



Romel Cabiling ▾



[Home](#)

[Home](#) > [My courses](#) > [Network Defense and Remote Access Configuration](#) > [14 \[Enter Module Title Here\]](#) > [Lesson Proper for Week 14](#)

Lesson Proper for Week 14

INTERNET AND WORLD WIDE WEB SECURITY

Examining the Structure of the Internet

Internet use has increased exponentially in the past 10 to 15 years. Government organizations use it to streamline services and communications, for example, and businesses use it to conduct transactions and marketing operations. For many organizations, the Internet is indispensable in today's competitive marketplace.

Opportunists seek ways to exploit poorly designed systems on the Internet. Whether the aim is political, criminal, driven by greed, or just inspired by curiosity, the cycle of offense and defense is constant: Attackers discover new exploits, and then hardware and software vendors distribute notifications and patches to defend against these exploits. In the following sections, you examine the structure of the Internet and see how administrators and users can minimize risks.

Understanding the Structure of the Internet

The Internet is a group of networks tied together to form an infrastructure for communication. The terms Internet and World Wide Web (WWW) are often used interchangeably, but they are quite different. The Internet, a massive public medium established in the mid-1960s, is an interconnected web of networks and computers that work together to provide worldwide communication. The World Wide Web, which uses Hypertext Transfer Protocol (HTTP), is just one of the services the Internet offers. Many other services are offered, such as e-mail, which uses Simple Mail Transfer Protocol (SMTP) for communication, and file transfer, which uses File Transfer Protocol (FTP). The Web is a method of accessing information through the Internet by using HTML hyperlinks. It uses Web servers, Web browsers, and Web pages to communicate information through the Internet network.

The original Internet backbone was based on the U.S. Defense Department's ARPANET infrastructure and later the National Science Foundation's NSFNET, both of which relied on interconnected Internet backbones that support a distributed mesh topology. This early network defined the hierarchy of networks that is still in operation today.

By the early 1990s, the National Science Foundation (NSF) decided to stop funding NSFNET and move toward commercialization of the Internet. Many of the NSF's regional networks became commercial network service providers (NSPs), such as Netcom, UUNet/MCI World- com, and PSI Net. Also known as backbone Internet service providers, these NSPs expanded their backbone networks. To maintain their government funding, the NSF required the providers to allow free flow of traffic from one backbone to the other.

Tier System

The Internet is a tier system that connects networks around the world. This system starts with a backbone network connected via network access points (NAPs) to regional Internet service providers (ISPs). Regional ISPs service point of presence (POP) ISPs that connect to business, education, or home networks. The Internet is composed of a collection of backbones serviced by major carriers. The following sections explain NAPs and ISPs in more detail.

Routers and the Internet Communication Backbone

The Internet communication backbone is an interconnected network of backbones owned by businesses or NSPs. Much like an anatomical backbone, where nerve signals travel across the spinal cord, an Internet backbone provides a conduit for network communication between different points of Internet access.

Routers direct network traffic to its destination via routing tables and updates from routing protocols. Routers in NSP backbones differ from routers in a LAN by the high amount of traffic they are designed to handle and the routing protocols they use for the Internet back- bone environment. The physical memory, CPU speeds, interfaces, and operating systems (OSs) of routers used in NSP backbones can support enormous amounts of traffic and large routing tables.

Network Access Points

Network access points (NAPs) are highly secure public facilities in which backbones have interconnected data lines and routers exchanging routing and traffic data. NAPs provide physical space, power, and network connectivity between different levels of the Internet's tier system, such as between a regional ISP and a POP ISP or between a backbone and a regional ISP.

Backbones exist in different regions of the world, and NAPs are positioned in each country to provide interconnectivity between these backbones. Each NSP backbone exchanges routing and traffic data in one of two ways: via NAPs or private peering relationships (see Figure 12-1). Private peering relationships are contracts between commercial NSPs or ISPs that enable them to bypass the Internet backbone for data and route exchanges.

Internet Service Providers

An ISP provides access to the Internet at different levels depending on the type of ISP. A local ISP or POP ISP provides Internet access directly to consumers or businesses. A regional ISP sells bandwidth to local or POP ISPs or to organizations with high bandwidth requirements. A backbone ISP or NSP gives regional ISPs back-bone access. If an ISP is large enough, it might offer services at all these levels, from local/ POP access to backbone ISP.

Domain Name System

Domain Name System (DNS) is a name-resolution service that translates fully qualified domain names, such as www.cengage.com, to IP addresses used to identify host computers, such as 69.32.133.79. Thirteen root servers named A through M are operated by commercial, educational, and government organizations to form the foundation of the Internet DNS. These servers are often targets of attack.

DNS is a hierarchical system, as shown in Figure 12-2. Root servers know which servers on the Internet are responsible for top-level domains (such as .org or .com). Each top-level domain has its own servers that delegate responsibility for domain name-to-IP address resolution to name servers lower in the hierarchy. In reality, most DNS information on the Internet is cached, so DNS lookups rarely have to go all the way up the hierarchy to root servers.

