CHAPTER 7

Information Security Issues in Cyberspace

Edward Skoudis

THIS CHAPTER examines attacks and defenses associated with Internet technologies, defined as systems that are either directly or indirectly connected to the global Internet, relying on the transmission control protocol (TCP)/Internet protocol (IP) communications suite.[1] As described in chapter 4, cyberspace relies on a vast collection of underlying technologies. Security issues associated with these technologies will be analyzed from an attack and a defense perspective to provide a better understanding of the technological options available to both attackers and defenders and of various policy issues associated with security technologies. This chapter focuses first on the most common attacks today and those that will likely pose a significant threat for the foreseeable future. It then discusses the most widely available defensive technologies, how they thwart attacks, and various public policy issues associated with each defense.

With the increasing convergence of technology, Internet-based systems are among the fastest growing components of cyberspace. Phone, television, radio, and financial transactions increasingly rely on the Internet. This chapter focuses on Internet attacks and defenses; analogous concepts, if different technologies, can be applied to other cyberspace elements not associated with the Internet. For example, an attacker might deliberately cause widespread Web site outages via a denial-of-service flood against various Internet components (comparable to jamming or overwhelming a radio communications channel to block the exchange of information). With these Web sites down and the links connecting them to the Internet clogged with flood traffic, other ancillary services that use the same network, such as Internet telephony, streaming radio, and television services, could likewise be impacted.

Internet Attacks

Various forms of cyberspace attacks occur every day on the Internet, ranging from virus infections to massive automated credit card thefts involving millions of consumers. In this chapter, attacks are categorized as small- or large-scale.[2] Small-scale attacks may cause limited damage to a relatively small number of people (from one to many thousands). For example, most viruses today infect hundreds or thousands of people and represent a nuisance that can typically be cleaned quickly with an antivirus program. Examples of spyware, programs that surreptitiously undermine a user's privacy, fall into this small-scale category as well. While spyware infections in their totality may affect 100 million machines or more, each individual falls into the small-scale range.

Large-scale attacks are those that have potential impact on millions of users, possibly disabling pieces of the Internet so users within a given country or geographic region cannot get access at all, or compromising the accounts of millions of users to steal sensitive information for committing fraud.

Small-scale Attacks

Attacks on the small scale are far more common than large-scale attacks. Throughout the 1990s, the dominant threats to systems connected to the public Internet came from hobbyists experimenting with viruses and other malicious code and small-scale criminals looking to cash in on isolated cyber crime. More insidious threats may have existed, such as organized crime or nation-states engaging in Internet-based reconnaissance, but these were not a major issue at the time. However, around 2002, the threat quickly changed: organized crime groups began to use a variety of cyber crime techniques, including shady advertising schemes, online scams, and other attacks, to make money. Today, worldwide criminal businesses are based on cyber crime. This section summarizes the most common of these attacks.

Spyware. Many contemporary small-scale attacks on the Internet are carried out using spyware, which focuses on gathering information from and about users and is installed on a user's machine without notice or consent. Very aggressive companies and cyber criminals use spyware to gather information about users, often treading in a gray area of legality. Some online advertisers, gaming companies, pornographic Web sites, and others stray into violating users' privacy with spyware as they carry out various money-making practices. Spyware is installed on an end-user's desktop or laptop computer by an attacker either exploiting software vulnerabilities on the machine or tricking the user into installing the spyware by bundling it with some other program. Once installed, such spyware might track user activity, inject advertisements into a user's browsing session, or even steal confidential information or documents.

The most benign forms of spyware track user interactions across a series of Web sites. Advertisers and other organizations have a financial interest in determining user Web surfing habits so they can customize advertisements to maximize sales of a product or service. The simplest form of spyware is a tracking cookie, a small piece of data that a Web site pushes to a user's browser for storage on the user's machine. Whenever the browser accesses that Web site again or other sites affiliated with the spyware company, the cookie is presented back to the associated site, and by this means the user's access of all affiliated sites can be tracked. Tracking cookies are only transmitted to and from Web sites affiliated with the site that originally sent the cookie to the user's browser and can only be used to track user actions between affiliated sites, not all of the user's surfing outside of the affiliate sites.

To get around these limitations, more aggressive spyware companies may install software on a user's machine that tracks all Web surfing activities to any sites. Instead of relying on a cookie, the spyware companies install software on the user's machine that watches the user's activities. The data entered into Web sites and the date and time of each interaction may be transmitted to the spyware company across the Internet. This stolen data may include credit card numbers and passcodes, which a thief can use to make illicit charges against the victim's account.

Other forms of spyware make use of duplicitous advertising and redirection to try to cash in on Internet-based advertising, a multibillion-dollar-a-year business. The advertiser typically pays the referrer (the Web site that forwarded a user to the ad Web server) a fraction of a cent or more for each ad displayed in a user's browser. If the user clicks through on the ad, the rate of pay is higher. These small payments aggregate to billions of dollars. Spyware purveyors may

surreptitiously install software on users' computers that fetches ads from the Internet and presents them on the browser or via a popup ad. Some of the more aggressive spyware customizes ads to appear inside the content of other Web sites. For example, spyware may wait for a user to perform a search at a popular search engine such as Google, Yahoo!, or MSN Search. Before the results are displayed in the browser window, they are edited by the local spyware on the victim's machine. This spyware might reorder search results, or inject its own ads inline onto the browser screen. Ads that appear to come from Google might actually be generated by locally installed spyware on a machine.

Some spyware redirects browsers to affiliates of the spyware creator. When the user tries to access a major search engine, for example, the spyware may redirect the browser to a search engine that displays advertisements for which the spyware creator is paid. The user's browsing is hijacked and directed to locations chosen by the spyware creators. For example, when a user tries to access a major retailer online, their browser may automatically jump to a different retailer associated with the spyware creator.

Some spyware focuses on stealing information from users by searching a hard drive for sensitive or proprietary documentation that it sends to the attacker. One of the most insidious forms is a keystroke logger, which records everything typed into the keyboard of an infected machine, such as account numbers typed into a financial services Web site login page, and transmits the information to an attacker.

Bots and rootkits. An attacker may use *bot* software—the name is derived from the word *robot*—to get complete control of an infected machine across the Internet. Such an attacker is sometimes referred to as the *bot-herder*. A collection of bot-infected machines is known as a *botnet* and may range in size from a few thousand machines to many millions of systems.

By controlling an infected system, bots can be used as a form of spyware, tracking user Web surfing habits, stealing files, or logging keystrokes. Bot-herders can also do more damage by harnessing the computing and network resources of a botnet to achieve other goals. For example, software distributed to all systems in a botnet could operate as a distributed supercomputer to crack encryption keys or passwords many thousands of times faster than a single computer could. Attackers can send spam email via the botnet at a much faster rate than they could with only a single computer. Bots can launder the Internet source address of an attacker who configures bots as anonymous packet forwarders, which strip information from packets directed through them. Forwarding information through a dozen or more anonymizing bots makes tracking criminal activity back to an individual attacker much more difficult for investigators.

Bot-herders interact with and control their botnets in a variety of ways. They may, for example, place commands on a Web page, either one maintained by the attacker or a public one such as a social networking site. Alternatively, they may control their botnet using an Internet Relay Chat (IRC) server, which is designed for chatting but also is useful in sending messages to large numbers of systems. Such methods expose attackers to a single point of failure: if an IRC server, social networking Web site, or user profile controlling the bots disappears due to the work of diligent investigators, the botnet becomes headless. Thus, some bot-herders have turned to distributed peer-to-peer communications protocols, similar to the Internet telephony service Skype. Instead of relying on a centraliz- ed point of control, individual nodes seek and find other nearby nodes in peer- to-peer networks. Diligent investigators may shut down hundreds or thousands of bots in the network, but they will still have disrupted only a small

fraction of the botnet.

Large-scale removal of bots would undermine the attacker's business model, so bots are increasingly bundled with rootkits, software that alters the operating system to lie about and hide the attacker's files, programs, and network communications, thus concealing the attacker's presence on a machine. Some bots prevent antivirus and personal firewall protections of the system from working. Bots hidden by rootkits are widely used by cyber criminals today.

Spam and phishing. Unsolicited commercial email, commonly known as spam, makes money. Millions of email messages daily advertise pharmaceuticals, software, and consumer electronics goods. Most spam is merely an annoyance and can be controlled with antispam email filters, which look for common patterns associated with spam. However, the infrastructure created by spammers to send email is increasingly being used to commit scams, fraud, and cyber crime.

Phishing attacks are one of the most common forms of online scam. They typically involve millions of emails, apparently from a legitimate company such as an online bank or merchant, announcing a problem with the recipient's accounts. The emails try to dupe users into clicking on a link that appears to point to a legitimate business Web site but actually takes the user to an imposter site controlled by the attacker and designed to resemble the e-commerce site. The site asks the user for a login name and password or other account information, which the attacker's software retains for fraud and criminal use. There are over 100,000 phishing Web sites on the Internet, and millions of phishing emails are sent every day.[3] Phishers are also increasingly using email that appears to come from taxing authorities, government benefits organizations, and other government agencies.

Increasingly common is *spear-phishing*, in which phishers choose a particular group of target recipients either more likely to succumb to a scam, such as the elderly, or who have access to sensitive information, such as some military personnel.

Spam and phishing email distribution infrastructures have become more resistant to suppression by law enforcement. From 1999 to 2003, thwarting schemes that used a dozen or so email servers to send spoofed email was quite straightforward. Once investigators detected the large number of bogus email messages originating from the spammer's mail servers, they could add each server's address to a blacklist to be blocked by legitimate mail servers around the world. Once on the blacklist, the attacker's email servers would be shut down, forcing the attackers to interrupt their business as they moved to different servers.

