vulnerabilities. In Hawaii International Conference on System Sciences, 2009.

[19] Microsoft. Microsoft security bulletins. http:// technet.microsoft.com/en-us/security/bulletin, 2012.

[20] C. Miller. The legitimate vulnerability market: Inside the secretive world of 0-day exploit sales. In Workshop on the Economics of Information Security, Pittsburgh, PA, June 2007.

[21] OSVDB. The open source vulnerability database. http://www.osvdb.org/, 2012.

[22] A. Ozment and S. E. Schechter. Milk or wine: does software security improve with age? In 15th conference on USENIX Security Symposium, 2006.

[23] P. Porras, H. Saidi, and V. Yegneswaran. An anlysis of con cker's logic and rendezvous points. http://mtc.sri.com/Conficker/, 2009.

[24] Qualys, Inc. The laws of vulnerabilities 2.0. http://www.qualys.com/docs/Laws_2.0.pdf, July 2009.

[25] T. Dumitraş and D. Shou. Toward a standard benchmark for computer security research: The Worldwide Intelligence Network Environment (WINE). In EuroSys BADGERS Workshop, Salzburg, Austria, Apr 2011.

[26] E. Rescorla. Is nding security holes a good idea? In IEEE Security and Privacy, 2005.

[27] U. Rivner. Anatomy of an attack, 1 April 2011. http: //blogs.rsa.com/rivner/anatomy-of-an-attack/ Retrieved on 19 April 2012.

[28] SANS Institute. Top cyber security risks - zero-day vulnerability trends. http://www.sans.org/ top-cyber-security-risks/zero-day.php, 2009.

[29] B. Schneier. Cryptogram september 2000 - full disclosure and the window of exposure. http://www.schneier.com/crypto-gram-0009.html, 2000.

[30] B. Schneier. Locks and full disclosure. In IEEE Security and Privacy, 2003.

[31] B. Schneier. The nonsecurity of secrecy. In Commun. ACM, 2004.

[32] M. Shahzad, M. Z. Sha q, and A. X. Liu. A large scale exploratory analysis of software vulnerability life cycles. In Proceedings of the 2012 International Conference on Software Engineering, 2012.

[33] Symantec Corporation. Symantec global Internet security threat report, volume 13. http://eval.symantec.com/mktginfo/enterprise/ white_papers/b-whitepaper_internet_security_ threat_report_xiii_04-2008.en-us.pdf, April 2008.

[34] Symantec Corporation. Symantec global Internet security threat report, volume 14. http://eval.symantec.com/mktginfo/enterprise/ white_papers/b-whitepaper_internet_security_ threat_report_xv_04-2010.en-us.pdf, April 2009.

[35] Symantec Corporation. Symantec global Internet security threat report, volume 15. http://msisac. cisecurity.org/resources/reports/documents/ SymantecInternetSecurityThreatReport2010.pdf, April 2010.

[36] Symantec Corporation. Symantec Internet security threat report, volume 16, April 2011.

[37] Symantec Corporation. Symantec Internet security threat report, volume 17. http://www.symantec.com/threatreport/, April 2012.

[38] Symantec Corporation. Symantec threat explorer. http://www.symantec.com/security_response/ threatexplorer/azlisting.jsp, 2012.

[39] Symantec.cloud. February 2011 intelligence report. http://www.messagelabs.com/mlireport/MLI_2011_ 02_February_FINAL-en.PDF, 2011.

[40] N. Weaver and D. Ellis. Re ections on Witty: Analyzing the attacker. ;login: The USENIX Magazine, 29(3):34{37, June 2004.

CHAPTER 5

SOCIAL ENGINEERING

5.1 Introduction to Social Engineering [1-6]

As we continue to increase our reliance on computer systems by using them to store and process the world's information, they have become increasingly popular targets for attackers looking to disrupt, or steal from, the target company. In response to this threat, companies around the world are projected to invest over 151 billion dollars on IT security projects in 2012 in an attempt to protect their businesses. So why, despite these large investments, do we often see constantly see large companies with strong IT systems suffer from service disruptions and leaked information? According to Kevin Mitnick, described by the US government as the most dangerous hacker in the world, the cause of this could be because "it is much easier to trick someone into revealing a password for a system than to exert the effort of hacking into the system".

Social Engineering, according to Mitnick, is the use of influence and persuasion to deceive people into divulging information. While Mitnick was the first to coin the term, the art of social engineering itself is nothing new; in a crude example, International Intelligence claims that social engineering started "way back in time when man started to lie to woman." However, the art of social engineering has enjoyed a strong resurgence due to the low cost of attacks, anonymity of electronic communication and the ease of researching potential targets thanks to the popularity or email and social engineering sites.

The widespread use of social engineering is undeniable. In the few years alone, we have seen social engineering play a critical role in a number of high profile attacks such as in the government-sponsored attacks that have crippled a highly secure nuclear reactor in Iran and amateur attacks that have been responsible for the leakage of over 500,000 client records by a

cloud application provider. These attacks clearly show that in many cases, exploiting the human mind is easiest way to breach an organization's defenses and that social engineering has made IT security a pervasive problem that cannot simply be solved through the provision of hardware or software.

As auditors, we are uniquely positioned to protect our organizations and those we service against social engineering. Unlike most parties within an organization, we are already familiar with the entire organization's internal controls and have an understanding of day-to-day operations. This helps in the identification of soft spots that a social engineer could exploit when attempting to use social engineering to steal information, passwords or even tangible goods. While auditors may not necessarily possess the technical competency to advice on attacks that incorporate the latest software exploits, the ability to identify potential targets of social engineering, prepare the potential targets from attack and to promote a culture of vigilantly skepticism makes auditors an excellent candidate to address the risks posed by social engineering.

This report will serve as a primer on social engineering by discussing several key aspects of social engineering:

- The psychology that powers social engineering
- The Broad categories of social engineering attacks
- Common areas of vulnerability
- Notable cases of social engineering
- Steps to prevent social engineering attacks from succeeding
- Methods to limit the damage caused in a security breach.