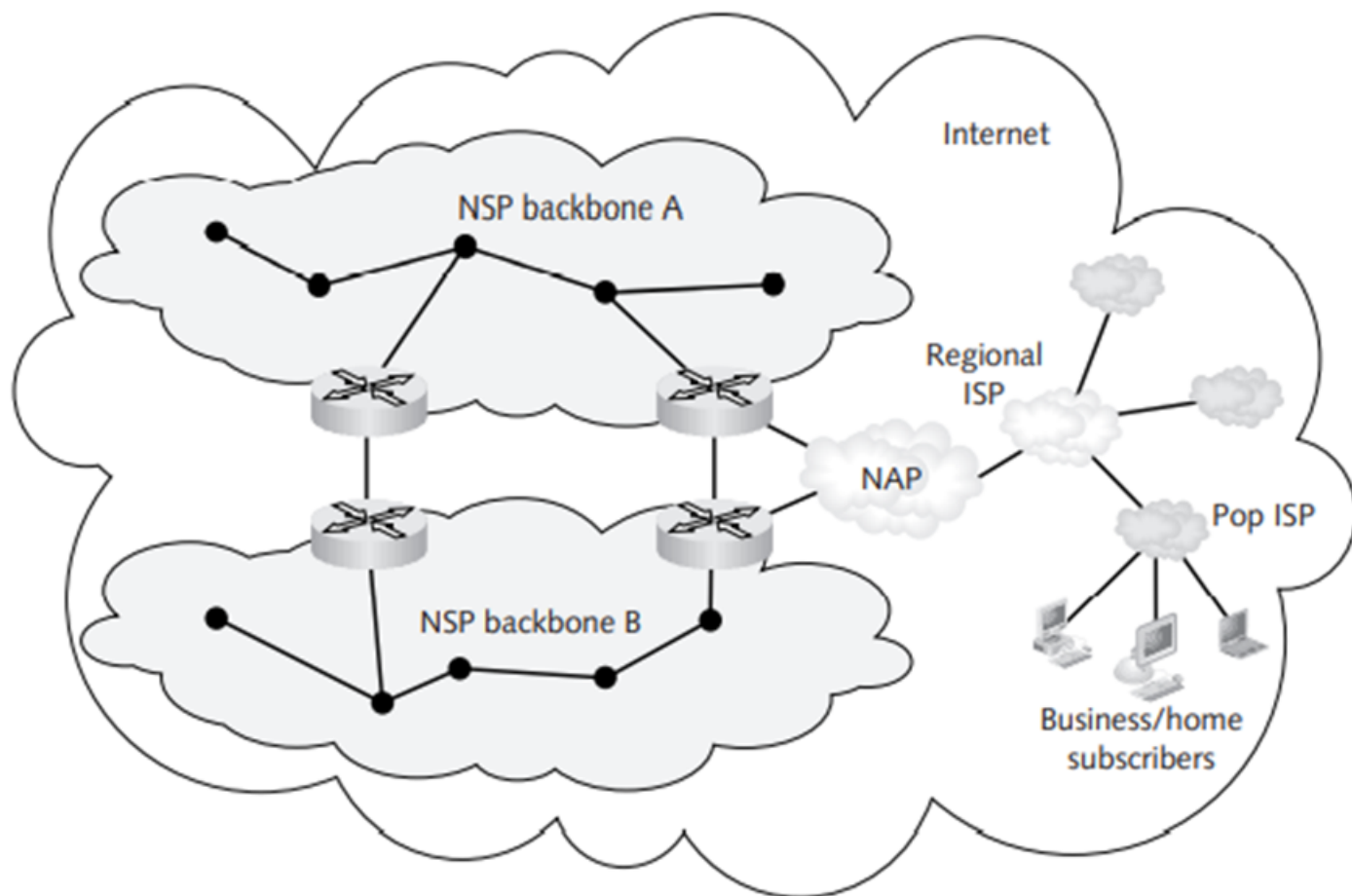


Figure 12-1 The Internet backbone: a network of NSP backbones

The 13 root servers were located in the United States originally. However, servers C, F, I, J, K, and M are on other continents now, which is possible because they make use of anycast addressing. Unlike unicast and multicast addressing, anycast addressing enables any group of servers to act as a root server, regardless of location. Anycast addressing is a way of decentralizing DNS services, and balancing the load among several servers improves availability

You can learn more about the DNS root server system at www.root-servers.org.

At ISPs, a local DNS server replicates entries from servers that are higher in the DNS hierarchy and resolves DNS requests. This server also forwards queries up the hierarchy if it cannot resolve a request with its own DNS table or cache.

Understanding Weak Points in the Internet's Structure

For all the usefulness the Internet offers, risks are inevitable when people and organizations operate in this environment. Attackers constantly discover new ways of exploiting the Internet infrastructure, and IT professionals must often play a catch-up game to stay ahead of attackers' exploits. In the following sections, you examine some techniques that attackers use to exploit weaknesses in the Internet's structure.