Starting around 2003, however, spammers and phishers began using mail servers operated by others that had been configured inappropriately to act as mail relays. Mail relays accept email and forward it to other mail servers. The attackers would send their spam through these innocent but poorly configured third-party mail servers, which in turn would forward the spam to other mail servers and to its recipients. Again, the Internet community did a reasonable job of handling this problem, typically through blacklists. When the third-party mail server being used by the attacker was added to a blacklist, mail servers around the world would start dropping messages from all users on that server, regardless of whether the mail was legitimate or not. The legitimate users of the third-party mail server would complain about the dropped messages, and the company would quickly turn off mail relaying, a function seldom needed on Internet-accessible mail servers today.

Attackers responded with further refinements, such as relying on botnets of 100,000 or

more systems instead of a dozen mail servers or 100 mail relays. Disabling or even blacklisting such an enormous number of systems is impractical. Because consumer end-user computer systems almost never need to act as mail servers or relays, many Internet service providers (ISPs) block any normal mail server traffic on their networks that is associated with end-user computer systems. Unfortunately, most ISPs outside of the United States and Europe do not implement such defenses, due to the costs and lack of perceived direct benefit to the ISP itself. Furthermore, such defenses only apply to consumer systems, because most enterprise networks require mail server traffic.

Credit card fraud and identity theft. The Internet is a major vehicle for the theft and illicit use of consumer credit card accounts. Attackers steal card numbers in a variety of ways. Bot-based keystroke loggers and phishing attacks are two methods for stealing such information from end-users one at a time. Other attackers focus on sources of aggregated information, such as companies that accept credit cards. Every week, new incidents are reported involving data breaches that affect 50,000 to 1 million or more users. Attackers can sell credit card account information on the black market; card-selling exchanges on the Internet are hosted in a variety of countries, especially in Eastern Europe. Other information is also useful in fraud, such as credit histories, mortgage information, and Social Security numbers. By exploiting Web site vulnerabilities, poorly secured wireless network access, or improperly configured corporate networks, attackers frequently infiltrate systems to steal such information from companies and government agencies in order to commit fraud.

Credit card fraud usually starts with a small charge against the account to make sure that it is working. Next, the attackers continue increasing the amount they charge until the activity is detected by automated antifraud systems operated by credit card companies and issuing banks. Then the account is usually deactivated and a new card issued to the consumer. The consumer suffers the inconvenience of waiting for a new credit card, while the back-end banks typically write off the fraudulent charges.

Some attackers establish complete dossiers of information about individuals. Cyber crime organizations compile vast databases of user information from multiple breaches over time, with fields including Social Security numbers, multiple credit cards, and mortgage loan information. Given enough information, attackers can engage in identity theft. Posing as that user, criminals may acquire new credit cards or even mortgages, destroying the victim's credit in the process.

Corporate information theft. Cyber criminals may seek to steal corporations' business operations data, including trade secrets, business strategies, and other sensitive corporate information. Using the same technical means used to steal credit cards, attackers may compromise corporate networks to steal corporate secrets. Such secrets can be used for gaining competitive advantage or causing economic damage to the company. Such attacks have received little press attention, due to the reluctance of victimized companies to disclose them publicly.

Denial-of-service extortion. A particularly disruptive form of attack involves sending a large number of packets to one or more target machines to overwhelm their ability to communicate on the network, a technique known as packet flooding. With a small-scale botnet of a mere 1,000 machines, attackers could hobble the Web site of a typical medium-sized organization with a packet flood. For some organizations, such as Internet merchants, such a packet flood could have catastrophic economic consequences. If customers cannot reach the site, some companies are, in

effect, closed for business. Cyber criminals have capitalized on this possibility through the use of extortion, demanding payment to prevent a flood or offering to provide "security" or "protection" services for a price and threatening dire consequences if payment is withheld. Attackers can usually sustain a flood for three days to a week; any longer and most ISPs can differentiate the flood traffic so that they can start shunning it. This type of extortion scheme proliferated in the mid-2000s, first targeting online gambling and commercial pornographic Web sites, then small- to mid-sized mainstream e-commerce sites and financial services institutions. A variety of companies have paid under such threats; others have refused and may have suffered the consequences.

Large-scale Attacks

In the early 2000s, the dominant threats on the Internet shifted from hobbyists and isolated criminals to organized crime groups. The major threat vector could evolve again, possibly toward larger scale attacks waged by nation-states or nonstate actors seeking to cause widespread damage. The Estonian cyber attacks in spring 2007 could be a harbinger of future attacks: for several weeks, waves of packet floods targeted major financial institutions and government agencies in Estonia, which is heavily reliant on its Internet infrastructure.[4] Filters and shunning technology blocked the attack only after extended downtime for some sites. The attack may have been retaliation for a decision by Estonian officials to move statues commemorating the Soviet victory over the Nazis in Estonia during World War II. It is unclear whether the attack was directed by individuals inside or outside the Russian government; Russian officials have denied government involvement. Either way, this directed attack represents the first explicit large- scale computer attack for political rather than economic purposes.

Four types of cyber attack could damage a large target population: denial-of-service packet floods, exploitation of infrastructure components, damage of client systems with widespread botnets, and mass credit card fraud with identity theft. Each uses existing capabilities and technological means already accessible to sufficiently funded and motivated nations and nonstate actors. For each type of attack vector, smaller scale historical examples from the past decade suggest the shape of larger scale attacks in the future.

Denial-of-service packet floods. Web servers of various companies and organizations are frequently subject to packet-flooding attacks by disgruntled customers, political opponents, or others. These are small-scale attacks by the definition offered in this chapter, but attackers might also target systems that have a larger impact, as this section explains.

Today, most flood traffic originates with a botnet of perhaps 100,000 to millions of machines, a size already commonly used by organized crime. The largest botnet publicly documented to date used 1.5 million machines. Dutch law enforcement authorities documented such a botnet in October 2005. A handful of other multimillion-system botnets controlled by organized crime groups have also been identified.

Although most of today's botnets are used by criminals in small-scale attacks to make money, some of the larger botnets have been used in large-scale flooding attacks against specific Web sites or the Internet's infrastructure. A large-scale attack might involve flooding

critical Web sites and related systems of a given organization, business sector, or country. In 2000, for example, many major e-commerce sites in the United States, including stock-trading firms Ameritrade and E*Trade, were attacked with a packet flood. The 2007 attacks against Estonia were a similar large-scale operation, launched from several cooperating botnets. In a similar fashion, several major e-commerce retailers might be flooded in an effort to strangle a whole country's financial transactions.

Alternatively, to broaden the damage, attackers might seek to flood the Internet infrastructure of a target country or even the Internet as a whole. While the Internet was devised to have redundancy and the ability to bypass interruptions of traffic flow, an attack against certain portions of the Internet infrastructure could restrict traffic flow and impact millions of users.

Of particular concern from an infrastructure perspective are the domain name system (DNS) and backbone routers. DNS is a network service that converts a system's name (such as <www.ndu.edu>) into a numeric IP address. When a user types a domain name into a browser, software on the browsing machine queries a nearby DNS server, which in turn queries other servers in the DNS hierarchy to retrieve the domain-name-to-IP-address mapping, a so-called address record. The destination IP address for the given Web site is placed in every packet of data for that site, so that the network can carry the packets to the right location. Thus, DNS represents a critical component of the Internet, mapping human-entered names into network-understandable and -routable IP addresses.

At the top of the DNS hierarchy are 13 root servers that are distributed around the world; they provide information about various lower level DNS servers.[5] When lower level servers are booted up, they may contact the root DNS servers to retrieve and cache information about other components of the DNS hierarchy so that they can start responding to user queries. If one root name server is unavailable, software automatically adjusts, moving to the next root name server to retrieve the information requested.

However, if all 13 root name servers could not be accessed because of a packet flood of bogus DNS requests, the Internet itself, for most users, would decay as more and more IP addresses could not be fetched. The decay would be gradual, because lower level DNS servers temporarily hold on to records in their local cache for a period that typically varies between a few seconds and a few days, depending on how frequently the owner of the record plans on updating DNS entries. If the root DNS servers were all annihilated, more and more systems on the Internet would become unavailable to users as records expired over time, except those users who had memorized or stored name-to-address mappings for sites that they wanted to access.

Flood attacks against the 13 root DNS servers were attempted in 2002 and again in 2007.[6] In 2002, nine of the 13 servers were taken offline in a massive flood. While the four remaining root DNS servers were able to handle the load, the attack did cause a great deal of concern about the robustness of the DNS infrastructure. To help alleviate this concern, the operators of most of the root name servers deployed a technology called *anycast*, which allows multiple distributed machines to act together as one server. Therefore, while there are still 13 named root DNS "servers," many of them are really collections of dozens of machines deliberately distributed across different continents. As a result of anycast deployment, the next attempted DNS root server flood in February 2007 was far less successful: only two of the root name servers were significantly affected, and the vast majority of Internet users did not even

notice the attack.

Besides floods against DNS servers, backbone routers operated by ISPs represent another critical infrastructure component. These routers are the central points in the network through which traffic flows. Routers are essentially specialized computers that move traffic from subnetwork to subnetwork between different physical interfaces of the router itself. These backbone routers are operated by large ISPs. Many major countries rely on several ISPs, although some lack such diversity. In some countries, a single legacy telecommunications provider offers Internet access; elsewhere, corporate mergers may be decreasing the number of unique ISPs. Typically, all traffic is directed through 10 to 100 backbone routers (or perhaps as many as several hundred) that constitute the main infrastructure of an ISP. A determined attacker with a large botnet—perhaps hundreds of thousands or millions of machines—could target the infrastructure routers of a single ISP, or the ISPs of a whole country, to try to overwhelm them with bogus traffic. With all of the routers choking on bogus packets, users and servers within that country would have difficulty accessing systems within the country and could be completely blocked from accessing systems outside of the country.