5.2 The Psychology of Social Engineering [7-15]

Although social engineering results in employees simply handing valuable information, products or access to an attacker, it is important to note that the employee generally does not do so maliciously. For example, when social engineers need to gain physical access to an organization,

a common tactic is to tailgate an employee into the building in order to bypass the ID scan that opens the door, an automated control. In this case, the employee holding the door for the attacker would not feel as if they were doing anything wrong; the employee was just following normal social conventions that one holds the door for people behind them.

As the aforementioned example demonstrates, in order to understand why social engineering is so effective, we must first understand the qualities of human nature that social engineers prey upon. Thomas Peltier, the author of several books on information security, suggests that there are four fundamental aspects of human nature that social engineers prey upon: the desire to be helpful, the tendency to be trusting, the fear of offending others and the tendency to cut corners:

5.2.1 The desire to be helpful to others

One of the most popular targets for social engineers is an organization's customer facing personnel8 that provide information and support to external customers because they tend to be easily accessible to an attacker. While companies typically attempt to train these employees to guard confidential information and access to the company's systems by providing detailed conversation scripts, social engineers have found that these employees are easy to manipulate. Customer facing personnel spend every day continually helping a never-ending line of customers and psychological research has shown that it is incredibly difficult in this situation to question the validity of every interaction. Instead, the customer-facing employee will display a tendency to try to help the customer with his or her problem, even if it deviates from the controls put in place by the company to prevent social engineering from succeeding.

For example, a social engineer could take advantage of this by feigning a strong accent or a poor ability to communicate in English in hopes that a call center employee will circumvent the controls in place to better assist the troubled "customer". If successful, the social engineer may be able to bypass security questions put in place to verify the caller's identity.

5.2.2 The tendency to trust others

In his book, The Art of Deception, Kevin Mitnick describes a fatal flaw that most people share: a tendency to have trust and faith in each other. This blind trust in others has resulted in thousands of people believing stories as ridiculous as a Nigerian prince that needs to enlist the help of a random stranger to transfer vast amounts of money out of his own country. While it is possible that the popularization of the Internet has hardened our defenses against such obvious attempts at social engineering such as advance fee fraud perpetuated by "Nigerian prince", this has not changed the fact that we are still very vulnerable to well-crafted social engineering attacks.

In fact, as presenters hold up obvious cases of social engineering in their organization's awareness campaigns, they unfortunately reinforce a tragic misconception that the average person possesses: that they are too smart to be deceived. The result is that the person has an inflated sense of security and will be easily exploited by social engineers that are discreet enough to only make reasonable requests that will draw no suspicion until it is too late.

5.2.3 The fear of offending others

Being raised in a world of political correctness, we have a tendency to worry about offending others through our words or actions. For example, many people would reluctantly spend several minutes listening to a telemarketer's monologue instead of simply hanging up and/or hesitate to stop a uniformed repairman in their place of business.

Social engineers have been able to exploit this characteristic by gathering information on their targets over "telephone surveys" that can be used to reset the target's password on sites that incorporate security questions. For example, common security questions such as, "what is your hometown" or "what is your first pet's name", could easily be captured by posing as a surveyor asking about a target's pets.

As we are so hesitant to stop a uniformed employee from doing their jobs, it is not surprising that social engineers frequently put on disguises to gain access to employee-only areas. By walking

confidently across hallways, the social engineer could gather information or breach systems without any interference at all. For example, by donning the uniform of an electrical technician, a security auditor was able to trick a dozen bank branches into allowing him to wander freely around the premises and install miniature computers onto the organization's network that allowed the auditor to remotely access computers within the network after his departure.

5.2.4 The tendency to cut corners

Above all else, social engineering is incredibly successful because of the inherent laziness present in the average person and their willingness to cut corners, especially when the shirking appears to be relatively harmless or inconsequential. This is because people, unlike computerized controls, will naturally become tired and distracted, especially near the end of a shift.

A social engineer can exploit this tendency of human nature by observing the employees within an organization and making a note of employees that frequently cut corners in their duties. This process could lead to the discovery of a critical control weakness, which will allow the social engineer to develop a plan of attack to bypass the control. For example, a social engineer could notice that a certain clerk at the post office was erroneously not verifying the identity of patrons that came in with delivery slips. The social engineer could then collect other peoples' delivery slips from nearby apartment buildings to claim as his or her own.

5.3 Categories of Social Engineering Attacks [16-17]

While the term social engineering is often used to describe all trickery used to manipulate people into performing actions or giving up information, the rapid development of electronic means of deception have led some security professionals to believe that social engineering should be segregated into human-based and technology-based components.

5.3.1 Human-based attacks

The human-based type of social engineering relies exclusively on person-to-person communications in order to achieve the desired result. The defining element of these attacks is that they do not require any special technical knowledge and rely completely by gathering information through exploiting the fundamental flaws in human nature previously discussed; For that reason, these techniques are timeless and have been successfully employed throughout human history.

Human-based attacks typically employ one or more of the following methodologies:

i. Impersonation: By finding out the reporting relationships in an organization, the attacker impersonates a person within the organization with power over someone with the necessary information or access privileges and asks the subordinate to gather the information or perform the task required. For example, a university student could attempt to boost his or her mark by impersonating a professor on the phone to the teaching assistant.

ii. Third Party Authorization: Through research into the organization, the attacker finds the name of a person that has the authority to authorize a certain course of action and attempts to trick a subordinate to do so. For example, a person trying to gain unauthorized access to a club could attempt to convince the bouncer that the owner personally invited him or her in.

iii. In Person: The attacker can simply walk into a building pretending to be an employee, visitor or a contracted service personnel and gather information left unattended. For example, a social engineer notes that by simply wearing a “Xerox” promotional shirt and carrying a toolbox, he is able to walk into secure areas of almost any company.

iv. Dumpster Diving: The attacker attempts to gain valuable information about the target by going through the company’s trash. This typically leads to valuable information such as discarded bank statements or price lists that are carelessly discarded.

v. Shoulder Surfing: With the rising popularity of laptops in public spaces, a perceptive social engineer could gain information by simply sitting at a coffee shop and watching unsuspecting people reveal their email username and passwords. Using this core piece of info, the social engineer could then gain access to the person's entire online identity.