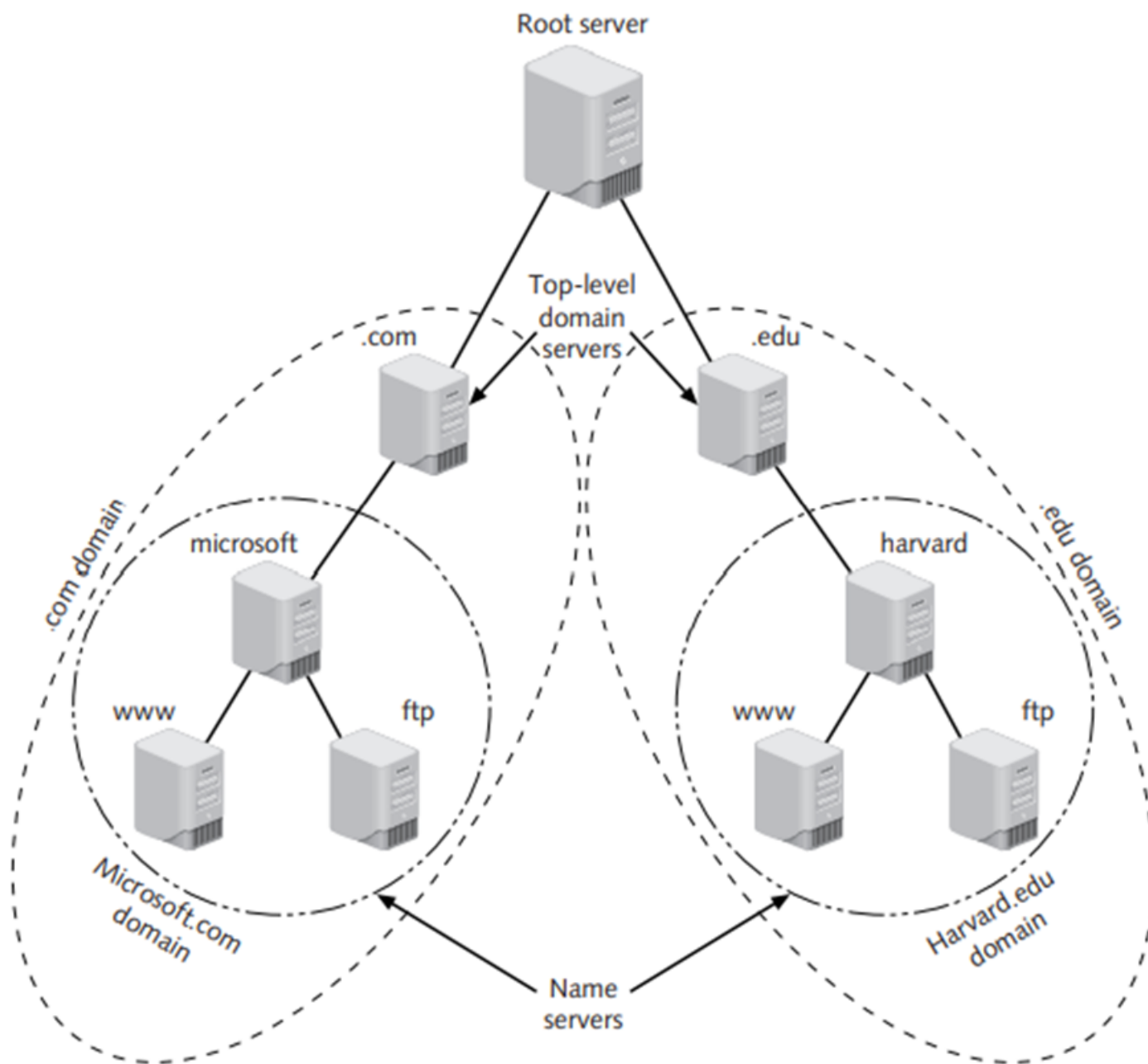


Figure 12-2 DNS hierarchy

IP Spoofing

Computers on the Internet are identified mainly by their IP addresses, which are not authenticated by TCP/IP. This lack of authentication makes IP spoofing possible. Attackers change the source IP address in the headers of malicious packets they are sending to match a trusted host's IP address. To find a legitimate IP address, attackers sometimes send ping packets into a network and wait for responses. IP spoofing is most often used in denial of service (DoS) attacks. Attackers do not care about receiving responses to their packets; they want to trick a network's defense systems into accepting packets so that they can flood the network with packets and cause it to crash. Attackers sometimes use IP spoofing to fool servers into sending responses to their forged addresses, however. Because IP spoofing is so widespread, it makes accountability difficult for malicious actions on the Internet; someone who has intentionally accessed a restricted site can claim to be the victim of IP spoofing.

Modern network routers and firewalls offer software protection against known forms of IP spoofing. Packet filtering through routers is a major defense, for example. It includes ingress filtering to prevent spoofed IP source addresses from entering a network and perhaps egress filtering to prevent users inside a network from sending spoofed packets outside the network. In addition, IPv6 offers improvements to prevent IP spoofing, such as authenticated headers in each packet.

Routing Security

The Internet network is linked by routers. Routing protocols such as Border Gateway Protocol (BGP) are used to communicate information updates for routing tables. However, routing information is not authenticated, so it is vulnerable to compromise. Attackers could send modified routing updates that misdirect data to a destination of their choosing. With this data, they can launch DoS attacks, use IP spoofing to intercept packets, or launch man-in-the-middle attacks, in which the attacker's computer or router is placed between the source and destination of a communication to intercept and steal information.

DNS Security

DNS was originally designed as a public database for name-resolution services, so checking the authenticity and integrity of information stored in name servers was not considered necessary. This lack of security has caused several problems for the Internet community.

One problem is DNS cache poisoning, also known as DNS spoofing. When a name server is queried for DNS information that is not in its cache, it queries other servers. Because DNS information is not authenticated, attackers can send false data to a name server; the DNS cache is said to have been "poisoned." Attackers often use cache poisoning to steer unsuspecting victims to a server of their choice instead of the Web site where users intended to go.

A DNS name server for an organization contains database entries about every host on the network. Another problem caused by lack of authentication is DNS information leakage, which might occur if attackers access the database and use it to map target systems in the network. This information can be partially secured in DNS by blocking zone transfers, which replicate a name server's DNS information to other servers. Attackers could still retrieve the information by using DNS tools to query systems in an organization's IP name space one by one until they capture a complete listing of DNS information. However, this method is slower and more painstaking.

Internet Host Security One of the bastions of Internet interconnectivity is the millions of host computers worldwide, but ironically, they are the weakest point of the Internet infrastructure. Attackers hijack many unprotected computers around the world and use them as "zombie" computers to deliver spam e-mail, DoS attacks, and malicious code. Attackers often assemble these zombies into botnets (networks of zombie computers) to magnify the scope and intensity of their attacks. According to M86 Security Labs, 91 percent of spam e-mail sent in May 2012 was delivered by hijacked zombie computers.