Of the three packet flood targets for large-scale attacks described above, the most likely to succumb to such a flood are the e-commerce and e-government Web servers, followed by DNS servers, followed by ISP backbone routers, because of the redundancy introduced by DNS with anycast technology and the redundancy of the ISP architectures of most countries. While major e-commerce Web site operators often have 10 or more redundant Web sites, these systems are not as robust as the DNS or ISP infrastructure.

Among the different kinds of packet floods, the most frequent today are synchronize (SYN) floods, hypertext transfer protocol (HTTP) floods, and DNS amplification attacks.

The first widespread SYN floods started in the United States in 1996 and have since become a daily occurrence on the Internet around the world. SYN floods involve undermining the session initiation technique used by the TCP. Most Internet-based services rely on TCP as a transport protocol, including the HTTP for Web surfing, file transfer protocol (FTP) for file transfer, and secure shell for remote system access and administration. One of the crucial properties of TCP involves the sequencing and reliable transport of packets. TCP is designed to ensure that packets arrive and that they arrive in order.

To achieve these properties, TCP depends on sequence and acknowledgment numbers that it embeds in the headers of packets. All legitimate TCP connections begin with the TCP three-way handshake, a simple interaction designed to exchange sequence numbers for that connection. To start a connection, the initiating machine (such as a Web browsing machine) generates a TCP SYN packet that includes an initial sequence number for packets going from the initiator to the receiver of that connection (say, a Web server). If the receiving machine is configured to accept the connection request, it responds with the second portion of the three-way handshake, a SYN–ACK (acknowledgment) packet, indicating that it acknowledges the sequence number it received and that it will synchronize around a new initial sequence number for all response packets. To complete the three-way handshake, the initiator then sends an ACK packet, acknowledging the sequence number the recipient wants to use. Thus, the two systems have exchanged sequence numbers, so that all packets that follow on the connection will increment the appropriate sequence number for each byte of payload data transmitted.

A SYN flood exploits this three-way handshake by stalling it two-thirds of the way through the connection initiation. The attacker sends a SYN packet, to which the recipient

responds with a SYN–ACK packet. The attacker never sends the ACK to complete the three-way handshake, leaving a dangling, half-open connection on the destination machine. If this incomplete exchange is repeated thousands or millions of times per second, the target machine can become unable to respond to other requests.

In recent years, most ISPs have deployed traffic sensors to detect the dis- tinctive traffic pattern associated with SYN floods.[7] If it is detected, they start shunning such traffic automatically, lowering or even eliminating the damage from a SYN flood. Although such technologies are increasingly widely deployed, attackers still attempt SYN floods and are sometimes successful.

Because of ISP success in preventing SYN floods, some attackers are evolving new kinds, including HTTP floods. An HTTP flood looks like legitimate traffic but involves a massive number of legitimate-looking requests. Automated tools designed to detect flood patterns have more difficulty in differentiating these attacks from normal traffic and thwarting them.

ISPs do have some mechanisms for dealing with HTTP floods. Analysts at the ISP can study the bogus HTTP request traffic coming from bots and try to characterize patterns in that traffic, such as a repeated request for a given Web page with certain parameters. Attackers seek to make their traffic difficult to identify by removing patterns that might allow an ISP to differentiate the flood traffic from the legitimate traffic. More sophisticated attackers are better at disguising their traffic.

Another flood type is a DNS amplification attack. The attacker sends small query packets to hundreds of thousands of third-party DNS servers; each query causes each server to send a larger response packet, resulting in an amplification of the traffic load. To direct this traffic load at a victim machine, the attacker sends each query from a spoofed source address, as if it came from the targeted machine. Responses addressed to the victim overwhelm the victim's network connection. The servers are not themselves the targets, but rather are used as amplifiers to inundate another victim machine with a flood.

Since 2005, DNS amplification has generated flood rates in excess of 20 gigabits per second, equivalent to the bandwidth of some backbone routers and very large e-commerce facilities. An attacker using this technique could interrupt the service of even large e-commerce players unless their ISPs can devise signatures to characterize and thwart the spurious DNS responses.

Flood attacks of these types would likely have significant impact over short periods, affecting millions of users for 12 to 72 hours. ISPs would then most likely be able to characterize that specific flood's traffic and devise methods for filtering it. However, a more persistent and technically adept attacker might plan methods for altering the characteristics of the flood as it occurs, perhaps starting with SYN floods, then simple HTTP floods, followed by even more complex traffic forms. Keeping up with such an adversary could prove difficult, and an attack might be sustained over several weeks or more. Moreover, while the ability of U.S. ISPs to devise a signature and coordinate its deployment of filtering is quite good, coordination with overseas organizations could be difficult.

Exploiting infrastructure components. Another avenue for large-scale attack involves exploiting vulnerabilities in infrastructure systems, such as backbone routers or DNS servers. The software at the heart of major infrastructure devices may have bugs or flaws; most are

mere annoyances, but attackers might deliberately trigger some flaws to harm a system. Software programs to trigger such vulnerabilities are known as *exploits*.

Some software vulnerabilities could allow an attacker to cause a target machine to crash, resulting in a denial-of-service condition, or could compromise a system, with the attacker taking over administrative control of the machine and bypassing normal security measures. An attacker could simply shut the system down or use it to conduct even more insidious attacks. Having taken control of critical infrastructure components such as routers or DNS servers, attackers could redirect traffic to other destinations anywhere on the Internet so that, for example, all traffic destined for a given country's banks would be directed to a different country. System compromise might let attackers capture traffic going across the Internet, such as sensitive financial transactions, management data associated with the infrastructure itself, or business communications, which could be recorded for later analysis and use.

Once or twice a year for the past decade, independent researchers have discovered vulnerabilities in parts of the Internet infrastructure that could be exploited.[8]

Two examples illustrate the issues underlying these types of vulnerabilities and the large-scale attacks that could have resulted from them. In early 2004, researcher Paul Watson identified a TCP reset technique that could prevent routers from exchanging routing updates with each other.[9] This approach could disable routing updates on the Internet, which would have caused the network itself to degrade over several hours, ultimately resulting in a loss of connectivity for chosen segments of the Internet, such as the entire United States, or perhaps all systems associated with U.S.-to-Europe connectivity. Before this vulnerability was exploited, however, Watson publicized its existence, and large ISPs around the world and government, commercial, and military enterprises deployed a patch to mitigate the vulnerability.

Similarly, in July 2005, researcher Michael Lynn discovered a way of exploiting routers manufactured by Cisco Systems, Inc., that could have been used to crash or take over the routers and reroute traffic.[10] Lynn announced his approach to Cisco and made a presentation on it at a hacker conference; the associated vulnerabilities were then patched. This type of vulnerability could go beyond denial of service to rapid takeover and crashing of large numbers of key routers. The fix for this type of flaw would be difficult to distribute and deploy if the Internet itself were down.

Popular DNS server implementations have also had significant vulnerabilities over the past decade.[11] To launch a large-scale attack involving the exploitation of critical infrastructure systems, attackers would have to find vulnerabilities before vendors or well-intentioned security researchers do. In the past two decades, most of such flaws publicly disclosed have been discovered and publicized by independent hobbyists, commercial researchers, and the vendors themselves. Would-be attackers do not need to do any tremendously difficult analysis to find these flaws; their discovery involves looking for a series of known kinds of flaws in commercial products. Product evaluation methodologies, applied in a comprehensive fashion, can discover a significant number of these flaws. Because ISPs rely on much of the same routing software that smaller institutions do, and because other Internet infrastructure components run on software that is free or available inexpensively, a well-stocked research lab for finding these kinds of flaws can be created for between $3,000 and $20,000, a relatively

small investment to discover high-impact security flaws.

The amount of time between discovery of a security vulnerability and the public release of exploit code by malicious attackers that could take over a target machine is shrinking, from six to 12 months in 2000 to a few days or less in 2007. Both legitimate researchers and attackers have developed automated methods of finding security-related flaws and creating exploitation code. Exploitation has begun to appear "in the wild" before vendors or legitimate security researchers discover flaws; only when systems start succumbing to the exploit is the vulnerability detected. Such attacks using previously undisclosed vulnerabilities are occurring regularly against desktop and laptop computers today. In the future, they could be used to target routers, domain name servers, and other critical infrastructure components.

Other kinds of infrastructures are increasingly being managed and controlled using the TCP/IP protocol suite and other Internet technology, including commercial-off-the-shelf switches, routers, and operating systems. Historically, supervisory control and data acquisition (SCADA) systems, which are used to manage complex installations such as nuclear power plants and electric grids, were based on proprietary communications protocol and isolated networks. This arrangement made it difficult for would-be attackers to find vulnerabilities in the technology, which was relatively difficult to acquire. Now, however, such systems—including aviation systems of commercial aircraft, military equipment, nuclear power plant monitoring, and other technologies—are increasingly using standardized TCP/IP components.

The move toward using Internet technology to interface with and manage other types of systems is based on simple economics as the technology has grown cheaper, smaller, and lighter. Internet technology is especially attractive where minimizing weight and size are important, such as in aircraft avionics and control. Rather than design and deployment of a massive and expensive network of custom equipment for managing an electrical plant, use of off-the- shelf Internet technologies can significantly lower costs.

With TCP/IP spoken by most computing systems, from mainframe computers and desktop machines to cell phones, use of commonly available hardware and software makes systems more flexible, with the ability to interface with numerous inexpensive devices. However, use of standardized protocols and common implementations by sensitive infrastructures introduces significant risk. There are tradeoffs between cost and security. Most secure would be air- gapped or isolated networks that use entirely different protocols from others; a sensitive network, such as the aviation controls of an aircraft, the SCADA systems controlling components of a power grid, or military command and control systems, could use custom network technology (not Internet protocol version 4 [IPv4]) on a network that is completely separate from the Internet. On the other end of the spectrum is integration using a common network protocol such as IPv4 on networks that are interconnected. Some protection might be offered by a firewall that polices the traffic, filtering out unneeded services and potential attacks. Even with such filters, at this end of the spectrum, there is some connectivity.