5.3.2 Technology-based attacks

On the other hand, the advent of modern computing has brought along a new type of technology-based social engineering. Taking advantage of vulnerabilities by using malware or other computer-based exploits, a social engineer attempts to use technology to complement human-based social engineering by either:

i. Tricking the user into installing malware that provides the attacker with remote access to a computer that would otherwise be too secure to breach. This allows the social engineer to gain a level of access to information almost impossible under human-based techniques.

ii. Tricking employees into sending confidential information to the attacker by using technology to create the appearance of legitimate communications. While it can be argued that phishing is nothing new and has existed in traditional mail, the rise of inexpensive anonymous communications has resulted in sophisticated phishing techniques unique to the computer.

Technology-based attacks tend to fall under one of the following methodologies:

i. Software exploits: Technology-based social engineering attacks are constantly evolving in response to new exploits that are found. For example, about a decade ago, Internet Explorer had a flaw that allowed a website owner to read visitors' clipboard contents without their knowledge – social engineers were quick to exploit this vulnerability by finding ways to convince people to copy and paste important information and viewing their webpages.

ii. Email Phishing: By using an email address that looks legitimate, for example, admin@UWATERL00.CA, an attacker may be able to trick a user into responding with

confidential information such as their username and password. Alternatively, the attacker may be able to trick the user into opening infected attachments contained within the email message by naming the attachment with a tempting filename such as “Quest Username/Password List 2012”.

iii. Website Phishing: An attacker could redirect a person to a website with a URL that appears to be legitimate at first glance. As with the email phishing example, this is usually done by swapping out a letter with a similar looking number or inserting a character into a long URL. This would allow an attacker to direct the user to a login screen that is used to harvest passwords from unsuspecting victims.

5.4 Common Areas of Vulnerability [18-19]

One of the first things that a social engineer does is to create a list of the potential targets within the organization. This list tends to be generated through publically available information such as the company website, employee profiles on LinkedIn, or a call to the company itself. In order to narrow down the list of candidates, a social engineer needs to consider a number of factors:

i. Appropriate access: The social engineer must know that the person being targeted has the appropriate credentials to manipulate or access the information or system that is of interest. For example, if a social engineer is attempting to steal confidential information about software being developed by a firm, gaining access into the secretary’s computer may not yield results if the secretary’s computer is not connected to the server that holds the software’s code.

ii. Assessed resistance: Although only 11 of the 138 people put up any resistance to the social engineering attempts demonstrated at DEFCON 18 security conference, the social engineer must evaluate the selected target’s resistance to an attempt at social engineering in order to determine whether the target’s viability. For example, a social engineer attempting a technology-based attack would be interested in factors such as the employee’s level of technical knowledge while an attacker using a human-based approach may seek out disgruntled ex-employees that will not hesitate to share information about their past employer.

iii. Information availability: Google, Facebook and LinkedIn are three sources of information that are universally used by social engineers in order to obtain background information on their targets. For example, attackers at DEFCON 18 all used LinkedIn to recreate organizational charts by piecing together the profiles of employees at a given company so that they could select an ideal target.

Using this information, attackers are able to develop a plan of attack tailored to the employee. For example, if it is discovered through a search of Facebook that a certain employee plays Farmville during work hours, an attacker may choose to call this employee while pretending to be from the company's IT department noting that an unauthorized site has been visited so that the employee believes that that the attacker must be legitimately IT personnel due to the knowledge of his or her browsing history.

Therefore, it is no surprise that we often find that secretaries, assistants and call-center personnel are often targeted for their access to large amounts of sensitive information and low perceived amount of technical knowledge about information security. Despite the higher likelihood that security experts within the organization would be more likely to spot technology-based attacks, they remain a high-value target to social engineers because of their nearly unlimited access to the company's information systems.

5.5 Notable Cases of Social Engineering [20-21]

Since social engineering relies on deceiving an insider into proving access or information, the target company may never notice a perfectly executed attack. Even when the company is aware of an attack, in most cases, the company will do everything in their power to keep the breach a secret in fear of lawsuits and an overall weakened reputation. However, this secrecy shrouds the community as a whole from the importance of effective defenses against social engineering and its power to completely bypass both hardware and software defenses.

5.5.1 Attacks against individuals

Developers of online games use unique CD keys in order to prevent pirated versions of their game from accessing online content. In order to obtain a valid CD key, many pirates turn to social engineering to get a valid key by tricking legitimate customers on online communities. As seen in exhibit I, the attacker does this by posting a set of instructions on how to convert one CD key into multiple keys online, waiting for people to email with the resulting key that doesn't work, and reversing the process to find the legitimate key.

Another common attack is to send a phishing email in an attempt to trick the recipient into entering their username and password into a site that is controlled by the attacker. This is done by sending an official looking email to the target, urging them to log onto the bank's site, along with a fake link that leads to the attacker's site as shown in exhibit II.

5.5.2 Attacks against organizations

One of the most high profile cases of social engineering in recent history involved the destruction of components of an Iranian power plant through a joint American-Israeli cyber-espionage operation. Critical infrastructures such as these power plants were once through to be absolutely impossible to hack because they are not connected to the Internet at all. However, through social engineering, the Americans and the Israelis were able to trick plant workers into bringing infected USB drives into the plant21and plugging them into the machines that once were isolated from the outside world.

The result of this breach is that the US and Israeli government was able to completely destroy critical pieces of equipment and set the Iranian nuclear program back by months without stepping foot inside the facility. In addition, if the governments did not later choose to claim responsibility for the attack, it would have been completely untraceable.