For the Internet and its Web component to function, many computers must be connected globally into one network. Each computer differs in the way it has been prepared to handle dangers on the Internet. For example, the risks that a careless Internet user takes in the United States, China, or South Africa can have dire consequences for careful users whose host computers are also connected to the Internet. The dangers of virus, Trojan, or DDoS attacks on your system might exist simply because you are connected to the Internet. Therefore, good computing practices to minimize risks, such as antivirus software, firewalls, and system patches, are essential to withstand attacks from the Internet.

Web Site Attack Techniques

To exploit the Internet's weaknesses, attackers use a variety of innovative techniques. In the following sections, you examine how attack techniques are constructed and then learn how to best defend against them. First, you examine attacks against Web servers, and then you learn about attacks on the client side of the Internet: Web browsers and e-mail applications.

Attack Techniques against Web Servers

The World Wide Web operates on the principles of a client/server network, and its basic building blocks are Web servers and client computers. Because millions of these network components are distributed around the world and their hardware and software configurations are so similar in nature, they are the Internet components that attackers target most often. Attackers probe common hardware/software server configurations, such as Windows running Internet Information Services or Linux running Apache Web Server, in an attempt to discover security holes.

Sensitive transactions, such as banking and e-commerce, are commonplace now on the Internet, so attackers often select Web servers that handle these transactions as targets for identity theft. The following sections explain ways that a Web server can be compromised.

Buffer Overflow Attacks

A buffer overflow attack exploits software vulnerabilities over which users and even network security personnel have little or no control. These common attacks often come with no warning and are almost impossible to detect and fix. These attacks have been used since the mid-1980s, when attackers discovered how to manipulate computer memory remotely by using worms and Trojans.

Generally, commercial OSs, Web servers, and databases are more vulnerable to buffer attacks than customized software that companies create for internal use. The source code is wrapped in a "black box" to protect it from tampering, but many attackers have the skill to access this code. News of pinhole vulnerabilities in black boxes travels quickly through the hacker

community, and decompile tools for commercial software are readily available. After attackers have access to an application's code structure, they can look for weaknesses and errors in the source code—the root of buffer overflow attacks.

SQL Injection Attacks

Structured Query Language (SQL, pronounced sequel) is used to communicate with most relational database management systems (RDBMSs), such as Oracle, MySQL, SQL Server, and DB2. Because SQL is used so widely, particularly in e-commerce databases, it is a favorite target of attackers for data theft and destruction. As you have learned, buffer overflows are a result of poor coding. The same is true for SQL injection attacks: Web sites that have not been sanitized correctly are vulnerable to attack. The term sanitized describes computer applications or processes that have been protected against attacks.

A buffer overflow attack requires programming expertise, which limits the number of potential attackers. SQL injection, however, is plaintext scripting that is easy to learn and apply, making it a favorite language for fledgling attackers (often called script kiddies). No special tools are needed—just a computer connected to the Internet, a Web browser, and patience. The good news is that, unlike buffer code written by a third-party programmer, Web pages usually consist of custom-written code. Therefore, coding vulnerabilities can be fixed if they are detected in time.

SQL injection does not attack a Web server directly. It attacks the database used to support Web sites housed on the Web server, and more sophisticated attacks can be extended to attack the database server and its partner Web server. The next sections examine two SQL injection attack methods and their effects on database and Web servers.

SQL Injection: Web Form Attacks

Web forms used to gather information, such as login pages or order forms, are potential entry points for attackers probing for Web site vulnerabilities. These forms are usually connected to a Web server's database, and a verification process checks information entered in the form and rejects incorrect entries. If the form's entry text boxes are not verified correctly, however, attackers can use them to send malicious code to the database, the database server, and perhaps even the partner Web server.

SQL Injection: Query String Attacks

The second method used for SQL injection attacks involves the query string used to send information to a database. When a user clicks a link on a Web page, information is sent to the Web server. For instance, on a retail site, a user might click a product picture to see more information, which is stored on a database that supports the Web server. The product's ID code and perhaps other information is attached to the Web page address and sent to the Web server for action. The information being sent is clearly visible in the browser address bar and can be the source of an attack on the Web site's database.

Defenses against SQL Injection Attacks

SQL injection attacks are isolated to custom applications, so administrators can prevent them, unlike buffer overflows, which require third-party vendors to make code adjustments. The first course of action is to prevent malicious code from being entered in Web pages that allow user input. A common mistake is for site administrators to stop there and take no further protective measures. Attackers can exploit SQL in many other ways, however, so you should take the following steps to close all potential holes:

- Tighten database authentication and limit table access. Always require password access to the database, and never leave default usernames set up during installation. Most attackers are familiar with the default administrative username sa, so make sure to change it.
- Use stored procedures to eliminate passing any SQL commands to the database.
- Validate all user entries to make sure they are formed properly. Perform this validation in several places if necessary. There should be two layers of validation: form-level validation at the browser before the Web page is submitted and server-level validation when the information reaches the server for processing.
- Place the Web server and database server in a network DMZ.
- Use nonstandard naming conventions in database construction. To thwart attackers, you should make database names, table names, and field names difficult to guess.
- Inevitably, database errors do occur, so configure a custom error message that does not reveal information for attackers to exploit. The standard 404 error message often reveals server information that attackers can use.

With these simple precautions, you can immunize a Web server and its database server against SQL injection attacks.

