A Checklist for Network Security

By J. Craig Lowery, Ph.D.

Although networking has made the explosive growth of computer applications possible, the security liabilities it introduces are extremely problematic. In fact, a system's network connection is the primary target of most modern security attacks. This article provides a checklist of security safeguards for system administrators to implement.

Networking—connecting one computer to other computers—improves user productivity and quality of life. It boosts efficiency, provides new communication methods, enhances research capabilities, and expands commerce opportunities. But networking also increases a computer's vulnerability to threats such as data theft, tampering, and service disruptions. Still, these inherent security problems are more than an equitable exchange for the capabilities that networking provides. The challenge for system administrators is to minimize security risks without unnecessarily diminishing networking benefits.

The practical checklist of safeguards presented here addresses the networking-specific vulnerabilities described in a previous article.[1] System administrators can use this list as a guide for achieving a reasonable level of network security.

Fortify the firewall

A *firewall* is a single point of controlled communication between trusted and distrusted networks. It can take several forms, ranging from a simple, packet-filtering gateway router to a sophisticated, traffic-analyzing appliance. Firewalls protect against many IP-based attacks such as spoofing (see sidebar "Help stop IP spoofing"), ping flooding, and denial of service (DoS) attacks. Firewalls can also establish *demilitarized zones* (see sidebar "Deploy a DMZ").

Firewall capabilities depend on the device used to create it. Each device provides both benefits and liabilities. For example, high-end firewalls have extensive customization features for implementing security policies but may unduly restrict users if misapplied.

How a firewall provides protection is largely determined by how the trusted network connects to the distrusted network, which is typically the Internet. Figure 1 shows two connection possibilities: a routed connection and a proxied connection.

Firewall with routed connection

A gateway router simply extends the Internet address space and routing into the trusted network. Routers employ packet filtering, deciding whether a particular packet should be allowed to traverse the boundary based upon source and destination IP address, protocol, or port numbers. For example, it is common practice to exclude all Internet Control Message Protocol (ICMP) packets, or datagrams, because they can be vehicles for many types of DoS attacks.

Most Internet routers provide this simple, network-layer filtering, but sophisticated firewall software can extend packet filtering so that routing decisions are based on more complex criteria. Firewall software can examine the actual data within a traffic stream and interpret the data within the context of the communicating applications. This technique, called *stateful inspection*, identifies packets that are part of the same ongoing communication and maintains state information about the packets. The firewall then uses this information to make filtering decisions for future packets.

[1] Lowery, J. Craig. "Computer System Security: A Primer." Dell *Power Solutions*, March 2002.

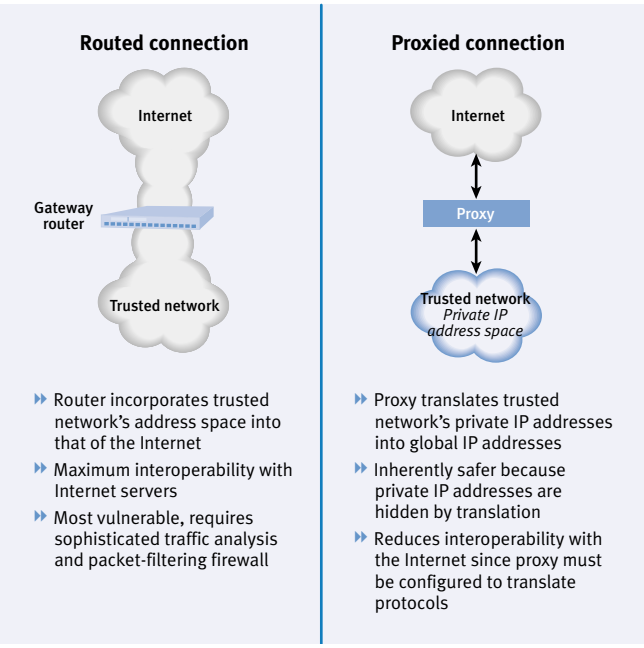

Figure 1. Implementations for connecting the Internet to trusted networks

Firewall with proxy connection

With proxies, packets never cross the boundary between trusted and distrusted networks. Instead, the proxy acts as a go-between, presenting a single "face" to the distrusted network and hiding the trusted one. Systems such as SOCKS or Network Address Translation (NAT) maintain logical associations across the separate networks for each member of a communicating pair. Proxies facilitate only the communication initiated by internal clients, refusing unsolicited connections from the Internet.

Harden the software

Network-related software, particularly operating systems, should be scrutinized for security reasons. Because such software provides shared programming interfaces and libraries, vulnerabilities in this code affect all processes accessing a resource such as the network. Network stacks that can detect DoS attacks, such as ping or SYN floods, are preferred, and most operating systems now provide this feature.

Although service programs such as Web servers are technically applications, they are commonly installed with the operating system and enabled with low-security thresholds by default. Typically, each service creates a listening port to accept incoming requests; an unused open port is an invitation to security attacks. For these reasons, administrators should tighten services configurations and enable only necessary services. Logs from services such as FTP or the Web server are also essential to help identify and track the perpetrators when a security breach occurs.