5.6 Preventing Social Engineering Attacks [22-24]

As auditors, we have already assumed a great deal of responsibility by ensuring that our organizations' systems meet the control objectives as identified by the CICA IT Control Guidelines. Notably, the control objectives that directly relate to security over IT are as follows:

i. We must ensure the integrity, confidentiality and availability of information technology processing throughout the enterprise.

ii. We must ensure that access to the enterprise's systems and information is reliably controlled.

iii. To ensure that appropriate consideration is given to security issues and technical skills when management and staff are hired into IT positions.

Despite this responsibility, the results from the DEFCON social engineering contest have clearly demonstrated that all companies are vulnerable against social engineers. The following are a series of practical solutions to common social engineering techniques.

i. Human-based controls should be replaced with electronic controls where possible. This will eliminate the ability of employees to be tempted into circumventing the system by a social engineer. For example, if the company's call-center policy is that no information can be released prior to a caller verifying their name, address and credit card number over the phone, the company can implement a control where the employee must enter all three pieces of information correctly before being able to view the information within the system.

ii. Despite lectures and workshops about security and password protection, people have been shown to constantly write their passwords down or refuse to use different passwords for their various accounts. As a result, two-factor authentication should be considered. Two-factor authentication uses a token with a constantly changing second password that must be entered in conjunction with the user's static password. A social engineer would have great difficulty convincing an employee that there is any legitimate reason why the token's password is required,

especially if a warning is printed on the token itself. See exhibit III for an example of a strong implementation of two-factor authentication.

iii. Regularly educate employees on the techniques that are employed by social engineers, conduct internal audits regularly to show employees how an attacker could have potentially gained access to the system, and ensure that employees that fail to follow best practices are reprimanded. It is simply not good enough to show employees any single case of social engineering: it is critical to train the employee to be able to heuristically identify when they are being targeted and alert the company to the attack.

iv. Every employee with access to confidential information should be provided with a shredder at his or her desk. Given the human nature to be lazy, sensitive information will invariably end up in garbage cans or a shred pile of sensitive information will end up piled up in an easy-to-steal box otherwise. While this may appear to be a relatively silly idea, such an initiative would have prevented Oracle's investigators from uncovering Microsoft covert antitrust activities.

5.7 Mitigating the Damage of Social Engineering Attacks [25-27]

While the best practices mentioned above are likely to prevent many social engineers from successfully breaching an organization, Kevin Mitnick laments that you simply cannot stop a determined social engineer because "You can't go and download a Windows update for stupidity... or gullibility". As a result, a proper security plan should incorporate safeguards designed to mitigate the extent of damage that a social engineering attack could cause.

5.7.1 Segregation of access

By ensuring that users only have access to the information and systems that they absolutely require for their jobs, the organization can limit the amount of damage that a social engineer with access to the system could cause.

5.7.2 Maintain access logs

By ensuring that the company retains an access log, it will be possible for the company to find out what the attacker was able to access before the company was able to cut off his or her access. This will allow the company to make informed decisions about the extent of the damages that the company is facing and whether any immediate response is necessary. For example, if a pharmaceutical company notices that research for a particular unpatented drug was stolen, it can work with its lawyers on defending that particular drug from being infringed upon.

5.7.3 Ensure that backups occur regularly

While some intruders may be content to simply steal as much information as possible from a target company, others may be more interested in simply harming the company by deleting the company's data. Companies need to perform backups on a regular basis and ensure that an intruder on the network would not be able to also destroy the backup. An example of an attack that could have been mitigated by backup procedures occurred on March 31, an ex-employee of a data storage company was able to break into the servers and delete over 304 gigabytes of data that could not be recovered.

5.7.4 Automatically revoke user privileges if suspicious activity is detected

When an intruder enters a system, he or she will immediately search for the files or applications of interest by browsing around the computer's directories or performing searches of the drive and transfer as much information as possible back to his or her own computer. Using anomaly-based intrusion detection systems, companies can automatically monitor the normal levels of disk and network usage and shut down a suspicious account's access within seconds.

5.8 References:

1. 419 Eater. What is the '419' scam? 20 June 2012. 20 June 2012<http://www.419eater.com/html/419faq.htm>.
2. Boritz, Efrim. Information System Security and Availability. 1997. 20 June 2012

<http://accounting.uwaterloo.ca/ccag/6CHAP97.htm>.

3. Bort, Julie. The 10 Most Outrageous Stories about Larry Ellison. 17 January 2012. 20 June 2012 < http://www.businessinsider.com/the-10-most-outrageous-larry-ellison-stories-2012-1?op=1 >.

4. Damle, Pramod. Social Engineering: A Tip of the Iceberg. 2002. 20 June 2012 <http://www.isaca.org/Journal/Past-Issues/2002/Volume-2/Pages/Social-Engineering-A-Tip-of-the-Iceberg.aspx>.

5. Farivar, Cyrus. Stuxnet expert calls US the "good guys" in cyber-warfare. 6 June 2012. 20 June 2012 <http://arstechnica.com/tech-policy/2012/06/stuxnet-expert-calls-us-the-good-guys-in-cyber-warfare/>.

6. Gedda, Rodney. Hacker Mitnick preaches social engineering awareness. 21 July 2005. 20 June 2012 <http://www.computerworld.com.au/article/136508/hacker_mitnick_preaches_social_engineering_awareness/>.

7. Gragg, David. A Multi-Level Defense Against Social Engineering. 1 January 2003. 20 June 2012 <http://www.sans.org/reading_room/whitepapers/engineering/multi-level-defense-social-engineering_920>.

8. Hadnagy, Christopher J. Defcon 18: Social Engineering Capture the Flag Results. 2010. 20 June 2012 <http://www.social-engineer.org/resources/sectf/Social-Engineer_CTF_Report.pdf>.

9. International Intelligence Limited. Social Engineering. 8 December 2008. 20 June 2012 <http://www.hg.org/article.asp?id=5778>.

10. Laflotte, Duane. The Dark Art of Social Engineering. 6 October 2005. 20 June 2012 <http://www.informit.com/articles/article.aspx?p=417272&seqNum=3>.