Attack Techniques against Web Users

In the previous section, you learned about attack techniques that are directed against computer or network systems. Now you examine attacks directed at Web users through commonly used applications, such as Web browsers and e-mail programs. These attacks fall into the category of social engineering because they prey on emotions such as curiosity, anxiety, fear, and greed. Unlike some attacks against systems, almost all attacks against Web users can be prevented.

Every user takes on a measure of security risk when interacting on the Web. Attacks on Web users center on a variety of objectives, including identity theft and simple malicious behavior. Informed Web users should understand these attack methods and know how to prevent them. In the following sections, you explore some of these risks and examine measures that can eliminate or at least minimize the risks.

Phishing Attacks

Phishing is an attack through a Web browser that displays false information masquerading as legitimate data. Phishing is a deception designed to steal personal information such as credit card data, account numbers, usernames, and passwords.

Phishing attacks can take many forms that range from simple to quite sophisticated. A simple form that has persisted for many years is the Nigerian money scam. The perpetrator sends millions of e-mails to a random selection of addressees, asking for help in transferring a large sum of money from Nigeria to the United States. To be "rewarded," the e-mail recipient must provide personal banking information to assist in making the transfer. The scam has grown in popularity to include letters from China, North Korea, and Russia. The letters are almost comical in their wording, and it is hard to imagine that anyone would take them seriously. People are still taken in by this scam, however.

File Attachment Attacks

E-mail attachments are a common vehicle for introducing malicious code into a network. These attacks first occurred in 2002 when JPEG attachments were discovered to have virus code embedded in the file header code. Until then, data files had been relatively immune to infection.

The attack requires two virus components. The first part spreads in the form of a traditional Win32 executable virus, arriving via e-mail or portable media. This virus makes changes to the Registry so that JPEG files are run through an extractor before they are displayed. The virus strikes if the user attempts to view a JPEG image and the extractor finds the second virus component in the graphics file header. Having both parts on the same computer is rare, however, because standard virus protection typically detects the presence of a virus arriving via e-mail or portable media. However, users should be cautious about viewing image file attachments from unknown sources.

ActiveX Control Attacks

An ActiveX control is a Windows object coded in languages such as C++, Visual Basic, and Java. Its purpose is to deliver dynamic, interactive content to Web pages. A control object is compiled and stored in a CAB file, which is stored on a Web server and accessed by referencing the object's assigned CLASSID, as shown in the following code from the Developer Connection QuickTime site:

```
<OBJECT CLASSID="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B" WIDTH="160" HEIGHT="144"
```

```
CODEBASE="http://www.apple.com/qtactivex/qtplugin.cab">
```

```
<PARAM name="SRC" VALUE="sample.mov">
```

```
<PARAM name="AUTOPLAY" VALUE="true">
```

```
<PARAM name="CONTROLLER" VALUE="false">
```

```
<EMBED SRC="sample. mov" WIDTH="160" HEIGHT="144" AUTOPLAY="true" CONTROLLER="false"
```

PLUGINSOURCE="http://www.apple.com/quicktime/download/">

</EMBED>

</OBJECT>

When used for legitimate purposes, an ActiveX control can be a beneficial addition to a Web site; however, attackers have discovered that an ActiveX control can be programmed to run malicious code on a user's Web browser. ActiveX controls do not require user action to be activated. They run automatically when the browser loads the Web page that contains them. ActiveX controls have almost full access to the Windows OS and can perform many functions, including running code on an unprotected computer, which could involve accessing and downloading files, planting Trojan programs and worms, or destroying system programs.

The defense against malicious ActiveX controls is to make sure that you scrutinize them by using security settings on Web browsers. Browsers can be set to block ActiveX controls from running on Web pages, for example. You can also adjust browser settings to permit certain types of ActiveX controls to run and block others.

Java Applet Attacks

A Java applet is a small program sometimes used as embedded code in Web pages. Java applets were considered immune to hacking because they were encased in a "sandbox" that could not interact with a host computer outside the confines of a browser; they could communicate only back to their code base. However, Java applets have been used with Internet Explorer and Netscape to exploit the OS and access system files. In the Internet Explorer attacks, malicious code embedded in a Java applet was used to exploit a proxy server network connection. The user's session was then redirected to a location of the attacker's choice without the user's knowledge, and the attacker was able to capture the user's information.

In the Netscape attacks, vulnerabilities in Netscape Communicator and Navigator made it possible for Java applet code to gain unauthorized local and remote file access. A malicious Java applet could read files from the local file system by opening a connection to a URL and gaining complete file access. The applet could then send files back to the server from which it originated.

Using this method, the connection is reversed, and the user of the browser sends information to the applet originator. This communication reversal negates the protection of the user's firewall, which watches for incoming vulnerabilities, not outgoing ones.

Although the combination of circumstances in these attacks is rare, it does emphasize the need to patch your system with the most recent updates and hot fixes. Furthermore, only signed applets should be permitted to run on Web browsers. In the Java applet exploits discussed in this section, software patches from vendors fixed the problem.

Hardening Web and Internet Resources

Establishing and maintaining a hardened network with secure hosts requires continuous vigilance and regular updates of components. New versions of software, hardware, and network media are released frequently, but the threats against networks and systems change just as often. A network security administrator alone cannot keep up with the daily deluge of security concerns. By enlisting the help of security experts and adopting a preventive stance toward network security, the task becomes far less daunting.

Seeking the assistance of security experts need not be an expensive venture. Most of the help you need is free. Your first stop should be the supplier of your firewall and antivirus software. All reputable vendors maintain informative Web sites with excellent guidelines for how to best use their products. They also offer automatic, timely downloads of the latest virus signature databases so that your systems are protected as soon as possible. With your network systems protected, you can then push the updates to all connected host computers automatically.

