Chapter 9

Network Infrastructure

In This Chapter

- Selecting tools
- Scanning network hosts
- Assessing security with a network analyzer
- Preventing denial-of-service and infrastructure vulnerabilities

Your computer systems and applications require one of the most fundamental communications systems in your organization — your network. Your network consists of such devices as routers, firewalls, and even generic hosts (including servers and workstations) that you must assess as part of the ethical hacking process.

There are thousands of possible network vulnerabilities, equally as many tools, and even more testing techniques. You probably don't have the time or resources available to test your network infrastructure systems for *all* possible vulnerabilities, using every tool and technique imaginable. Instead, you need to focus on tests that will produce a good overall assessment of your network — and the tests I describe in this chapter will do exactly that.

You can eliminate many well-known, network-related vulnerabilities by simply patching your network hosts with the latest vendor software and firmware patches. Since most network infrastructure hosts are not publicly accessible, odds are that your network hosts *will not* be attacked from the outside and even if they are, the results are not likely to be detrimental. You can eliminate many other vulnerabilities by following some solid security practices on your network, as described in this chapter as well as in the book *Network Security For Dummies*. The tests, tools, and techniques outlined in this chapter offer the most bang for your ethical-hacking buck.

TIP

The better you understand network protocols, the easier network vulnerability testing will be for you because network protocols are the foundation for most information security concepts. If you're a little fuzzy on how networks work, I highly encourage you to read *TCP/IP For Dummies,* 5th Edition, by Candace Leiden and Marshall Wilensky (Wiley Publishing, Inc.), as well as the Request for Comments (RFCs) list at the Official Internet Protocol Standards page, `www.rfc-editor.org/rfcxx00.html`.

A case study in hacking network infrastructures with Laura Chappell

Laura Chappell — one of the world's foremost authorities on network protocols and analysis — shared with me an interesting experience she had when assessing a customer's network. Here's her account of what happened:

The Situation

A customer called Ms. Chappell with a routine "the network is slow" problem. Upon her arrival onsite, the customer also mentioned sporadic outages and poor performance when connecting to the Internet. First, she examined individual flows between various clients and servers. Localized communications appeared normal, but any communication that flowed through the firewall to the Internet or other branch offices was severely delayed. It was time to sniff the traffic going through the firewall to see whether she could isolate the cause of the delay.

The Outcome

A quick review of the traffic crossing the firewall indicated that the outside links were saturated, so it was time to review and classify the traffic. Using the Sniffer Network Analyzer, Ms. Chappell plugged in to examine the protocol distribution. She saw that almost 45 percent of the traffic was listed as "others" and was unrecognizable. She captured some data and found several references to pornographic images. Further examination of the packets led her to two specific port numbers that appeared consistently in the trace files — ports 1214 (Kazaa) and 6346 (Gnutella), two peer-to-peer (P2P) file-sharing applications. She did a complete port scan of the network to see what was running and found over 30 systems running either Kazaa or Gnutella. Their file transfer processes were eating up the bandwidth and dragging down all communications. It would have been simple to shut down these systems and remove the applications, but she wanted to investigate them further without the users' knowledge.

Ms. Chappell decided to use her own Kazaa and Gnutella clients to look through the shared folders of the systems. By becoming a peer member with the other hosts on the network, she could perform searches through other shared folders, which indicated some of the users had shared their network directories! Through these shared folders, she was able to obtain the corporate personnel roster, including home phone numbers and addresses, accounting records, and several confidential memos that provided timelines for projects under way at the company!

Many users said they shared these folders to regain access to the P2P network because they had previously been labeled *freeloaders* because their shares contained only a few files. They were under the delusion that because no one outside the company knew the filenames contained in the network directories, a search wouldn't come up with matching values, and so no one would download those files. Although this onsite visit started with a standard performance and communication review, it ended with the detection of some huge security breaches in the company. Anyone could have used these P2P tools to get onto the network and grab the files in the shared folders — with no authorization or authentication required!