11. Langner, Ralph. How did Stuxnet reach its target? CBS Online. 4 March 2012.

12. Lineberry, Stephen. The Human Element: The Weakest Link in Information Security. November 2007. 20 June 2012 <http://www.journalofaccountancy.com/Issues/2007/Nov/TheHumanElementTheWeakestLinkInInformationSecurity.htm>.

13. Gragg, David. A Multi-Level Defense Against Social Engineering. 1 January 2003. 20 June 2012 <http://www.sans.org/reading_room/whitepapers/engineering/multi-level-defense-social-engineering_920>.

14. Hadnagy, Christopher J. Defcon 18: Social Engineering Capture the Flag Results. 2010. 20 June 2012 <http://www.social-engineer.org/resources/sectf/Social-Engineer_CTF_Report.pdf>.

15. International Intelligence Limited. Social Engineering. 8 December 2008. 20 June 2012 <http://www.hg.org/article.asp?id=5778>.

16. Laflotte, Duane. The Dark Art of Social Engineering. 6 October 2005. 20 June 2012 <http://www.informit.com/articles/article.aspx?p=417272&seqNum=3>.

17. Langner, Ralph. How did Stuxnet reach its target? CBS Online. 4 March 2012.

18. Lineberry, Stephen. The Human Element: The Weakest Link in Information Security. November 2007. 20 June 2012 <http://www.journalofaccountancy.com/Issues/2007/Nov/TheHumanElementTheWeakestLinkInInformationSecurity.htm>.

19. Mann, Ian. Hacking the human social engineering techniques and security countermeasures. Aldershot: Ashgate, 2008.

20. McMillan, Robert. The Pwn Plug is a little white box that can hack your network. 3 March 2012. 20 June 2012 <http://arstechnica.com/business/2012/03/the-pwn-plug-is-a-little-white-box-that-can-hack-your-network/ >.

21. Mitnick, Kevin. The Art of Deception. Indianapolis: Wiley, 2002.

22. Nemani, Purna. Hacker Deleted Entire Season, TV Station Says. 31 March 2011. 20 June 2012 <http://www.courthousenews.com/2011/03/31/35406.htm>.

23. Office of Inadequate Security. WHMCS victim of social engineering; over 500,000 client records stolen, deleted from server, and dumped publicly. 22 May 2012. 20 June 2012 <http://www.databreaches.net/?p=24284>.

24. PammingSodom. Nigerian prince wants my help? 20 June 2009. 20 June 2012 <http://answers.yahoo.com/question/index?qid=20090807194219AALXAcZ>.

25. Panda Security. Crimeware: the silent epidemic. 2011. 20 June 2012 <http://www.pandasecurity.com/homeusers/security-info/types-malware/crimeware/>.
26. Paul, Mano. Phishing: Electronic Social Engineering. 1 January 2009. 20 June 2012 <http://www.certmag.com/read.php?in=3594>.
27. Peltier, Thomas R. Social Engineering: Concepts and Solutions. 20 June 2012. 20 June 2012 <http://www.infosectoday.com/Norwich/GI532/Social_Engineering.htm>.

CHAPTER 6

NETWORK SECURITY MECHANISMS

6.1 Introduction [1-3]

This chapter discusses network security. Computer networks are analyzed and discussed in the previous related chapters. Based on the previous terminology, this chapter describes the security issues in computer networks. A broad definition of network security can be constructed by defining its two components, security and networks. Security is a term which has been given a wide variety of definitions. According to dictionary definitions, security is the freedom from danger or anxiety. Among others, security is also defined as: (1) A situation with no risk, with no sense of threat, (2) The prevention of risk or threat, and (3) Assurance, sense of confidence and certainty.

6.1.1 Security in Information Technology

In traditional information theory, security is described through the accomplishment of some basic security properties, namely Confidentiality, Integrity and Availability of information. Confidentiality is the property of protecting the content of information property of protecting information from non-authorized, temporary or permanent withholding of information from all users other than those intended by the legal owner of the information. The non-intended users are generally called unauthorized users. Integrity is the property of protecting information from alteration by unauthorized users. Availability is the Other basic security properties are the Authentication and the Non-repudiation. Authentication is divided into peer-entity authentication and data-origin authentication. Peer-entity authentication is the property of ensuring the identity of an entity (also the information. Finally, non-repudiation–is the property of ensuring that principals that have committed to an action cannot deny that commitment at a

later time. Detailed known as ‘subject’), which may be a human, a machine or another asset such as a software program. Data- origin authentication is the property of ensuring the source of treatment of security properties can be found in several security standards, such as the ISO/IEC 7498-2 and the ITU-T X.800 security recommendation.

In a practical Information Technology approach, security involves the protection of information assets. In a traditional Information Technology risk analysis terminology, an asset is an object or resource, which is “worthy” enough to be protected. Assets may be:

- physical (e.g. computers, network infrastructure elements, buildings hosting equipment),
- data (e.g. electronic files, databases) or
- software (e.g. application software, configuration files)

The information assets must be protected by security threats. A security threat is any event that may harm an asset. When a security threat is realized, then an IT system or network is under a security attack. The attacker or threat agent is any subject or entity that causes the attack. Of course an asset may be threatened by various threats and each threat has a different threat level against each asset. An example of a threat agent in a computer network is a malicious outsider (external user) who attempts to bypass security measures and access the network. A threat which may be caused by such an attacker is unauthorized access to network resources.

A security vulnerability is any characteristic in a system, which makes one or more assets more vulnerable to threats. In the above example of a threat, a security vulnerability which exposes the system to unauthorized access is the lack or the misconfiguration of access controls. If access to the network is not properly controlled with well configured mechanisms, then it will be easier for a possible attacker to gain unauthorized access into the system and the network is more vulnerable to intrusion attacks.

The impact of the threat measures the magnitude of the loss that would be caused to the asset or asset owner if the threat were realized against it. The magnitude of loss is closely related with the

operational or business value of the attacked asset. The combination of threats, vulnerabilities and assets provides a quantified and/or qualified measure of the likelihood of threats being realized against assets, as well as the impact caused due to the realization of a threat. This measure is known as the security risk.