A common exploitation of network-accessible software is the buffer overrun. When a software routine does not adequately check that buffers are large enough to receive incoming data, systems

HELP STOP IP SPOOFING

IP spoofing is the purposeful forging of IP addresses in an Internet datagram (packet) so the datagram appears to originate from or be destined for locations other than the actual source or destination. Hackers often use spoofing to hide their location or to make an innocent third party appear as the source of the attack.

Network administrators can reduce spoofing on the Internet by ensuring that gateway routers employ effective packet-filtering rules for both outgoing and incoming traffic (see Figure A). To filter outgoing traffic, these rules should allow only the outgoing traffic that bears an internal network source address and is destined for an address outside the internal network. This type of filtering prevents a site from becoming an unwitting accomplice to a spoofing attack. Internet service providers (ISPs) should implement effective outgoing packet filtering for each client connection.

To filter incoming traffic, packet-filtering rules should refuse packets that have the source addresses of the internal network but actually originate from outside it. Hackers create such packets to exploit IP-based authentication in which a computer trusts a communication because it appears to come from a trusted source.

Finally, administrators should configure gateway routers to reject all packets bearing any address in the private, multicast, loop-back, or Dynamic Host Configuration Protocol (DHCP) spaces, as set forth in RFC-1918.

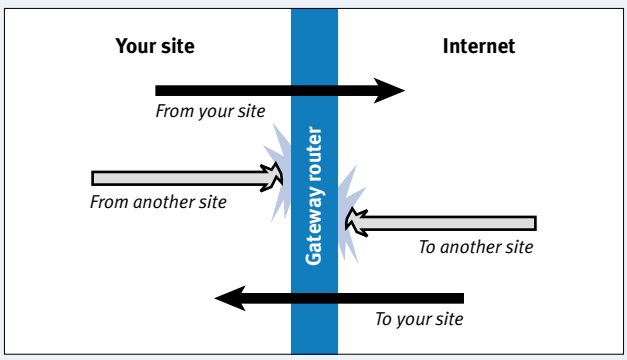

Figure A. Filtering spoofed packets at a gateway router

become vulnerable to buffer overruns. In this attack, hackers construct inputs that overwrite a process stack with malicious code.

Software developers struggle to eradicate this common coding error, and periodic patches are often required. But the immediate application of vendor patches is not prudent because a patch is not always completely effective. A patch can even introduce new vulnerabilities. If possible, administrators should test a patch in a sandbox system or apply it only after early adopters have proven its effectiveness. Administrators should immediately apply proven patches to systems that are reinstalled from source media.

Administrators should also closely inspect Web servers using the Common Gateway Interface (CGI) and remove all example scripts installed with the Web server. When writing CGI scripts, programmers should use languages that have safe and tainted modes, such as Perl and PHP (Hypertext Pre-Processor), to reduce the scope of sensitive operating system functions that a script can invoke. Administrators should also be aware of shell escapes. These programming blunders allow input from a Web-based form to become part of a command line passed to a command interpreter. If a Web site does not require CGI services, the administrator should disable CGI.

Desktop applications can also provide a conduit for network attack. E-mail worms and viruses can exploit weak software design and user ignorance. To combat these threats, system administrators should mandate the use of up-to-date virus scanning software. For centralized e-mail systems such as Microsoft® Exchange, server-side virus and worm scanning can help squelch attacks.

Defend the DNS

The Domain Name System (DNS) is vulnerable to attack in that when anyone requests a name resolution, top-level name servers redirect the query to an authoritative server. If hackers change the authoritative server records hosted by a registrar, they can effectively "hijack" that domain.

System administrators can protect DNS server information by ensuring that only designated persons can alter the information. Some registrars still support outdated methods, such as e-mail, for updating DNS records. These outdated methods need additional security, such as encrypted passwords, Pretty Good Privacy (PGP) encryption, or the secure Web-based tools that registrars also provide for managing domain records.

Sites running their own DNS servers must configure secure zone transfers of authoritative information from the primary to the secondary servers. Hackers can use the contents of a zone file to gather sensitive information about a target network, including the network's IP addresses, types of operating systems, and mail server identities. This information enables hackers to launch specific attacks.

Newer DNS software can place access controls on DNS records so that a name server responds only to queries sent by legitimate inquirers. Locally originating queries have full access to zone information; external queries have access only to a few publicly viewable records. Administrators also can use a split DNS implementation: internal computers consult DNS servers that host full zone information while external computers consult other DNS servers that host only the publicly viewable records.

> The challenge for system administrators is to minimize security risks without unnecessarily diminishing networking benefits.