Other possible topologies strike different balances between economic benefit and lower security risk. For example, a single physical network with two or more different network protocols could achieve, if not complete isolation, at least some separation of traffic. But such solutions still involve security risk. To see why, consider one physical network with systems that

use two different network protocols; call them Protocols X and Y. In this hypothetical multiprotocol network, most rank-and-file users might rely on Protocol X, whereas some special and important equipment uses Protocol Y. Only the special equipment has endpoint software to speak Protocol Y. The routers that comprise the network may, in the simplest but least secure solution, understand both Protocols X and Y so they can route packets for both protocols across the network. However, instead of making the routers aware of both Protocol X and Y, another option for implementing a multiprotocol network is to use a process whereby the entire network, including all endpoints and routers, speaks Protocol X, but only certain specialized endpoints (not routers) on the network have software that embeds Protocol Y inside of Protocol X packets. Such a "tunneling" solution is used for implementing mixed networks of IPv4, the current dominant Internet protocol, and IPv6, its successor, whose deployment is under way.

Multiprotocol networks, whether implemented with routers that speak multiple protocols or through tunneling, do yield some isolation. Attackers who had software that only spoke Protocol X could not directly attack Protocol Y systems; however, they might create, buy, or find software that can speak Protocol Y. To lower this concern, all Protocol Y traffic could be encrypted as it traverses the network, so the attacker would also have to break the encryption keys or find a flaw in the implementation of Protocol Y.

Even with encryption of Protocol Y traffic, another security problem frequently encountered in multiprotocol networks arises with the use of network gateways. With shared physical infrastructure, some users almost always will want to move data associated with Protocol X applications to Protocol Y applications. To speed up the process of such conversion, these users may introduce gateways that speak Protocol X on one side and Protocol Y on the other so they can shuttle data between the two. Even if encryption is used for all Protocol Y traffic, the gateway may act as an endpoint for the encryption.[12] A gateway may be introduced by users who are unaware of how it undermines the security of the Protocol Y machines, and such gateways could allow an attacker with only Protocol X software to exploit the machines.

Multiprotocol networks are at risk of denial-of-service flooding attacks. Even an attacker with only Protocol X software could launch a flood attack against the routers, consuming them with so many bogus Protocol X packets that they drop Protocol Y altogether, leaving the special equipment that uses only Protocol Y unreachable. Some networks configure routers to favor Protocol Y over Protocol X, but a deluge of Protocol X packets would overwhelm such a prioritization scheme.

If there is a flaw in the Protocol X routing software in the routers (such routing flaws are fairly common), an attacker could send exploit packets to compromise the routers via the Protocol X flaw, gaining control of the router, which handles both Protocol X and Protocol Y. The attacker could then intercept Protocol Y traffic, and possibly decode it, gaining the capability of using the Protocol Y software to compromise the special equipment (in a sense, exploiting a router to create its own gateway between Protocol X and Y to exploit the specialized equipment). Multiprotocol networks offer better security than having all network components speak the same protocol, but they are not as secure as air-gapped networks in protecting against the compromise.

Real examples of such mixed networking include the Internet's most common protocol, IPv4 (usually referred to as simply IP), and Novell's old IPX protocol. Although the acronyms IP

and IPX sound similar, the two protocols are very different. IP is an open protocol with numerous vendor implementations for use on the Internet, while IPX is a proprietary protocol used by Novell for some of its older enterprise network products. In the 1990s IP and IPX were often mixed on corporate networks, with IP used for Internet access of Web sites and email, and IPX used for Novell file and print sharing. Protocols X and Y could represent a myriad of other protocols as well.

Damaging client systems with widespread botnets. Another form of large-scale attack directly targets massive numbers of end-user systems such as desktop and laptop computers, cell phones, personal digital assistants, and other computing devices. Attackers could affect millions of users with bot software installed on end-user systems to cause harm to the systems themselves, such as deleting critical files or damaging the video card, hard drive controller, processor, or other hardware. An attack might corrupt files, such as financial information associated with purchases or tax preparation software. Each of these attacks could be done on a small scale, but accomplished across a large botnet, such malicious activity could have an impact on large numbers of users. Although criminal enterprises have the capacity to engage in this type of attack, such disruptions have not occurred, because of the more lucrative use of botnets for small-scale criminal activities discussed earlier.

To construct a large-scale botnet, attackers rely on various methods for compromising new hosts, such as the use of worms, self-replicating code that spreads across a network infecting vulnerable machines. With a worm, one infected machine scans for and finds other vulnerable systems. The worm then spreads to those new victims, copying its code; victim machines then find other targets, and worms may spread exponentially. Since at least 2004, worms have been used to carry and install bot software, allowing the attacker to control the newly infected machines.

Some bots are distributed as executable email attachments, duping users into installing the bot by claiming that an important attachment needs urgent attention. Despite publicity warning against reading email attachments from unknown senders, a significant number of users succumb to this form of subterfuge. Attackers also use spear-phishing attacks for bot distribution. Bots are also bundled with apparently benign or useful applications, such as system add-ons or games.

A frequently used bot distribution mechanism involves exploiting a client- side program, such as a Web browser, a document-viewing application such as a word processor or slide viewer, a video-viewing application, or music-playing software. Every month, numerous vulnerabilities are discovered in these types of applications. If a user views content posted by an attacker on a Web site, the content itself (which could be a document, audio file, or other file format) could trigger the vulnerability in the client program, making it install the attacker's bot software. Some refer to this technique as a drive-by download of malicious code known as *malware*.

As cell phones, music-playing devices, and personal digital assistants become more powerful, attackers are compounding the problem by devising worms and bots to attack them, too. A botnet made up of millions of Internet-capable cell phones could cause significant damage.

Mass credit card fraud to disable accounts. Large-scale credit card fraud and

identity theft could be another attack vector via cyberspace. Today, most of the fraud committed by theft of credit card account numbers falls into the range of small-scale attacks. Attackers might grab a million credit cards from a merchant that suffers a breach, but automatic fraud detection tools operated by the credit card companies and issuing banks detect the fraud and react rapidly. Software running on the credit card companies' computers rapidly determines that the given merchant has had a breach, given the uptick in fraudulent activity tied to cards used at that merchant recently. These cards are then disabled, along with other cards used at that merchant, foiling the attacker's chance to use the other stolen account numbers. Quite often, the credit card companies detect a breach based on the fraudulent use of cards before the merchant that suffered the breach realizes it. Thus, despite the large number of compromised accounts, the actual amount of fraud committed with stolen credit cards has been kept between one and four percent of all credit card transactions.

However, the antifraud systems could also be used to instigate a large-scale attack. When a breach is detected and card accounts are disabled, consumers may have to telephone the issuing bank to either reactivate an account or to request issuance of a new card. An attacker who wanted to cause economic damage could purposely generate bogus transactions for tens of millions of credit cards, triggering mass account shutoffs. In the 1990s, generating that number of credit card transactions was difficult if not impossible, but today, a million-system botnet could initiate transactions through thousands of e-commerce sites. Consumers would not be able to use their cards unless credit card and bank personnel temporarily suspended automated antifraud account shutdown functions. Such an attack could have a noticeable impact on the economy.

Defensive Technologies and Associated Public Policy Issues

This section surveys common and powerful cybersecurity defensive technologies, explaining the concepts underlying each and some policy options and questions they pose. The discussion examines network- and host-based defenses as well as defensive concepts that apply to both.

Public policy decisionmakers can influence network-based and host-based defenses in different ways. Network-based defenses differ in that they tend to be more scalable and applicable within enterprises and government agencies, and possibly even nationwide, through the deployment of systems at network interconnection points to protect large numbers of computers on the given network. For example, an ISP might deploy defensive technologies that can benefit all of its customers through a relatively small number of critical network junctions; perhaps 10 to 100 systems can coordinate in the defense of all hosts on the ISP's network. Network-based defenses might be required of network operators, including ISPs and large enterprises, or vendors of network equipment and software might be required to offer certain security capabilities with their products.

Host-based defenses, by contrast, involve installing software on a system- by-system basis; they can protect large numbers of machines, but with a more expansive, invasive, and usually more expensive technological deployment. For widespread host-based defenses, software would have to be deployed on perhaps millions of machines or more, including consumer, commercial, and government systems. Such defenses can be incorporated into the operating

system itself or as standard features of other widely used packages, such as productivity suites or antivirus tools.

Requirements for host-based defenses could be applied either to end-users or to vendors (of operating systems, browsers, databases, office suites, and the like). Given most users' relative lack of rigor and technical sophistication in configuring complex security software, requirements placed on software vendors are likely to have greater impact in improving host-based security.

Network-based Defenses

Operators of large-scale networks, including commercial ISPs, major enterprises, government agencies, and the military, must weigh the impact of a variety of competing goals, including performance, manageability, and scalability, against the impact of security measures. A security technology that combs all network traffic looking for signs of attack with a high degree of accuracy but that slows the network to a crawl would be unacceptable. Also unacceptable would be security technologies that make the network so complex that it cannot be managed effectively or that impair its ability to grow enough to support its entire user base. For this reason, the vendors offering security technologies and the network operators using them must carefully vet their security tools before deploying them. While each of the technologies covered in this section has been applied successfully in large-scale networks, not every defensive technology is suitable for every network provider.