The protection of assets can be achieved through several security mechanisms. A security mechanism is any type of measure, (technical, procedural, legal etc.) which may protect an asset from security threats, reduce their vulnerability and more generally reduce the level of security risks. A security mechanism may be:

- Preventive, if its goal is to prevent the realization of a security attack. Such mechanisms mainly reduce the threat–level of security attacks.
- Detective, if its goal is to detect a security attack as fast as possible and thus restrict the consequences of the attack. These mechanisms mainly reduce the vulnerability level of security attacks.
- Recovery, if its goal is to recover the system after a security attack in the shortest possible time. These mechanisms mainly reduce the impact level of security attacks.

Thus, the security mechanisms provide capabilities that reduce the security risk of a system. Note that system and network security does not rely solely on technical security mechanisms. In almost every information system and network, procedural and organizational measures are generally required in addition to technical mechanisms, in order to accomplish the desired security goals.

6.1.2 Computer Networks

A computer network or simply a network is a collection of connected computers. Two or more computer systems are considered as connected, if they can send and receive data from each other through a shared access medium. The communicating entities in a computer network are

generally known as principals, subjects or entities. These principals can be further divided into users, hosts and processes.

- A user is a human entity, responsible for its actions in a computer network.
- A host is an addressable entity within a computer network. Each host has a unique address within a network.
- A process is an instance of an executable program. It is used in a client/server model, in order to distinguish between the client and the server processes.
- A client process is a process that makes requests of a network service.
- A server process is a process that provides a network service, for example as daemon process running continuously in the background on behalf of a service
- In order to formalize the way that networking is performed, network reference models have been developed, which group similar functions into abstractions known as layers. Each layer's functions can communicate with the same layer's functions of another network host. On the same host, the functions of a particular layer have interfaces to communicate with the layers bellow and above it. This abstraction simplifies and properly defines the necessary actions for networking.
- The International Standards Organization (ISO) Open Systems Interconnection (OSI) Reference Model defines seven network layers, as well as their interfaces. Each layer depends on the services provided by its intermediate lower layer all the way down to the physical network interface card and the wiring. Then, it provides its services to its immediate upper layer, all the way up to the running application. The network layers in the ISO/OSI Reference Model are the following (from the lowest to the highest): 1) The Physical Layer, 2) The Data Link Layer, 3) The Network Layer, 4) The Transport Layer, 5) The Session Layer, 6) The Presentation Layer and 7) The Application Layer. The X.200 recommendation of the ITU-T is aligned with the ISO/IEC 7498-1 standard. More details on network reference model can be found in Models and Layered Protocol Organization.

- Each reference model needs a suite of network protocols in order to implement the functions of each layer. Generally, a network protocol is a well-defined specification which allows network hosts to communicate in a particular and predefined way. From a point of view, protocols define the "syntax" of the communication. By properly combining protocols in protocol stacks, the layers of network reference models can be implemented and allow network communication. It needs to be noted that not all protocol suites include all the seven layers of the ISO/OSI model. The most popular protocol suite, the Transmission Call Protocol/ Internet Protocol (TCP/IP), has five layers. There are no Presentation and no Session layers; the functions of these layers are incorporated in the layers above and below. Although detailed description of the TCP/IP is given elsewhere, it is important to understand how it works, in order to understand network security.
- A network is considered as a wired or fixed network if the access medium is some kind of physical cable connection between the computers, such as a copper cable or a fiber optic cable. On the other hand, a network is considered as a wireless network, if the access medium relies on some kind of signaling through the air, such as RF communication. A network can also be divided according to its geographical coverage. Depending on its size, a network can be a Personal Area Network (PAN), a Local Area Network (LAN), a Metropolitan Area Network (MAN) or a Wide Area Network (WAN).

6.1.3 Telecommunication Networks

A telecommunication network is a collection of connected links, which allow messages to pass from one part of the network to another, through the intermediate links. In the general term, computer networks may be considered as telecommunication networks. However, the term telecommunication networks are basically used to describe telephone networks. These include fixed networks, such as the Public Switched Telecommunication Network (PSTN), which is globally used for wire-line telephone communications. They also include mobile networks, such as the Global System for Mobile communications (GSM), which is the most common cellular phone network, or the next generation Unified Mobile Telecommunication System (UMTS)

network. The GSM is considered as second-generation (2G) mobile network, while UMTS is considered as a third generation (3G) mobile communication network.

The General Packet Radio Service (GPRS) is a service that provides packet radio access for GSM users as a step towards the third-generation telecommunication networks. GPRS reserves radio resources only when there is data to be sent. In this way it enables the efficient provision of a variety of packet-based services to the mobile subscribers of second generation networks. GPRS attempts to reuse the existing GSM network elements as much as possible, in order to effectively build a packet-based mobile cellular network.

UMTS is a realization of 3G networks, intending to establish an integrated system that supports different operating environments.–Users have seamless access to a wide range of new telecommunication services, such as high data rate transmission for high -speed Internet applications, independently of their location. Thus, mobile networks are a natural extension of the wired Internet computing world, enabling access for mobile users to multimedia services that already exist for non-mobile users and fixed networking.

Security in telecommunication networks has in general the same requirements as in computer networks, concerning the required security services and mechanisms, which are discussed in the following sections. However, the security design in telecom networks shall take into consideration several aspects and differences, such as the closed nature of telecom networks in comparison with the open nature of the Internet, the wireless access of mobile telecommunication networks and the end-user mobility, the particular security threats, the type of information to be protected, and the complexity of the network architecture. The radio transmission is by nature more vulnerable to eavesdropping, than fixed-line transmission. The user mobility and the universal network access certainly provoke security treats. As the telecommunication networks are converging towards IP-based communications (e.g. in the GPRS and UMTS) and as computer (information) and telecommunication networks are getting more and more interconnected, a holistic approach towards network security must be followed.

6.1.4 The Goals of Network Security

Regardless of the access medium and the coverage of a network, network security can be considered through the achievement of two security goals: computer system security and communication security.