Laura Chappell is Senior Protocol Analyst at the Protocol Analysis Institute, LLC (`www.packet-level.com`). A best-selling author and lecturer, Ms. Chappell has trained thousands of network administrators, security technicians, and law enforcement personnel on packet-level security, troubleshooting, and optimization techniques. I *highly* recommend that you check out her Web site for some excellent technical content that can help you become a better ethical hacker.

Network Infrastructure Vulnerabilities

Network infrastructure vulnerabilities are the foundation for all technical security issues in your information systems. These lower-level vulnerabilities affect everything running on your network. That's why you need to test for them and eliminate them whenever possible.

Your focus for ethical hacking tests on your network infrastructure should be to find weaknesses that others can see in your network so you can quantify your network's level of exposure.

Many issues are related to the security of your network infrastructure. Some issues are more technical and require you to use various tools to assess them properly. You can assess others with a good pair of eyes and some logical thinking. Some issues are easy to see from outside the network, and others are easier to detect from inside your network.

When you assess your company's network infrastructure security, you need to look at such areas as

- Where devices such as a firewall or IPS are placed on the network and how they are configured
- What hackers see when they perform port scans, and how they can exploit vulnerabilities in your network hosts
- Network design, such as Internet connections, remote access capabilities, layered defenses, and placement of hosts on the network
- Interaction of installed security devices such as firewalls, IDSs, and antivirus, and so on
- What protocols are in use
- Commonly attacked ports that are unprotected
- Network host configuration
- Network monitoring and maintenance

If a hacker exploits a vulnerability in one of the items above or anywhere in your network's security, bad things can happen:

- A hacker can use a DoS attack, which can take down your Internet connection — or even your entire network.
- A malicious employee using a network analyzer can steal confidential information in e-mails and files being transferred on the network.
- A hacker can set up backdoors into your network.
- A hacker can attack specific hosts by exploiting local vulnerabilities across the network.

TIP

Before moving forward with assessing your network infrastructure security, remember to do the following:

- Test your systems from the outside in, the inside out, and the inside in (that is, between internal network segments and DMZs).
- Obtain permission from partner networks that are connected to your network to check for vulnerabilities on their ends that can affect *your* network's security, such as open ports and lack of a firewall or a misconfigured router.

Choosing Tools

Your tests require the right tools — you need scanners and analyzers, as well as vulnerability assessment tools. Great commercial, shareware, and freeware tools are available. I describe a few of my favorite tools in the following sections. Just keep in mind that you need more than one tool, and that no tool does everything you need.

TIP

If you're looking for easy-to-use security tools with all-in-one packaging, *you get what you pay for* — most of the time — especially for the Windows platform. Tons of security professionals swear by many free security tools, especially those that run on Linux and other UNIX-based operating systems. Many of these tools offer a lot of value — if you have the time, patience, and willingness to learn their ins and outs.

Scanners and analyzers

These scanners provide practically all the port-scanning and network-testing tools you'll need:

- **Sam Spade for Windows** (`http://samspade.org/ssw`) for network queries from DNS lookups to traceroutes
- **SuperScan** (`www.foundstone.com/resources/proddesc/superscan.htm`) for ping sweeps and port scanning
- **Essential NetTools** (`www.tamos.com/products/nettools`) for a wide variety of network scanning functionality
- **NetScanTools Pro** (`www.netscantools.com`) for dozens of network security assessment functions, including ping sweeps, port scanning, and SMTP relay testing
- **Getif** (`www.wtcs.org/snmp4tpc/getif.htm`) for SNMP enumeration
- **Nmap** (`www.insecure.org/nmap`) or **NMapWin** (`http://sourceforge.net/projects/nmapwin`) which is a happy-clicky-GUI front end to Nmap for host-port probing and operating-system fingerprinting