Hardening DNS Servers

When attackers probe networks to look for vulnerabilities, they pay special attention to servers that host Internet services, such as DNS or Web servers, because they store valuable personal and corporate information. This section explains techniques for hardening these servers against attacks.

A primary DNS server is authoritative for specific domains and has DNS zone files that change as needed. A zone file is a set of instructions for resolving domain names into IP addresses. A secondary DNS server receives a read-only copy of the zone file to improve the query performance of DNS services. An internal DNS zone file contains entries of all internal hosts on a network, and an external zone file contains only host entries that are visible to the public. A zone transfer occurs when a zone file is sent from the primary DNS server to secondary DNS servers for updating. A zone is just an abbreviated way of referring to the domain name for which a DNS server is configured. If the domain name is myschoolsite.edu, for example, its components are the name (myschoolsite) and the generic top-level domain (edu). A subdomain might be staff.myschoolsite.edu, for example. Figure 12-6 shows an example of a small zone file.

```
C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32\dns>type myschoolsite.edu.dns

Zone File for myschoolsite.edu
myschoolsite.edu IN      SOA      ns1.myschoolsite.edu. hostmaster.myschoolsite.edu.
<
20080211      ; serial number
1000      ; refresh
600      ; update retry
172800      ; expiry
14400      ; default TTL
>

Zone NS records
      IN      NS      ns1.myschoolsite.edu.
      IN      NS      ns2.myschoolsite.edu.
      IN      NS      10      mail.anotherdomain.edu.

jenn      IN      A      192.168.10.1
www      IN      CNAME      jenn
```

Figure 12-6 A zone file for myschoolsite.edu

If zone transfers are not secured, attackers might be able to intercept them and retrieve a complete listing of network resources and possible targets for attack. One of the most serious mistakes a network administrator can make is to allow untrusted Internet users to perform zone transfers. Transfers should be allowed only between primary and secondary DNS servers. If the DNS server does not use a segregation method to separate external DNS information from private internal information, internal IP address and host name information could be exposed to attackers, who would then have an electronic road map of the organization. The following example shows probing code that an attacker might try as a starting point:

C:\Nslookup

This code might yield the following information:

Server: somedomain.com Address: 10.10.10.10

With this information, the attacker could attempt to change the server to the network's primary DNS server and then list and pull records from it.

Securing zone transfers is usually straightforward; you simply configure all DNS servers to restrict zone transfers to specific authorized servers. Using selective transfers minimizes the risk of unauthorized users getting a copy of a zone file.

If your organization has a DNS server that is authoritative for your domain on the Internet, make sure that DNS servers are in a DMZ and that a split DNS architecture is used. A split DNS architecture physically separates public DNS servers from the organization's internal DNS servers. Public DNS servers are used for authoritative DNS services to the Internet. In addition, internal DNS servers use a non-Internet domain name, such as .corp or .local; the authoritative DNS servers conform to Internet domain requirements. A split DNS architecture like the example in Figure 12-7 prevents internal zone information from being stored on an Internet-accessible server and prevents internal DNS entries from being sent over Internet DNS.

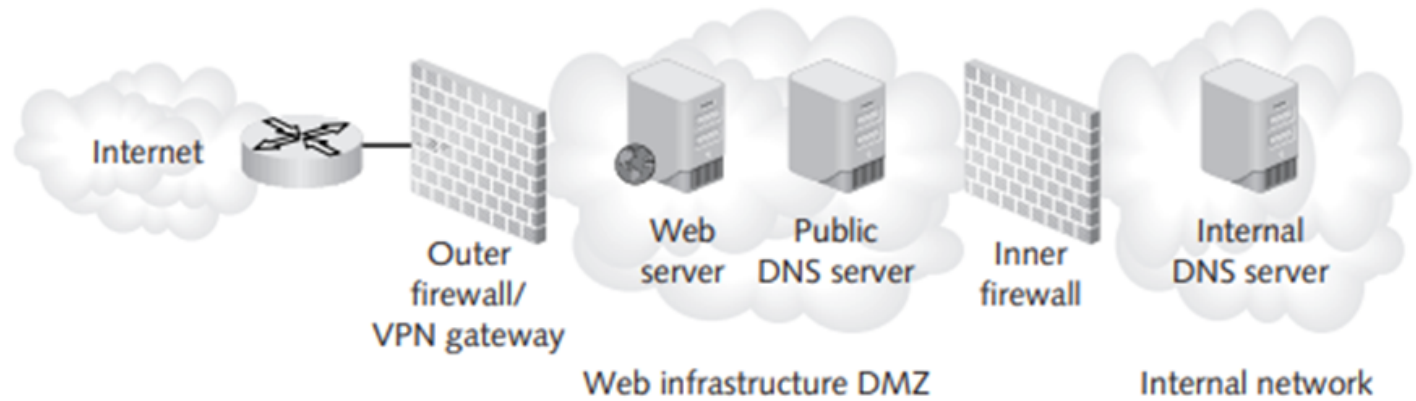


Figure 12-7 A split DNS architecture

A variation on this split DNS architecture is called split brain DNS architecture. In this system, a physical separation still exists between internal and external DNS servers, but both DNS systems use the same domain. While this system creates less confusion for internal users because they do not have to distinguish between internal and external DNS domains, it puts added responsibility on the DNS administration team, who must make sure that all references to internal hosts are removed from the DNS zone maintained in the DMZ.

Another layer of DNS security is at the network perimeter, where you should configure firewalls and routers with rules or filters to prevent zone transfers to the Internet. Zone transfers typically occur over TCP port 53.