- The goal of computer systems security is to protect information assets against unauthorized or malicious use, as well as to protect the information stored in computer systems from unauthorized disclosure, modification or destruction.
- The goal of communications security is to protect information during its transmission through a communication medium, from unauthorized disclosure, modification or destruction.

In order to achieve the goals of network security in any network, the following steps must be followed:

1. Define the assets to be protected and the perimeter of the network. Before implementing any security measures, the assets of the network must be identified and assessed. Furthermore, the perimeter of the network to be protected must be defined, in order to distinguish the internal or private network from the external or unreliable network.

2. Define the possible security threats and attacks. After the network assets and the network perimeter have been defined, the possible security attacks that threat the network must be defined and evaluated. This will help in focusing on the protection from the most possible threats. In this process it is very important to consult specialized Internet sites that focus on network security and security threats, either of proprietary products or from security threats and vulnerabilities databases.

3. Evaluate the security risks and define the desired security level. The following step is to evaluate the examined threats in conjunction with the existing vulnerabilities and assets. This can be performed by using a risk analysis methodology. Then, after the risks against network security have been identified, the desired security level must be defined, in order to set up the suitable security measures.

4. Define security policies that formally set up the desired security level. The desired security level must then be formalized through network security policies. These policies are a formal way to define what security services must be provided, in order to reach the network security goals and to reduce the risk to the desired and acceptable level.
5. Define the security services and implement the proper security mechanisms. The security services define what security properties must be maintained in each part of the network, such as authentication and access control. The security mechanism defines the way that will implement the functionality of the defined security services. More details about network security services and mechanisms are provided in the following sections. Note however that the apart from the technical security mechanisms, other non-technical security measures are also defined in order to achieve the desired security level that is formally described in the security policies. These non-technical measures are mostly security procedures.
6. Periodically assure that the proper security policies, services and mechanisms are in place. Although the security threats may have been properly recognized and security policies may enforce the desired security level with security mechanisms and controls, it is important to periodically assure that everything is set up correctly. Problems may arise due to new security threats and vulnerabilities, new security needs or attenuation of the existing security mechanisms. The period that each of the above must be examined differs, since due to technology changes it is usually required to examine the security mechanisms more frequently than the security policies or services, or the desired security level.

6.2 Security Services and Security Mechanisms [4-8]

X.800 defines a security service as a service provided by a protocol layer of communicating open systems, which ensures adequate security of the systems or of data transfers. Also the RFC 2828 defines security services as a processing or communication service that is provided by a system to give a specific kind of protection to system resources. Security Services implement security policies and are implemented by security mechanisms.

6.2.1 Authentication: The assurance that the communicating entity is the one that it claims to be.

- **Peer Entity Authentication:** Used in association with a logical connection to provide confidence in the identity of the entities connected.
- **Data Origin Authentication:** In a connectionless transfer, provides assurance that the source of received data is as claimed.

6.2.2 Access Control: The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).

6.2.3 Data Confidentiality: The protection of data from unauthorized disclosure.

- Connection Confidentiality: The protection of all user data on a connection.
- Connectionless Confidentiality: The protection of all user data in a single data block
- Selective-Field Confidentiality: The confidentiality of selected fields within the user Data on a connection or in a single data block.
- Traffic Flow Confidentiality: The protection of the information that might be Derived from observation of traffic flows.

6.2.4 Data Integrity: The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).

- Connection Integrity with Recovery: Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.
- Connection Integrity without Recovery: As above, but provides only detection without recovery.
- Selective-Field Connection Integrity: Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of

determination of whether the selected fields have been modified, inserted, deleted, or replayed.

- Connectionless Integrity: Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.
- Selective-Field Connectionless Integrity: Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.

6.2.5 Nonrepudiation: Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.

- Nonrepudiation, Origin: Proof that the message was sent by the specified party.
- Nonrepudiation, Destination: Proof that the message was received by the specified party.
- Security Mechanisms: The security mechanisms are divided into those that are implemented in a specific protocol layer and those that are not specific to any particular protocol layer or security service. X.800 distinguishes between reversible encipherment mechanisms and irreversible encipherment mechanisms. A reversible encipherment mechanism is simply an encryption algorithm that allows data to be encrypted and subsequently decrypted. Irreversible encipherment mechanisms include hash algorithms and message authentication codes, which are used in digital signature and message authentication applications.

6.2.6 Specific Security Mechanisms

Incorporated into the appropriate protocol layer in order to provide some of the OSI security services.

Encipherment: The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.

Digital Signature: Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery.

Access Control: A variety of mechanisms that enforce access rights to resources.

Data Integrity: A variety of mechanisms used to assure the integrity of a data unit or stream of data units.

Authentication Exchange: A mechanism intended to ensure the identity of an entity by means of information exchange.

Traffic Padding: The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.

Routing Control: Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.

Notarization: The use of a trusted third party to assure certain properties of a data exchange.

6.2.7 Pervasive Security Mechanisms

Mechanisms that are not specific to any particular OSI security service or protocol layer.

Trusted Functionality: That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).

Security Label: The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.

Event Detection: Detection of security-relevant events.

Security Audit Trail: Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.

Security Recovery: Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.

6.3 Personally identifiable Information (PII) [9-12]

PII, according to the U.S. Office of Management and Budget, is any information that can be used to uniquely identify, contact or locate an individual, or can be used with other sources to uniquely identify a person. It consists of a broad range of information that can identify individuals, including dates of birth, addresses, driver's license numbers, credit card numbers, bank account numbers, health and insurance records, and much more. Unless your organization keeps no payroll-related data about its employees, it has PII it needs to protect.

While most adults are careful about disclosing their personal information, this issue is particularly sensitive for organizations that have information on minors, such as schools, councils and medical services. It becomes incumbent on the holder of that PII to be vigilant about its use and access. According to the U.S. General Accounting Office, 87% of the U.S. population can be uniquely identified using only gender, date of birth and ZIP code. So it's not just the most obvious types of PII, like credit card numbers, that require protection.