- **Netcat** (`www.vulnwatch.org/netcat/nc111nt.zip`) for security checks such as port scanning and firewall testing
- **LanHound** (`www.sunbelt-software.com/LanHound.cfm`) for network analysis
- **WildPackets EtherPeek** (`www.wildpackets.com/products/etherpeek/overview`) for network analysis

Vulnerability assessment

These vulnerability assessment tools allow you to test your network hosts for various known vulnerabilities as well as potential configuration issues that could lead to security exploits:

- **GFI LANguard Network Security Scanner** (`www.gfi.com/lannetscan`) for port scanning and other vulnerability testing
- **Sunbelt Network Security Inspector** (`www.sunbelt-software.com/SunbeltNetworkSecurityInspector.cfm`) for vulnerability testing
- **Nessus** (`www.nessus.org`) as a free all-in-one tool for tests like ping sweeps, port scanning, and vulnerability testing
- **Qualys QualysGuard** (`www.qualys.com`) as a great all-in-one tool for in-depth vulnerability testing

Scanning, Poking, and Prodding

Performing the ethical hacks described in the following sections on your network infrastructure involves following basic hacking steps:

1. Gather information and map your network.
2. Scan your systems to see which are available.
3. Determine what's running on the systems discovered.
4. Attempt to penetrate the systems discovered, if you choose to.

TIP

Every network card driver and implementation of TCP/IP in most operating systems, including Windows and Linux, and even in your firewalls and routers, has quirks that result in different behaviors when scanning, poking, and prodding your systems. This can result in different responses from your varying systems. Refer to your administrator guides or vendor Web sites for details on any known issues and possible patches that are available to fix them. If you have all your systems patched, this shouldn't be an issue.

Port scanners

A port scanner shows you what's what on your network. It's a software tool that basically scans the network to see who's there. Port scanners provide basic views of how the network is laid out. They can help identify unauthorized hosts or applications and network host configuration errors that can cause serious security vulnerabilities.

The big-picture view from port scanners often uncovers security issues that may otherwise go unnoticed. Port scanners are easy to use and can test systems regardless of what operating systems and applications they're running. The tests can usually be performed fairly quickly without having to touch individual network hosts, which would be a real pain otherwise.

The real trick to assessing your overall network security is interpreting the results you get back from a port scan. You can get false positives on open ports, and you may have to dig deeper. For example, UDP scans — like the protocol itself — are less reliable than TCP scans and often produce false positives because many applications don't know how to respond to random incoming UDP scans.

TIP

A feature-rich scanner often can identify ports and see what's running in one step.

WARNING!

Port scan tests can take time. The length of time depends on the number of hosts you have, the number of ports you scan, the tools you use, and the speed of your network links.

Scan more than just the important hosts. Leave no stone unturned. These *other* systems often bite you if you ignore them. Also, perform the same tests with different utilities to see whether you get different results. Not all tools find the same open ports and vulnerabilities. This is unfortunate, but it's a reality of ethical hacking tests.

If your results don't match after you run the tests using different tools, you may want to explore the issue further. If something doesn't look right — such as a strange set of open ports — it probably isn't. Test it again; if you're in doubt, use another tool for a different perspective.

TIP

As an ethical hacker, you should scan all 65,535 TCP ports on each network host that's found by your scanner. If you find questionable ports, look for documentation that the application is known and authorized. It's not a bad idea to scan all 65,535 UDP ports as well.

For speed and simplicity, you can scan the commonly hacked ports, listed in Table 9-1.