DNSSEC

The DNS protocol was designed long before the designers of the Internet became concerned about security. Not surprisingly, the protocol has been found to be vulnerable to exploitations, as you learned earlier in this chapter. These exploitations can result in loss of confidentiality (tapping Voice over IP), Web site impersonation, e-mail hijacking, DNS cache poisoning, and theft of information, including logon credentials and credit card data. DNS Security Extensions (DNSSEC) was created to thwart some of these attacks. The goals of

DNSSEC are to provide authentication of DNS data, ensure integrity of DNS data, and authenticate the denial of existence of DNS data. It is important to note that DNSSEC addresses only these goals; it does not provide message confidentiality or protect against DDoS attacks.

DNSSEC uses cryptographic techniques to provide security for DNS data. Digital signatures are created and stored in a new type of DNS resource record—an RRSIG record—as part of the process of creating a signed DNS zone. A security-aware resolver is a system that is compliant with DNSSEC and that attempts to use a DNS server to resolve a fully qualified domain name to an IP address (or vice versa); this system can access the RRSIG for the DNS zone in question. To authenticate the address resolution, the resolver can access the public key associated with the RRSIG record by accessing the DNSKEY resource record where the public key is stored. A Public-key Infrastructure (PKI) exists to provide a chain of authentication for these keys. Just as a system that uses typical asymmetric encryption must be configured with at least one root key, a security-aware resolver must be configured with at least one DNSSEC trust anchor. A trust anchor is the top-level digital certificate in the PKI chain.

Larger ISPs have begun the process of implementing DNSSEC. This development is important because most DNS queries are serviced by ISP-maintained DNS resolvers. For example, Comcast, a major ISP in the United States, began migrating customers to DNSSEC resolvers in late 2010. By January 2012, Comcast had created signed DNS domains for all the domains under its control and had migrated all customers to DNSSEC-validating resolvers.

As you learned earlier, DNSSEC does not provide data confidentiality. Another weakness is that an attacker may be able to enumerate the contents of a DNS zone by following the NSEC resource record chain. An NSEC resource record is the Next Secure record that allows a resolver to trace the authentication path of the RRSIG. Also, DNSSEC is considerably more complicated than traditional DNS, which increases the possibility of errors and therefore a loss of service. Finally, the effectiveness of DNSSEC depends on unbroken chains of authentication. Until all Internet DNS zones are DNSSEC compliant, there is no assurance that the goals of DNSSEC can be met.

Hardening Windows Web Servers

Regardless of its platform, a Web server is usually secured by hardening the underlying OS, installing patches, disabling unused services, and restricting the number of user accounts and their access permissions. In addition, you can use platform-specific software tools.

Internet Information Services (IIS) is the Web server used in Windows 2000, Windows XP Professional, Windows Server 2003 and 2008, Windows Vista, and Windows 7.

Authentication

Authentication is an important consideration in all forms of information security, but considering that Web servers are often accessible by untrusted users, it is particularly important when configuring Web server security. IIS 7 allows you to select one of two forms of authentication:

- Challenge-based authentication, in which the Web client must respond to a challenge from the Web server. An example is Integrated Windows Authentication, in which Active Directory credentials are used.
- Login redirection-based authentication, in which users must enter credentials on a login page. IIS 7 also supports SSL-based digital certificate authentication.

Many Web servers do not formally authenticate users; they permit anonymous authentication. E-commerce sites, for example, want users to access their Web sites without authentication until the users intend to buy merchandise. In IIS 7, these users are logged in with the account IUSR.

Windows Basic Authentication requires that users enter a username and password, but it is not browser specific. The downside is that it transmits passwords in plain text. Windows Digest Authentication uses Active Directory to authenticate users, but the client browser must support the HTTP 1.1 protocol. Windows Authentication supports both Kerberos and NTLM (New Technology LAN Manager, a legacy authentication method). Extended Protection is an authentication method available in IIS 7.5. Extended Protection is designed to decrease the risks associated with man-in-the-middle attacks by providing additional information, such as channel-binding tokens and service-binding identifiers.

Access Control

IIS 7 allows you to restrict access to the Web server based on IP address, IP address ranges, and domain names. Access can be limited based on other parameters as well, such as computers, groups of computers, or domains. Also, access is granular, meaning it can be restricted to certain Web sites, applications, directories, and individual files. Furthermore, you can filter the types of HTTP requests that will be processed by the Web server.

Data Confidentiality

IIS supports SSL encryption, so you can request and install Internet server digital certificates, install domain server digital certificates, or even create a self-signed server certificate. IIS configuration of SSL includes the ability to require SSL to access the Web server, determine the bit length of cryptographic keys in use, and control whether clients must verify their identity to connect with the Web server.

Controlling Dynamic Content

Windows Web servers have traditionally used Internet Server Application Programming Interface (ISAPI) and Common Gateway Interface (CGI) to provide interactive and dynamic content. ISAPI extensions are applications, and ISAPI filters are programs used to modify or enhance IIS functionality. Both of these ISAPI components are typically implemented as .dll files. CGI is a standard commonly used in the creation of interactive forms; it is implemented as .exe files. IIS 7 allows you to restrict the activity of ISAPI and CGI components.

Shared Configuration

Today, relatively few single Web servers are found in e-commerce settings. Server arrays and large server farms are common. IIS 7 supports shared configuration, which allows administrators to import configuration files and cryptographic keys from a centralized location. Similarly, configuration files and keys can be exported from a single server to the central location as a backup.