Firewalls. Firewalls filter network traffic, allowing certain types into a network while blocking others, based on the firewall's configuration. Most firewalls filter based on the type of network services; they may, for example, allow Web traffic while blocking network management traffic arriving from the Internet. More advanced firewalls may allow specific source or destination addresses to be blocked. The most sophisticated firewalls provide content inspection, analyzing the data inside packets to determine whether it contains application data, key words, attack patterns, or specific phrases that should be blocked. Network firewalls are often deployed at the interconnection points between two networks, such as the border between an enterprise network and the Internet. Such firewalls are usually configured to allow inbound access to specific Internet-accessible Web servers, mail servers, and other related systems. In most organizations, outbound filtering is far more open; many organizations choose to allow all traffic out of their networks. Because exfiltration of sensitive information from corporate and government networks represents a risk, some organizations also filter out- bound access.

Some countries, notably China and Saudi Arabia, firewall all of their outbound traffic, using numerous firewall machines operating in parallel to suppress access to Web sites and other Internet activities associated with unwanted political ideas or religious expression. However, such firewall deployments are not perfect. Political and religious dissidents have devised methods to fool them, often using tunneling technologies to carry controversial traffic that might otherwise be blocked inside of innocuous-looking packets, sometimes applying encryption to minimize the chance of inspection by authorities.

Even with the possibility of small-scale evasion of firewalls, these country-level firewalls provide a capability for near-complete disconnection from the Internet for the

countries that operate them. The network architecture of these countries is built around a large number of firewall chokepoints through which traffic must pass for analysis. If, due to a geopolitical crisis, these countries wanted to shut off both inbound and outbound Internet access, they could leverage their firewall infrastructure to block access very quickly, likely within a few minutes. In late 2007, Burma severed its Internet connectivity during political unrest, severely limiting the flow of information into, and perhaps more importantly, out of the country.[13] Countries without such firewalls and where international connectivity has blossomed with large numbers of ISPs and foreign interconnections would have a harder time doing such thorough blocking so quickly. It is unlikely, for example, that the United States would ever move to firewall all Internet connectivity or even contemplate the full-scale breaking of international connectivity, due both to the severe economic implications and the widespread connectivity offered by its multiplicity of ISPs.

However, given that some countries have the capability of rapidly implementing firewall-based isolation, the United States may want to consider methods for gaining access to the Internet infrastructure of a country that has employed firewall filtering on a countrywide level. Such methods could include satellite links, connections via international companies operating inside the country, or covert agents operating inside the firewalled country itself. Even if such connections cannot politically or diplomatically be implemented unless a definitive crisis offers an immediate justification, the United States may want to draw up plans for the rapid establishment of such connectivity for operations in various countries, should such access ever be needed.

Network-based intrusion detection systems. Network-based intrusion detection systems (NIDS) monitor Internet traffic looking for attacks. When an attack is discovered, the NIDS tool alerts network management personnel, operating like a network burglar alarm. Many commercial and government enterprises deploy NIDS monitoring sensors at their Internet gateways, just inside their firewalls, to determine if an attacker has penetrated their "front door." Some organizations deploy NIDS sensors throughout their internal networks, monitoring for attacks throughout.

Most of today's NIDS technology focuses on signature-based detection. For each known attack, the NIDS vendor creates a specific definition of telltale signs in packets that would indicate such an attack is under way. For example, a given software flaw may lead to a vulnerability in a router that attackers can exploit to take over the router. A NIDS vendor may write a signature describing the pattern of packets that have specific settings that indicate an exploit is attempting to trigger the vulnerability. Such signatures are published regularly, with many thousands available on a free and commercial basis.

Some NIDS technology also uses behavior-based protection, based on identifying deviations from "normal" usage of protocols in a given network. A NIDS sensor may detect an attack due to unusual protocol behavior, such as repeated SYN, SYN–ACK patterns, without completion of the TCP three-way handshake seen during SYN flood attacks described earlier. Another form of behavior-based NIDS tool looks at connection flow information, analyzing the source and destination points of connections going across a network, and the services associated with each flow, to determine whether it matches the normal expected patterns for the given network.

Attackers have an interest in devising methods for evading detection of both signature-

based and behavior-based NIDS tools. Exploits may split up and encode data so that it still functions against the target but without displaying signatures to NIDS tools. Attackers may also seek to make their attack mimic legitimate traffic in order to evade behavior-based defenses. Such stealth capabilities are increasingly available in commercial or free open-source hacking tools.

Attackers work to evade detection, but most attacks, especially very large-scale ones such as denial-of-service floods against infrastructure targets, are detected based on the immediate impact of the attack itself. Most major commercial enterprises and government agencies have some form of NIDS capability. Some, but not all, ISPs likewise have deployed NIDS tools on their networks. However, coordination and analysis across these NIDS do not exist in many industries. A determined, countrywide attack might not be recognized until it has already caused significant damage. Some industries have formed information sharing and analysis centers (ISACs), cooperative groups of information security professionals, to share information about attack activity. The financial services ISAC was one of the first, and they have been established for information technology, energy, state government, and other sectors. ISACs provide a window into activity associated with only one industry and, as might be expected, companies may hesitate to share information that could damage their reputation or competitive advantage. For these reasons, ISACs do not provide a comprehensive detection capability for widespread attacks against the United States.

To make them more useful against countrywide attacks, NIDS tools could be deployed on ISP networks at points of interconnection with other countries. The U.S. Government or military could apply such tools to monitor all traffic coming into or going out of the country, looking for coordinated attacks. ISPs may be reluctant to have such monitoring devices on their network, and privacy advocates might be concerned about intrusive monitoring. However, a program could be devised that minimizes impact by looking only at packet header or traffic connection flow information. If applied to the major fiber optic connections into and out of the country, monitoring could cover most but not all Internet traffic. A small segment of traffic that would be more difficult to monitor is satellite communications carrying IP traffic into the United States, due to their geographically widespread distribution, the lack of publicly available information about all satellite connection points in the United States, and the ephemeral nature of such connections. However, even a monitoring capability focused only on fiber-based transmissions could provide significant warning of widespread attacks.

Due to the large amount of traffic flowing in and out of the country through ISPs, a monitoring solution of this type probably would focus only on traffic flows and not individual packets and their contents. Such a detection capability at the network borders would be analogous to the U.S. Coast Guard's monitoring of the Nation's ports or the U.S. military's early warning missile launch detection systems, applied to the cyberspace borders of the United States.

Network-based intrusion prevention systems. While firewalls focus on allowing or denying particular services and NIDS tools detect attack activity, network-based intrusion prevention system (NIPS) tools combine these two concepts. NIPS devices monitor network traffic looking for attacks. When packets associated with an attack are detected, the NIPS may drop those packets or reset the connection, stopping the attack from functioning and thereby protecting the end system. Due to the risk of blocking traffic if NIPS misidentifies legitimate traffic as an attack (known as a false-positive condition), some NIPS tools are tuned

so that attacks commonly associated with false positives generate an alert rather than a blocking action. False positives could cause significant problems in an enterprise environment, breaking important applications if their traffic accidentally matched the attack patterns the NIPS is configured to detect. A NIPS tool configured merely to alert for some types of attacks takes on the same behavior as a NIDS tool. Like their NIDS cousins, NIPS products detect attacks using a signature-based approach, behavior-based identification, or a mixture of both techniques.

Some NIPS tools operate inline: traffic flows pass through the NIPS for inspection and possible blocking. Other NIPS tools sit beside the network and sample its traffic to detect attacks. Inline NIPS tools can provide a comprehensive view of attack activity in the stream because they inspect each and every packet. Inline NIPS tools can also effectively block unwanted traffic simply by dropping a packet and not allowing it through. NIPS tools that sit beside the traffic flow, on the other hand, may miss some dangerous packets in a fast stream. Furthermore, if an attack is detected, the NIPS tool that samples the traffic may not be able to reset the connection quickly enough, and an attacker might cause damage before it can be blocked. Thus, from a purely defensive measure, inline NIPS tools offer some advantages, but from a network performance and operations perspective, inline NIPS tools could become a performance bottleneck, slowing network traffic to inspect it. Because inline NIPS tools must both inspect and forward packets, the tools could become overwhelmed by a high traffic load; this could give attackers a way to launch a denial-of-service attack against the network by clogging up the inline NIPS tools with bogus packets. NIPS that sit beside a network sampling its traffic typically do not suffer from this performance bottleneck.

In variations of NIPS technology, some ISPs and large enterprises have distributed sensors throughout their networks to detect unusual traffic flows that might be attacks, especially denial-of-service floods. These tools may have the ability to throttle such traffic, holding back the onslaught of packets to a level that a target machine might be capable of handling, a technique called *traffic shaping*. Such sensors—the same technology as the packet-flood shunning concepts described earlier—are becoming more able to recognize attacks in an automated fashion and shun the traffic associated with the flood.

Either NIDS or NIPS tools could be used for a nationwide cyber monitoring system at the U.S. "borders." NIDS provides detection capabilities but cannot block or throttle an attack. Functionality for blocking the attack would have to come from other systems, perhaps the machines under siege. If NIPS tools were the early warning system, the United States could use the system to shun the attack, but this capability comes at the price of potentially lower performance and the risk of false positives.

Network encryption. The original Internet protocol specification included neither network-level encryption to prevent eavesdropping nor authentication mechanisms to identify machines or users. Security functionality was left to end- user computers and applications developers; the network protocols were geared more toward moving packets end-to-end than providing security. In the absence of network-level encryption, three end-to-end encryption technologies became popular in the 1990s. While any of the three could be used to encrypt any kind of data moving across the Internet, each found favor with a different segment of Internet applications and technology. The pretty good privacy (PGP) program created by cryptographic hobbyist Phil Zimmerman was commonly applied to encrypting email and files. The Secure Shell (SSH) suite was applied primarily to protect remote login capabilities in which an

administrator or user gained remote access to a system across the network. The most widely deployed encryption tool, the Secure Sockets Layer (SSL), was usually applied to securing communication between Web browsers and Web sites. These three technologies are still widely used to encrypt data flowing across the Internet, but each requires that both ends associated with the communication, the sender and receiver, have special software and encryption keys in order to use the application-layer encryption.