Shut out the sniffers

To obtain usernames and passwords transmitted during authentication sequences, hackers often employ packet sniffing—observing all traffic on a network segment by using a network interface in "promiscuous" mode. Telnet and FTP both require an authentication sequence, which historically has been unsecure; therefore, users should

DEPLOY A DMZ

In networking, the term *demilitarized zone* (DMZ) refers to a third network, separate from both the trusted and distrusted networks. Computers that are accessed regularly by incoming Internet connections—typically, Web and e-mail servers—reside in the DMZ (see Figure B). The firewall permits communication between the DMZ and the other two networks under less stringent rules than those applied to communication between the trusted and distrusted networks.

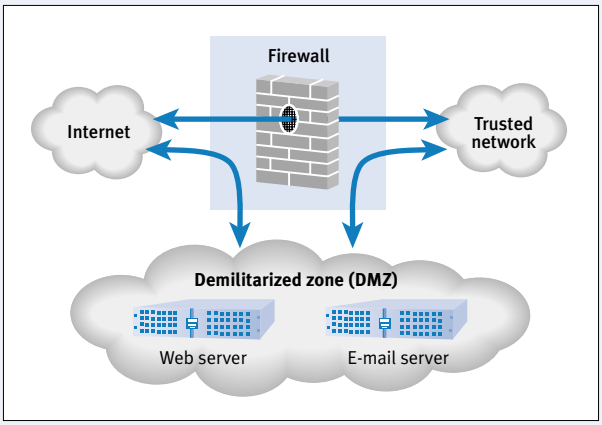

Figure B. Network design for a demilitarized zone

AVOID BECOMING A SPAM RELAY

Internet security organizations identify known spam-generating Web sites and place them on "blacklists," which many sites use to filter incoming e-mail. To circumvent these filters, spammers find an unsuspecting relay[2]-enabled e-mail system not on the blacklists.

Spammers use this mail relay, called a *reflector*, to bounce messages into a target site (see Figure C). Since the mail appears to come from a non-blacklisted site, it passes through the target system's filter. If not corrected, this misuse can cause the intermediate e-mail system to be blacklisted as well. For this reason, administrators should disable mail relay unless it is required and configured specifically for legitimate sites.

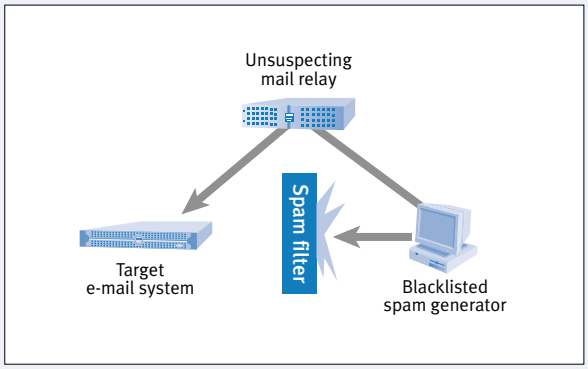

Figure C. Circumventing spam filters with mail relay

take advantage of newer Telnet and FTP authentication mechanisms, such as Kerberos, or deploy Secure Shell (SSH) as a replacement for Telnet.

Hackers can also launch replay attacks by recording network transmissions and then retransmitting them at a later time. Applications that perform sensitive operations in response to network input should do so only in the context of a session. Session communication incorporates a unique, time-sensitive identifier bound to a single, logical connection between client and server. Information exchanged during a session will be rejected if replayed during a future session. As applications migrate to the Web, developers must consider the replay attack threat, and organizations deploying Web applications must anticipate these attacks, even simple ones such as the reposting of a stale form from browser cache.

For tighter security, many network operating systems, such as Novell® NetWare®, provide optional encryption of all transmissions between clients and servers. A more expensive but transparent alternative is link-level encryption, in which routers encrypt and decrypt datagrams at each hop.

Patrol the passwords

Poor or missing passwords are a primary entry point for would-be hackers, and identifying target systems is becoming easier with each improvement in computing speed. As newer, faster processors appear, the time it takes a dictionary-based guessing program to crack weak passwords diminishes.

Administrators must enforce good password selection. Most operating systems provide authentication mechanisms that can check a candidate password's "crack-ability." System-assigned passwords are less likely to be cracked, but are unpopular with users. Another option is password aging. Although this tool requires additional support and may be considered a nuisance, it forces users to keep passwords fresh and prohibits password reuse.

Tighten network security, maintain system stability

The list of safeguards set forth in this article is not exhaustive, but it does include most of the common pitfalls encountered when setting up and managing a trusted network with Internet connectivity. Maintaining a safe computing environment requires vigilance and dedication to frequent, critical evaluations of threats and defenses.

J. Craig Lowery, Ph.D. *(craig_lowery@dell.com) is a software architect and strategist in the Dell Enterprise Software Development organization. Craig received an M.S. and a Ph.D. in Computer Science from Vanderbilt University, and a B.S. in Computing Science and Mathematics from Mississippi College. His primary areas of interest include networking, performance evaluation, and operating systems.*

FOR MORE INFORMATION

Visit the System Administration, Networking, and Security Institute online at http://www.sans.org

[2] Mail relay was originally intended to provide mail connectivity between endpoints that could not communicate with each other. It is usually turned on by default in older software.