Table 9-1 Commonly Hacked Ports

Port Number	*Service*	*Protocol(s)*
7	Echo	TCP, UDP
19	Chargen	TCP, UDP
20	FTP data (File Transfer Protocol)	TCP
21	FTP control	TCP
22	SSH	TCP
23	Telnet	TCP
25	SMTP (Simple Mail Transfer Protocol)	TCP
37	Daytime	TCP, UDP
53	DNS (Domain Name System)	UDP
69	TFTP (Trivial File Transfer Protocol)	UDP
79	Finger	TCP, UDP
80	HTTP (Hypertext Transfer Protocol)	TCP
110	POP3 (Post Office Protocol version 3)	TCP
111	SUN RPC (remote procedure calls)	TCP, UDP
135	RPC/DCE (end point mapper) for Microsoft networks	TCP, UDP
137, 138, 139, 445	NetBIOS over TCP/IP	TCP, UDP
161	SNMP (Simple Network Management Protocol)	TCP, UDP
220	IMAP (Internet Message Access Protocol)	TCP
443	HTTPS (HTTP over SSL)	TCP
512, 513, 514	Berkeley *r* commands (such as rsh, rexec, and rlogin)	TCP
1214	Kazaa and Morpheus	TCP, UDP
1433	Microsoft SQL Server (ms-sql-s)	TCP, UDP
1434	Microsoft SQL Monitor (ms-sql-m)	TCP, UDP
3389	Windows Terminal Server	TCP
5631, 5632	pcAnywhere	TCP

(continued)

Table 9-1 *(continued)*

Port Number	*Service*	*Protocol(s)*
6346, 6347	Gnutella	TCP, UDP
12345, 12346, 12631, 12632, 20034, 20035	NetBus	TCP
27444, 27665, 31335, 34555	Trinoo	TCP, UDP
31337	Back Orifice	UDP

Ping sweeping

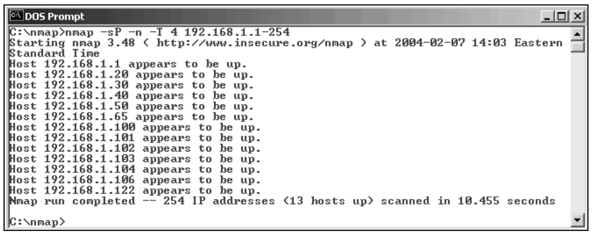
A ping sweep of all your network subnets and hosts is a good way to find out which hosts are alive and kicking on the network. A *ping sweep* is when you ping a range of addresses using Internet Control Message Protocol (ICMP) packets. Figure 9-1 shows the command and the results of using Nmap to perform a ping sweep of a class C subnet range.

Dozens of Nmap command-line options exist, which can be overwhelming when you just want to do a basic scan. You can just enter `nmap` on the command line to see all the options available.

The following command-line options can be used for an Nmap ping sweep:

- `-sP` tells Nmap to perform a ping scan.
- `-n` tells Nmap not to perform name resolution.

  You can omit this if you want to resolve hostnames to see which systems are responding. Name resolution may take slightly longer, though.
- `-T 4` option tells Nmap to perform an aggressive (faster) scan.
- `192.168.1.1-254` tells Nmap to scan the entire 192.168.1.x subnet.



```
C:\nmap>nmap -sP -n -T 4 192.168.1.1-254
Starting nmap 3.48 ( http://www.insecure.org/nmap ) at 2004-02-07 14:03 Eastern
Standard Time
Host 192.168.1.1 appears to be up.
Host 192.168.1.20 appears to be up.
Host 192.168.1.30 appears to be up.
Host 192.168.1.40 appears to be up.
Host 192.168.1.50 appears to be up.
Host 192.168.1.65 appears to be up.
Host 192.168.1.100 appears to be up.
Host 192.168.1.101 appears to be up.
Host 192.168.1.102 appears to be up.
Host 192.168.1.103 appears to be up.
Host 192.168.1.104 appears to be up.
Host 192.168.1.106 appears to be up.
Host 192.168.1.122 appears to be up.
Nmap run completed -- 254 IP addresses (13 hosts up) scanned in 10.455 seconds

C:\nmap>
```

Figure 9-1: Performing a ping sweep of an entire class C network with Nmap.

Using port scanning tools

Most port scanners operate in three steps:

1. The port scanner sends TCP SYN requests to the host or range of hosts you set it to scan.

   Some port scanners, such as SuperScan, perform ping sweeps to determine which hosts are available before starting the TCP port scans.