In the mid-1990s, the Internet Engineering Task Force[14] recognized the need for having network equipment, rather than end systems, encrypt packets and authenticate their origin. Thus, they defined Internet Protocol Security (IPsec), a protocol that was "retrofitted" into IPv4 and incorporated in IPv6 to provide network-level encryption. Any application on the end systems could thus take advantage of cryptographic protections from the network itself, without any changes to the application (unlike PGP, SSH, and SSL). In recent years, a variety of vendors have released IPsec-compatible software in operating systems, routers, firewall equipment, and a variety of other devices. IPsec is a very complex protocol with numerous options; it required rigorous compatibility testing and implementation fixes. Today, the most popular operating systems, including Windows, Linux, Mac OS X, and others, support IPsec, as does major network equipment.

IPsec offers various kinds of security capabilities via cryptography, including confidentiality, so no one can read the contents of packets without the decryption key; authentication, identifying which user or machine sent each packet; and integrity, to verify that a packet's contents were not altered. IPsec was designed to operate with traditional IP on the network so that secure and unsecured communications could coexist.

IPsec communication can be deployed in a variety of ways. The simplest example is a point-to-point connection between two end systems, such as two workstations or a workstation and a server. A system administrator can configure the two systems with a preshared encryption key, which in turn is used to exchange other keys that encrypt the data going from system to system. In network-to-network encryption, a network administrator can configure two routers or firewalls on the Internet so that all traffic going between the two systems is encrypted. Thus, two enterprises engaged in a joint venture can encrypt all traffic going between their two networks across the Internet by applying IPsec at their Internet gateways and configuring them with encryption keys. Or, in another example, encrypting data between many end-user computers and a network gateway, such as a router or firewall, allows a large number of users to access a corporate network, for example, in a secure fashion.

In each of these three deployment scenarios—system-to-system, network-to-network, and system-to-network—IPsec is used to create a virtual private network (VPN) to carry sensitive data securely across the public Internet. Such VPN capabilities are significantly less costly than dedicated dial-up or point-to- point facilities.

While IPsec, PGP, SSH, and SSL encryption can provide a great deal of security, a major impediment to universal deployment is that each requires distribution of encryption keys, which are essentially series of secret strings of numbers. Ensuring that encryption keys are distributed only to legitimate users and that they are protected from attackers is thus paramount. The simplest deployments of IPsec among a very small number of users or networks can use basic preshared keys, configured by administrators, but such deployments are

unworkable beyond about a dozen intercommunicating systems because of the complexities of key distribution and secrecy in a larger organization.

To scale to larger user and network populations, various organizations have devised management systems to distribute keys in a trusted fashion. Technologies for secure key distribution rely on one or more parties, trusted by all users of the keys, to digitally sign all keys in the system. These key signing entities are referred to as certificate authorities (CAs), because they generate certificates, or data packages that include a user's or network's key, name, and other attributes. Digitally signed by the CA, a certificate acts like a digital identity card, verifying users' or systems' identities and allowing them to encrypt communications for others who trust the CA that issued each system's certificate. The complexity of determining whom to trust to sign keys has limited their widespread deployment. Isolated pockets of encrypted communication use IPsec, PGP, SSH, or, to some extent, SSL. For example, many organizations have signed and distributed their own IPsec keys for use by their employees and business partners. Many individuals have created and signed their own PGP keys for use by their business associates. System administrators create and sign SSH keys for the users of their machines. But each of these uses results in pockets of communicating systems, not an all-encompassing encryption scheme. Each system may use the same software technology and encryption algorithms, but if they do not all trust the same CA, they cannot communicate with each other securely. A variety of companies offer commercial CA services to the public and to enterprises, but there is no interoperability between the certificates of different CAs. Some government agencies have contracted with a commercial CA or have started to issue their own certificates, acting as a CA for their own interactions with the private sector.

Market forces may lead a small number of trusted CAs to offer inter- operability between their certificates; this would result in a federated certificate infrastructure that Internet users could rely on. However, each commercial CA has an interest in selling its own certificates in a market with heavy competition. Various CAs have different standards for security and for issuing certificates.[15] An all-encompassing federated certificate infrastructure established by market forces might cater to the lowest common denominator of CAs, resulting in less security than some companies and government agencies require. For these reasons, government regulation or standards for CA practices might be necessary to establish a reasonable level of security. Multiple layers of security, each with its own set of security practices for CAs, might support different values and types of transactions.

While network-based encryption can provide security for information in transit, it can cause problems for the defensive technologies described earlier. When packets are encrypted, tools such as firewalls, NIDS sensors, and NIPS solutions cannot inspect the contents of the packets in order to detect attacks. Network floods and attacks that violate protocol specifications can still be detected based on traffic volumes, unusual network behavior, and signatures associated with packet header information. But most other forms of attack could be obscured by network-based encryption, especially those associated with the exploitation and compromise of a target machine or component of the network infrastructure. In order for firewalls, NIDS, and NIPS defenses to examine these packets, they would need to be configured with encryption keys for traffic flows.

Either individual cryptographic keys from each system or user or universal keys for

decrypting traffic for large groups of systems would need to be distributed to numerous network components. In most networks, however, placing keys on such network equipment results in significant privacy concerns, as well as increasing the risk that an attacker could steal the keys and completely undermine the cryptographic protection. Network-based encryption is therefore not a security panacea. Even if widespread key distribution issues could be solved with a set of trusted CAs, numerous security issues will remain. As network-based encryption becomes more common, attackers will be able to exploit systems across encrypted connections with diminishing risk of detection.

Host-based Defenses

Host-based defenses are those that protect individual systems against attack. They contrast with network-based defenses, which benefit from having a wide view of activity against a large number of machines but often cannot discern the details of action against individual hosts. This problem is compounded as attackers increasingly rely on network-based encryption technologies to obscure their attacks from network sensors. Host-based defenses, however, can see the action on a single machine and discern the details of an attack.

Host-based defenses protect both client machines and servers. Client systems include traditional workstation and laptop computers, as well as cell phones, personal digital assistants, and consumer electronics such as televisions and stereos that are controlled by built-in general-purpose computer systems. Server systems include Web, email, DNS, and file servers, and an increasing number of media servers that provide video and audio content to individuals. All of these types of hosts, both clients and servers, could utilize some form of host-based defenses, but current defensive technology has focused on workstations, laptops, and servers. As more valuable information and processing capacity propagate to other types of technologies, the types of defenses honed for traditional computer systems will likely be repurposed to these new platforms. This section analyzes some of the most common and widely used host-based defenses applicable to all of these types of machines.

Anti-malware tools. Today's malicious code takes many forms. Viruses were one of the first types of malware; these are self-replicating code snippets that infect files, copy themselves throughout a victim computer's file system, and spread wherever files are shared between machines via file servers, email, or the Web. Worms, a close cousin of viruses, also self-replicate, copying themselves across a network, usually by exploiting software vulnerabilities. Spyware, bots, and rootkits—other forms of malware described earlier—are also proliferating. Collectively, there are hundreds of thousands of specimens of malware, and attackers are continuously creating new examples.

Antivirus and antispyware tools are the most common defenses against such malware. Originally two different segments of the market, antivirus and antispyware tools have largely converged into a single product category of host- based protection suites, centered around an anti-malware scanner. Offered by a variety of commercial vendors, these tools scan a computer's hard drive and memory to detect and eradicate malware. Generally speaking, most modern anti-malware tools apply some combination of three approaches to detecting malware: signatures, heuristics, and behavior-based detection. Each approach has benefits and weaknesses.

Commercial solutions with signature-based malware detection have been available

since the 1980s. With these products, vendors regularly publish a database of malware signatures (essentially cryptographic fingerprints of viruses, worms, and other specimens), which their customers install on each system. As the anti-malware tool runs, it can detect and block the copying of malicious code onto the computer, and if malware is somehow placed on the machine, the anti-malware tool can prevent it from running. In the 1980s and 1990s, vendors told their customers to update their signatures approximately every month so that they could be protected against the latest threats the vendors identified in "the wild." As the rapid growth of the Internet in the late 1990s spurred the quick spread of new malware, monthly signature updates became inadequate, and most anti-malware vendors now publish new signatures every day or two to keep up with the rapidly evolving threat.

Attackers have a financial interest in holding on to compromised end-user machines for as long as possible and, to that end, increasingly rely on polymorphic code, that is, software that modifies itself as it runs or as it spreads to each newly infected machine. Even daily signature updates may not be enough to keep up with such attacks, but it is not realistic for most organizations and computer users to update their anti-malware signature databases more than once per day, given technical limitations on the architectures for distributing signatures and verifying their effectiveness. Thus, although signature-based solutions are helpful against the most widespread and least dynamic malware, other detection approaches are needed too.

Many modern malware attacks are thwarted using heuristic detection techniques. Whereas signature-based solutions look for an exact match of malware code against a signature, heuristic solutions look for partial matches of elements of the malware, including chunks of the malware's file, the configuration settings associated with the malware, and its file name. Heuristic defenses are "fuzzy" signatures that take advantage of the fact that attackers frequently reuse functional building blocks of code from previous malware in their new creations. Anti-malware vendors analyze malware specimens to isolate the most common element of the code, such as the instructions associated with polymorphic behavior, code used to infect sensitive portions of the computer system, or software that interacts with the network. Even out-of-date heuristic tools have a chance of detecting the newest strains of malware if the attackers reused some code, as they often do. In today's anti-malware tool suites, most of the protection is provided by heuristic capabilities.