Other Security Considerations

IIS security features are important to consider when planning Web server security, but they do not guarantee a hardened system with the fewest possible vulnerabilities. You should also follow these precautions:

- The underlying Windows OS must be hardened and maintained by installing the latest service packs, patches, and hot fixes, and by removing or disabling unnecessary services.
- A domain controller should not also function as an IIS Web server. Domain controllers store Active Directory information and control network access and authentication. This server handles critical services, so making it available on the Internet is not a good security practice. Domain controllers should be kept in the protected internal network and separated from the Internet with firewalls.
- Place the Web server in a secure room. An organization cannot achieve information security without physical security. Despite all the passwords and firewalls used on a network, a Web server can still be compromised if it is kept in an unsecured area. Restrict access to Web servers by using physical security measures, such as surveillance cameras, locks, smartcards, and other access control systems.
- Do not connect the IIS Web server to the Internet before it is fully hardened. When a server is connected to the public Internet, it usually does not take long for an attack to occur.
- Remove NTFS write and execute permissions when possible to minimize the risk of unauthorized users changing files or running programs.
- Grant permissions for modifying and viewing IIS logs to system and local administrators only. This precaution makes it harder for attackers to modify log files to hide their activities. As an added precaution, store logs on another server, not the IIS Web server.
- Allow only the administrator to log on locally to the Web server. Secure services outside the OS, such as SQL Server, to prevent them from being exploited as user accounts.
- If you are serving Web pages to the Internet, place the Web server in a firewall-protected DMZ.

As with all Microsoft products, service packs, patches, and hot fixes for IIS are released periodically. Installing them as soon as possible is important, especially when they address security issues. Subscription to the automatic Microsoft Security Notification Service is recommended for updated news on IIS as well as updates on all Microsoft products.

Configuring Security Settings in Apache Web Server

Apache Web Server, the most widely used Web server application, is installed mainly on UNIX and Linux systems, although a Windows version is available. Apache's vulnerabilities are not publicized as much as those for Windows Web servers, but they do exist. Some misguided Apache administrators believe that Apache is secure out of the box and do not pay much attention to hardening, which can be a grave mistake. As with any Web server, Apache requires hardening to ensure security for Web sites and users. The Center for Internet Security (CIS) recommends the following security settings for Apache:

- Harden the underlying OS as you would any OS by removing unused applications and sample code and updating OS patches and hot fixes.
- Install the latest Apache binary distribution code from the OS vendor. This approach is usually easier than compiling your own binary code for installation because the vendor has already done most of the configuration work for you.
- Disable unnecessary Apache modules and services, disable processing of server-side includes (SSIs), and delete unneeded or default Apache files and sample code. These measures reduce the number of Web processes that are available to attackers.
- Create Web groups so that users can be granted limited administrative rights without having root access.
- Create user and group accounts with limited privileges for running Apache Web Server, and never run Apache as the root account. If the Web service runs with root permissions, any compromise results in attackers having root access to the Web server.
- Subscribe to OS vendor and Apache security advisories to stay informed about security issues.
- Develop customized messages for Web pages that display error information. As you learned previously, attackers can use error messages to gather information about server setup.
- Install the ModSecurity module to have URLs in Web traffic inspected for anomalies. For example, an attacker might send the following URL request to a Web server to delete the accounts database table:

· `http://www.myweb.com/login.asp?username=admin';DROP %20TABLE%20accounts--` The ModSecurity module adds a filter to prevent these types of requests.
- To secure access, use Digest authentication instead of Basic authentication for accepting usernames and passwords. (If you need to review these authentication methods, conduct an Internet search for articles.)
- When setting access control lists (ACLs), determine whether allow or deny rules are evaluated first. An ACL's effect could change if you do not use the correct order of evaluation.

- Use Secure Sockets Layer (SSL) to encrypt the communication from user to Web server. First, download and install the mod_ssl module from www.modssl.org/source/. Then install an SSL certificate purchased from a recognized certification authority (CA), such as VeriSign or Geotrust. The CA includes directions for installing the certificate. For example, VeriSign provides instructions at <https://knowledge.verisign.com/support/ssl-certificates-support/index?page=content&id=AR212>.
- Limit the Web server to accepting and processing only certain HTTP request methods, such as GET, POST, HEAD, and PUT.
- Disable HTTP traces to prevent attackers from investigating HTTP request paths for potential targets. An HTTP trace asks the Web server to echo back an HTTP request's contents and is often used for debugging. Attackers could use this information to access sensitive data, such as authentication data or cookies from an established connection. For more information, see <https://www.kb.cert.org/vuls/id/867593>.
- Enable logging on the Web server so that you can spot potential problems and suspicious activity. If the server is compromised, logs also give you a record for forensics analysis. To prevent attackers from accessing and altering logs, store them on a separate network server, not the Web server.

◀ Preliminary Activity for Week 14

Jump to...




Analysis, Application, and Exploration for Week 14 ▶



Navigation

Home

 Dashboard

Site pages

My courses

Capstone Project 2

Network Defense and Remote Access Configuration

Participants


General

13 [Enter Module Title Here]

14 [Enter Module Title Here]

 Preliminary Activity for Week 14

 **Lesson Proper for Week 14**

 Analysis, Application, and Exploration for Week 14

 Generalization for Week 14

 Evaluation for Week 14

 Assignment for Week 14

15 [Enter Module Title Here]

16 [Enter Module Title Here]
17 [Enter Module Title Here]
OJT/Practicum 2
Seminars and Tours
Courses





Fair Warning

NOTICE: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for ***free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission.***

PROSECUTION: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following:
Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense.
Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

COURSE OF ACTION: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.

Activities

-  Assignments
-  Forums
-  Quizzes
-  Resources