Examples of PII

• First or last name (if common)
• Date of birth
• Country, state or city of residence
• Credit card numbers

• Immunization history/medical records
• Age
• Telephone numbers
• Email addresses
• Gender
• Race
• Criminal record

Consequences of not protecting PII

Regardless of how the data is lost, the cost of a data breach can be huge. Fines are one of the most widely-known consequences of losing personal data, and they can be very expensive (e.g., up to $1.5 million per year in the case of a breach of healthcare records in violation of the Health Insurance Portability and Accountability Act [HIPAA] regulation or up to £500,000 from the UK Information Commissioner). However, the consequences extend much further and include reputation damage, loss of customer trust, employee dissatisfaction and attrition, and clean-up costs following the breach. Examples include:

- Hartland Payment Systems committed $8 million to settle lawsuits following a data
  breach which compromised 130 million credit and debit cards
- Health Net of the Northeast Inc. agreed to pay for two years of credit-monitoring for 1.5 million members whose details were on a lost hard drive
- Sony provided free services to customers affected by their 2011 data breaches to help them protect against identify theft.

The three states of data

- Data in use is data on endpoints being used by employees to do their jobs.
- Data at rest is information stored on endpoints, file servers and information repositories like Exchange servers, Sharepoint and web servers.

- Data in motion is data sent over networks. Organizations must ensure they they consider data in all three states when protecting their PII.

5 steps to acceptable use policy

There are five key steps every organization must take to begin the process of preventing data loss:

- Identify PII your organization must protect
- Prioritize PII
- Find where PII is located
- Create an AUP
- Educate your employees about your AUP

How do you find the PII in your organization? It may be in multiple places, redundant on servers, laptops, PCs and removable media. Thinking about the data in each of its three states (see Table 6.1) will help you identify where it's located.

Once you've found the PII, you need to define what your organization's AUPs are for accessing and using it. AUPs will vary from organization to organization, but should accomplish three goals:

- Protect PII data
- Define who can access PII
- Establish rules for how authorized employees can use PII

The AUPs you develop will only be effective if your employees feel they have a part to play in protecting your PII. Comprehensively educating employees is a critical and often overlooked step. Deliver copies of AUPs to employees, offer training sessions and have them sign a statement acknowledging they will abide by the policies. This will make every employee an

active participant in the enforcement of AUPs, and the organization-wide effort to prevent data loss and the loss of PII.

Table 6.1: Five rating criteria to determine what data needs to be protected most

Distinguishability	Look for data that by itself can identify a unique individual.
Aggregation	Look for two or more pieces of data that when combined can identify a unique individual.
How PII is stored, transmitted, used	• Frequently transmitted over networks • Stored redundantly on servers or portable devices • Used by many people in the organization
Compliance	Your organization must comply with regulations and standards for protecting PII. Which ones will depend where you are based and scope of work. However these may include: • **Payment Card Industry Data Security Standards** (PCI DSS) (International) – setting out requirements for data security when handling card payments • **Data Directive** (EU) – requiring the safe storage using data loss prevention technology of data generated in connection with public electronic communication • **HIPAA and HITECH ACT** (U.S.) – enabling fines of up to $1.5 million per year for a breach of healthcare records • **Criminal Justice and Immigration Act** (UK) – giving the Information Commissioner power to levy fines of up to £500,000 for data breaches There are also a large number of data security regulations applicable at regional or state level. If you work in a geography covered by such legislation you should understand the implications for your organization.
Ease of access	Decide if the PII: • Is easily accessed by any employee • Can be copied, sent and saved without restriction • Is available for use by HR for employee management or by staff • Is not protected by PINs or passwords before being accessible by staff

Choosing the right solution to protect PII

After you've identified your organization's PII and adopted AUPs for its safe use, it's time to look at how to secure your network, endpoints, other devices and applications. Strong, system-level security can prevent accidental data loss and stop malicious threats before they harm your organization, while ensuring the right employees have access to the data they need to do their jobs within established AUPs. There is no silver bullet to accomplish these goals. Rather, it requires a combination of technologies for defense-in-depth or a multilayer security strategy.

6.4 References

1. Douligeris C. and Serpanos D. (eds) (2006). Network Security: Current Status and Future Directions.

2. Wiley – IEEE. [This book is a collection of surveys related with all the aspects of network security].

3. Douligeris C. and Mitrokotsa A. (2004). DDoS –attacks and defense mechanisms: classification and state-of-the-art. Computer Networks (44), 643-666, Elsevier. [It provides a survey on several attacks and defense mechanisms for Distributed Denial of Service attacks in networks].

4. Menezes A., J,van.Oorschot, P.C,Vanstone S. A. (1997). Handbook of Applied Cryptography, CRC Press. [This handbook is a major source of information on applied cryptography].

5. Peltier T.R. (2001). Information Security Risk Analysis, Auerbach Publications, 2001. [This book describes methodologies on information risk analysis and security policies for systems and networks].

6. International Standardization Organization (1994) . Information Processing Systems – Open Systems Interconnection – Part 1: Basic Reference Model, ISO/IEC 7498-1: 1984, also ISO/OSI 7498-1: 1994. [This ISO standard describes the basic reference model for networks].

7. International Standardization Organization (1989). Information Processing Systems – Open Systems Interconnection – Part 2: Security Architecture, ISO/IEC 7498-2: 1989. [This ISO standard describes the security architecture for OSI networks].

8. International Telecommunication Union (1991). Security Architecture for Open Systems Interconnection for CCIT Applications, Recommendation ITU-T X.800, 1991.

9. International Telecommunication Union (1994). Information Technology – Open Systems Interconnection – Basic Reference Model: The basic model, Recommendation ITU-T X.200, 1994.

10. International Telecommunication Union (1995). Information Technology – Open Systems Interconnection – Lower Layers Security Model, Recommendation ITU-T X.802, 1995.

11. International Telecommunication Union (1994). Information Technology – Open Systems Interconnection

12. Upper Layers Security Model, Recommendation ITU-T X.802, 1994.