Heuristics have their own limitations: an attacker who creates new malware without reusing any code is just as invisible to heuristics as to detection using strict signatures. An attacker may have a significant motivation for creating and using custom malware to evade signatures and heuristics in extremely targeted attacks against high-value systems.

A third common anti-malware approach is based on detecting the typical behavior of malicious code as it runs on a machine. By monitoring every running program on the protected computer, the anti-malware tool can look for aggressive behaviors such as the rapid opening, writing, and closing of thousands of files on the file system, typically associated with virus infection. For example, spyware often alters a user's browser settings to make it easier to inject ads and capture keystrokes from the victim; bots sometimes reconfigure the system in predictable ways to enhance the attacker's control of the machine. By looking for these actions as programs run, an anti-malware tool can stop the misbehaving program, uninstall it, and attempt to restore the computer's settings to their pre-malware configuration.

Behavior-based solutions afford a good deal of security, but with some fairly significant

costs. First, such solutions tend to lower performance; monitoring every running program taxes system processor and memory resources. Next, the anti-malware tool has to let the malware run at least briefly to observe its behavior before detecting it. Significant and possibly irreversible damage could occur during that time, such as the deletion or alteration of important data stored on the machine. Behavior-based solutions also have a much higher risk of false-positive detections than signature or heuristic solutions. In an enterprise environment, if the anti-malware tool identifies a legitimate program as being malicious, it could disable the program and might break a critical business application. Because impairing a corporate application could result in financial losses, some anti-malware vendors have tuned their behavior-based detection capabilities to be far less sensitive. Other vendors have avoided behavior-based solutions, focusing their energies on signature and heuristic defenses.

Anti-malware vendor solutions offer differing mixes of signature, heuristic, and behavior-based defenses that reflect that vendor's philosophy toward detection. The particular mix is typically not communicated to customers, who may assume that anti-malware protection is essentially interchangeable and that they are safe as long as some form of anti-malware tool is running. The vendors may claim that the subtle tradeoffs represented in their detection regimens are too complex for their customers to understand, which is certainly true for general consumers. Large enterprises, however, especially those associated with critical business, government, and military operations, are well able to apply technical understanding and analysis to these tradeoffs during the product selection process, and availability of such information would help to improve the match between business needs and the security tools chosen.

Host-based intrusion prevention systems. Like their network-based counterparts, host-based intrusion prevention system (IPS) tools detect various forms of attack and block them. But instead of analyzing network traffic for attacks, these tools focus on policing the running programs on each end host. The goal of network- based and host-based IPS tools is the same—blocking exploitation of target machines—but the technology and its implications are quite different. Some host-based IPS tools look for alterations in the memory of running programs that indicate that an attacker has injected exploit code into the machine. Others analyze the calls made into the underlying operating system kernel by programs as they run, checking to see if these calls are typical for the given program on that machine. Unlike most anti-malware tools that are focused on detecting malicious programs the attacker has placed on a machine, host-based IPS tools tend to look for existing legitimate programs that are misbehaving because they have come under the control of an attacker. Host-based IPS tools are an active area of research in academia and commercial security companies, given their relatively new status, the lucrative market for such solutions, and their great potential for blocking exploitation and takeover of end systems.

Host-based intrusion detection system tools that merely detect but do not block attacks have largely been subsumed into the host-based IPS market. Today's host-based IPS can be configured to detect or to block attacks.

Because host-based IPS tools by their nature analyze the activities of legitimate programs in order to look for deviations from normal program activity and enforce certain rules, they too face a risk of false-positive detections that could break important applications. In the past few years, some host-based IPS tools have gotten a reputation for overly aggressive enforcement, leading some companies to remove or disable the protection to restore an

application. Some host-based IPS tools require lengthy and complex configuration sessions by experienced administrators to "train" the tool about what is normal activity for a given application. Even after this training is completed, a new patch for the application may alter its behavior, requiring further tuning. While such tuning activities are costly, they can significantly improve the security of a system, making it immune to many of the common exploitation techniques in widespread use today.

Personal firewalls. Personal firewall software protects end-user computers and servers from network-based attacks by allowing only certain traffic into or out of the system from a specified list of programs configured by an administrator or user. For example, the personal firewall may allow the browser to make Web connections outbound to Web servers on the Internet, but it may block inbound connections seeking access to files on the protected machine unless file sharing is enabled. Personal firewalls block access to the network by malware installed by the attacker or injected into programs that would not normally require network access.

While personal firewalls do improve the security of a system, attackers have crafted numerous techniques for subverting them. Many modern techniques exploit programs such as Web browsers that are allowed access to the Internet, and then use the allowed program as a means to communicate with the victim machine. Because the personal firewall allows users to surf the Internet with a browser, the malware mimics a user's actions in the browser while polling an attacker for its commands and sending information taken from the victim machine. However, even though they can be bypassed, personal firewalls do provide some protection.

Microsoft bundled firewall technology in Windows XP Service Pack 2, released in 2004, and all subsequent versions of Windows. This personal firewall was designed in large part as a response to the significant number of worms that had infected Windows machines earlier in the decade. Although it was a very crude personal firewall in its configuration and capability, the protection thus made widely available helped to reverse an alarming rise in worm attacks from 2000 to 2004. The built-in Windows personal firewall is an example of how bundling security capabilities with an underlying operating system can help ensure that large numbers of consumers and enterprises have these protections. Other companies offer free or commercial personal firewalls that are far better than the fairly minimal capability offered to Windows users, which tech-savvy consumers and enterprises can deploy, but the Internet as a whole is better protected when these capabilities are built in.[16]

However, bundling security technologies into the underlying operating system has economic and political complexities. In the 1990s, for example, Microsoft bundled a Web browser into Windows but this resulted in significant antitrust litigation with the U.S. Department of Justice, several states, and some European countries. While the Windows personal firewall was largely successful, Microsoft has shied away from building anti-malware scanning capabilities into the operating system by default, perhaps fearing legal challenges from competitors in the security industry. Microsoft has released several anti-malware tools, some for free, but only as separate downloads that are not built into Windows. Microsoft has also altered recent versions of Windows, including Vista, to provide some capabilities of host-based IPS tools, such as preventing certain areas of memory usually associated with data from being used to run code, a technique often employed by malware. But these are only the lowest hanging fruit of host-based IPS capabilities, picked up by Microsoft as Windows evolves.[17]

Host-based encryption—file-level versus drive-level. Encryption technology can protect

data on a host either by encrypting individual files and directories or by encrypting the entire hard drive, including the operating system and software applications. File-level encryption is often faster than encrypting the entire drive. Drive-level encryption tends to offer better security, however: an attacker is less able to subvert the operating system and the encryption tool because the code for these programs is itself encrypted on the drive. Many organizations are deploying host-based encryption in light of the high number of cases involving loss of account information for millions of consumers due to theft of a laptop or compromise of a back-end database to steal the accounts.

Over 30 states and several countries have passed laws requiring disclosure of any breaches of personally identifiable consumer information. As of this writing, the United States does not have a Federal law regarding breach disclosure, but even without such a law, many breaches in the United States are disclosed nationwide. If a given e-commerce site has customers in a state that has a breach disclosure law, the organization is legally considered to be doing business in that state and must disclose the breach to its customers in the state. Most, therefore, disclose the breach to all of their customers, which makes such laws de facto a nationwide standard.

Many host-based encryption tools can be bypassed in various ways, which could have significant implications for breach disclosure laws. One method for bypassing cryptographic protections involves finding hidden temporary files created by the cryptographic tool during the encryption operation. For example, the built-in Microsoft Windows Encrypting File System technology leaves such files with the unencrypted data on the system until a user or administrator removes them manually or they are overwritten by another program, a fact that would come as a surprise to many users. Another method for bypassing host- based encryption tools involves exploiting an account or software running as a legitimate user on the system. Because the user or the software run by the user has the privileges and encryption keys to access the protected data, an attacker who compromises the account or exploits the software will gain exactly the same access to the data. In effect, because the encryption solution has to decrypt the data for legitimate users, an attacker can exploit software vulnerabilities to use those functions to retrieve the data.

A third way to bypass the encryption tools involves the attacker retrieving the decryption keys, usually stored on the system itself, often protected with a user's password. An attacker might be able to determine a user's password, either by guessing it or by launching a password-cracking tool, a process that might take a few minutes or many years, depending on how difficult the password is to guess for an automated tool that can try hundreds of thousands of guesses per second. An attacker who has determined the password can gain access to the data encryption keys and decrypt the data. For this reason, some encryption solutions do not rely exclusively on a user password, but augment protections with smart cards (credit card–sized computing devices that can store crypto keys), tokens, or biometric authentication. However, the majority of host-based encryption solutions deployed today rely exclusively on password protection.

If an attacker steals a laptop or gains access to a back-end database, the organization that suffered the breach is only required to disclose the breach if the attacker actually gained access to the consumer data. With host-based encryption solutions, a company may conclude that breach notification is unnecessary, given that the data was encrypted so the attacker should not be able

to gain access to it. However, the attacker might be able to bypass the encryption and read the consumer data. Thus, while host-based encryption does improve security, it could decrease the breach disclosure rate, making consumers unaware of violations of their privacy or potential compromise of their financial accounts.

Issues Applying to Network- and Host-based Defenses

This section describes some issues that apply to both network-based and host-based defensive approaches, including issues of patch management and the human factor. Their broad application offers significant opportunities for thwarting major attacks in cyberspace.

Patch management. Software vendors regularly release critical patches for their products that fix security vulnerabilities, either by tweaking the existing software or by issuing a whole new version of the product. The vast majority of security patches for consumer, enterprise, and infrastructure software products are downloaded through the Internet. Some systems, especially consumer machines, are configured to receive and install patches automatically. On other systems, such as those used by enterprises, a system administrator installs patches. This manual process slows down the application, but it allows administrators to vet patches carefully to make sure they will not break any enterprise applications.

Patch distribution through the Internet offers vendors the ability to disperse patches quickly but raises the chicken-and-egg problem of relying on the Internet to distribute patches for components of the Internet itself. If a major attack were to render the Internet unusable for a time, patch distribution could come to a halt. A few ISPs and other large enterprises have made plans for manual distribution of patches on physical media, such as CDs or DVDs with the software, carried by airplanes in the event of a catastrophic network failure. However, not all have done so.

Some vendors include a rating of the criticality of each patch to help organizations focus on the most severe vulnerabilities. Other vendors, including some associated with critical enterprise infrastructure applications, do not provide any criticality estimate but presume that their clients will install every patch they release. Unfortunately, because a bad patch can cause systems to crash or introduce additional security vulnerabilities, some enterprises choose to delay patching for issues of intermediate criticality, sometimes for many months. During that timeframe, the organization's systems are exposed to attack, and the organization may not even realize the threat if the vendor fails to specify a criticality rating with a patch.

Another concern with the state of software patching in cyberspace is the lack of vendor liability in most countries for security flaws in their original programs and for issues associated with patches. Most software contracts and license agreements explicitly disclaim vendor liability for flaws in their products, even blatant security vulnerabilities and the damage associated with exploited systems. The market does drive vendors to release patches, because of their customers' implicit threat not to repurchase or renew product license agreements. However, many software vendors have their customers locked in, as the customers' business processes are tightly intermingled with the vendors' software. This undermines market pressures to produce secure products. This situation creates incentives for vendors to push products out the door quickly, with plans for fixing security flaws later by means of patches. In the meantime, systems have exploitable vulnerabilities, and hundreds of new flaws are discovered and

publicized each month.

The human factor. Technological solutions may improve security, but only if solid security practices are followed by the human users and administrators of the systems. A user who reveals a password over the phone or is tricked into installing malicious software on a machine can undermine the most hardened enterprise or carefully configured operating systems. Enterprise administrators who fail to install a critical patch because they do not understand the security issues it fixes likewise leave their systems exposed to attack. Thus, user awareness education is just as vital a tool in protecting cyberspace as the latest firewall or encryption technology.

Many corporations and government agencies have, at best, rudimentary cybersecurity user awareness programs. Once a year, users may be exhorted to choose robust passwords, to avoid running untrusted programs downloaded from the Internet, and to avoid revealing sensitive information in email. While such advice is sound, enterprises handling sensitive data or operating critical infrastructures need to strive for a culture of information security, not just a yearly reminder that users quickly forget. Regular reminders of security practices are vital, as well as periodic audits to ensure that those practices are being followed. Given their access to and control over vital computing assets, system administrators and other information technology professionals are among the most important enterprise staff members to educate in comprehensive security practices. Enterprises themselves should require cybersecurity education of their system administrator staff, potentially with government requirements or oversight.

The sorry state of information security awareness for the public at large is an even bigger problem than the relative lack of security awareness in enterprises. Operating system and security software vendors may incorporate more and more defensive technologies into their products, but they are fighting a losing battle unless the public can be trained to use them. The large numbers of users who fall for phishing scams, lack anti-malware tools, run unpatched systems, and choose easily guessed passwords for their accounts indicate that the public is either not aware of sound security practices or does not understand the threats. The state of information security in cyberspace could be significantly improved by public service announcements and education campaigns. Like the anticrime, environmental awareness, and antismoking television ad campaigns of recent years, a comprehensive and repeated program of public awareness could help instill fundamental security principles to make cyberspace safer and more secure. Some agencies (including the Department of Homeland Security in the United States) and countries have experimented with small-scale user awareness initiatives for the public, but more and broader initiatives are necessary. Such awareness programs should point out that securing one's own computer not only lowers the risk for that individual, but also helps improve the security of cyberspace and the country as a whole.

Conclusion

The security concerns associated with today's Internet are based on the rapid evolution of technology, applied in ways unanticipated by its original designers. The Internet has grown considerably beyond the scope of the original experimental ARPANET. The Internet's

architects did not design the network or its protocols to handle the level of sensitive data and economic activity that they routinely carry today. The network has scaled to hundreds of millions of users around the globe, a vast and diverse population. The computers that the Internet connects typically use general-purpose processors and operating systems that can run any program presented to the machine, making them flexible and extendable. However, this flexibility results in the possibility of infection by numerous varieties of malicious code, such as viruses, spyware, worms, and bots. As significant vulnerabilities are routinely discovered in workstations, servers, and network equipment, and large numbers of malicious code specimens are introduced every day, the state of Internet security is cause for concern.

Today, small-scale attacks are commonplace. Attackers have the technical capabilities, but usually not the financial motivation, for large-scale attacks. Over the past two decades, the threat landscape has increased from experimental hackers and hobbyists to include organized cybercriminals. We may face a further evolution of the threat as terrorist groups and nation-states seek to utilize cyber attacks as a form of warfare.

However, as various threats grow and vulnerabilities proliferate, security technologies have been developed for the Internet and the computers it interconnects. These technologies can provide a good degree of security if they are judiciously deployed and carefully maintained by system administrators and users who are informed about good security practices.

---

[1] The transmission control protocol and Internet protocol are standards that make the Internet possible.

[2] For the purposes of this chapter, differentiating a medium scale is not necessary. Intermediate-sized attacks can be considered either smallish large-scale attacks or big small-scale attacks.

[3] Data from the Anti-Phishing Working Group, a nonprofit organization created to track phishing attacks and educate users in methods for avoiding such scams. See <www. antiphishing.org>.

[4] A nontechnical discussion of the Estonian attack is Mark Landler and John Markoff, "Digital Fears Emerge After Data Siege in Estonia," *The New York Times*, May 24, 2007. For technical details of the attack, see Beatrix Toth, "Estonia Under Cyber Attack," at <www.cert.hu/dmdocuments/Estonia_attack2.pdf>.

[5] The 13 root servers are housed in multiple sites worldwide; some are single-site installations with a single machine constituting the DNS server, while others use a technology called *anycast* to have multiple distributed machines function as a single root DNS server. The root servers that rely on anycast are less prone to outages from packet floods because they distribute the load across multiple separate machines. See <http://root-servers. org/>.

[6] ICANN Fact Sheet, "Root Server Attack on 6 February 2007," March 2007, available at <www.icann.org/announcements/factsheet-dns-attack-08mar07.pdf>.

[7] Legitimate TCP traffic follows the pattern SYN, SYN–ACK, ACK, followed by a connection. SYN floods have a SYN and a SYN–ACK, but no completion of the three- way handshake or the follow-on connection.

[8] Examples of some significant router vulnerabilities include: May 2001, vulnerability in routing protocol update (using the Border Gateway Protocol) could have been used to crash router (see <www.kb.cert.org/vuls/id/106392>); August 2003, four specially crafted packets could have stopped routers from routing (see <http://nvd.nist.gov/nvd. cfm?cvename=CVE-2003-0567>); April 2004, method for resetting routing updates could have made communications gradually decay as routers could not receive network topology updates (see <www.us-cert.gov/cas/techalerts/TA04-111A.html>); July 2005, method found for exploiting router coding flaws that lets an attacker control router (see <http://nvd.nist.gov/nvd.cfm?cvename=CVE-2005-3481>); May 2007, error in crypto library in major routers could have been used to take over routers (see <http://nvd.nist. gov/nvd.cfm?cvename=CVE-2006-3894>).

[9] For technical details of this attack, see <http://kerneltrap.org/node/3072>.

[10] See <www.wired.com/politics/security/news/2005/08/68365>.

[11] Examples of significant domain name system (DNS) server vulnerabilities include: November 2002, DNS vulnerability could have let attacker take over DNS server (see <http://nvd.nist.gov/nvd.cfm?cvename=CVE-2002-0029>); April 2005, DNS cache poisoning attacks could have let attackers redirect traffic by tricking servers into loading bogus DNS records (see <www.ncs.gov/library/tech_bulletins/2005/tib_05-4.pdf>); April 2007, protocol used for management of Windows DNS servers was vulnerable, allowing for takeover of DNS server (see

<www.us-cert.gov/cas/techalerts/TA07- 103A.html>); July 2007, flaw in DNS server allowed attackers to load bogus record and redirect traffic (see <www.isc.org/index.pl?/sw/bind/bind-security.php>).

[12] Examples of such protocol-converting gateways include IP-to-IPX converters, as well as IP to Signaling System 7 gateways that convert IPv4 to the protocol used to control public telephone network switches.

[13] See, for example, Seth Mydans, "Monks are Silenced, and for Now, Internet is Too," *The New York Times*, October 4, 2007.

[14] The Internet Engineering Task Force calls itself "a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet." See <www.ietf. org/overview.html>.

[15] The security of a certificate authority (CA) depends on how carefully the CA checks a given user's or enterprise's identity before issuing a certificate and how the CA protects the encryption keys used to sign the certificates. If a CA were to issue a certificate to an imposter that claimed, for example, to be a government agency, all users who relied on that CA's certificate would be exposed to the imposter.

[16] Similarly, Microsoft built rudimentary file encryption technologies into Windows 2000 and later with a feature called the Encrypting File System (EFS). Other companies offer far superior encryption functions, but because a baseline capability is available for most Windows users, consumers can choose to use these security tools. While the Windows personal firewall is in widespread use because it is activated by default, EFS is seldom used, likely because it is off by default.

[17] Microsoft's decisions about which security features to bundle into Windows and which to leave to third-party vendors require careful balancing of the interests of the company and its competitors, regulators, enterprises, and consumers.