   Most port scanners by default scan only TCP ports. Don't forget about UDP ports. You can scan UDP ports with a UDP port scanner such as Nmap.

2. The port scanner waits for replies from the available hosts.
3. The port scanner probes these available hosts for up to 65,535 possible TCP and UDP ports — based on which ports you tell it to scan — to see which ones have available services on them.

The port scans provide the following information about the live hosts on your network:

- Hosts that are active and reachable through the network
- Network addresses of the hosts found
- Services or applications that the hosts *may be* running

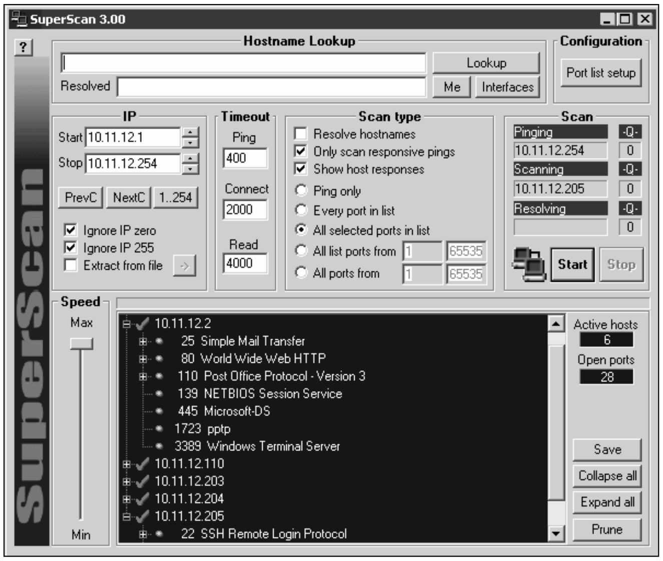
After performing a generic sweep of the network, you can dig deeper into specific hosts you've found.

SuperScan

My favorite tool for performing generic TCP port scans is SuperScan version 3.0. Figure 9-2 shows the results of my scan and a few interesting ports open on several hosts, including Windows Terminal Server and SSH.

In Figure 9-2, I selected the Only Scan Responsive Pings and All Selected Ports in List options. However, you may want to select some other options:

- If you don't want to ping each host first, deselect the Only Scan Responsive Pings option. ICMP can be blocked, which can cause the scanner to not find certain hosts, so this option can make the test run more efficiently.
- If you want to scan a certain range of well-known ports or ports specific to your systems, you can configure SuperScan to do so. I recommend these settings:
  - If you want to perform a scan on well-known ports, at least select the All Selected Ports in List option.
  - If this is your initial scan, scan all ports from 1 to 65,535.



Resolved

IP
Start 10.11.12.1
Stop 10.11.12.254
☑ Ignore IP zero
☑ Ignore IP 255
☐ Extract from file

Timeout
Ping 400
Connect 2000
Read 4000

Scan type
☐ Resolve hostnames
☑ Only scan responsive pings
☑ Show host responses
○ Ping only
○ Every port in list
● All selected ports in list
○ All list ports from 1 65535
○ All ports from 1 65535

Scan
Pinging -Q-
10.11.12.254 0
Scanning -Q-
10.11.12.205 0
Resolving -Q-
0

Speed
Max
Min

```
⊟ ✓ 10.11.12.2
   ⊞ ●    25  Simple Mail Transfer
   ⊞ ●    80  World Wide Web HTTP
   ⊞ ●   110  Post Office Protocol - Version 3
     ●   139  NETBIOS Session Service
     ●   445  Microsoft-DS
     ●  1723  pptp
     ●  3389  Windows Terminal Server
⊞ ✓ 10.11.12.110
⊞ ✓ 10.11.12.203
⊞ ✓ 10.11.12.204
⊟ ✓ 10.11.12.205
   ⊞ ●    22  SSH Remote Login Protocol
```

Active hosts 6
Open ports 28

Figure 9-2: A TCP port scan using SuperScan version 3.0.

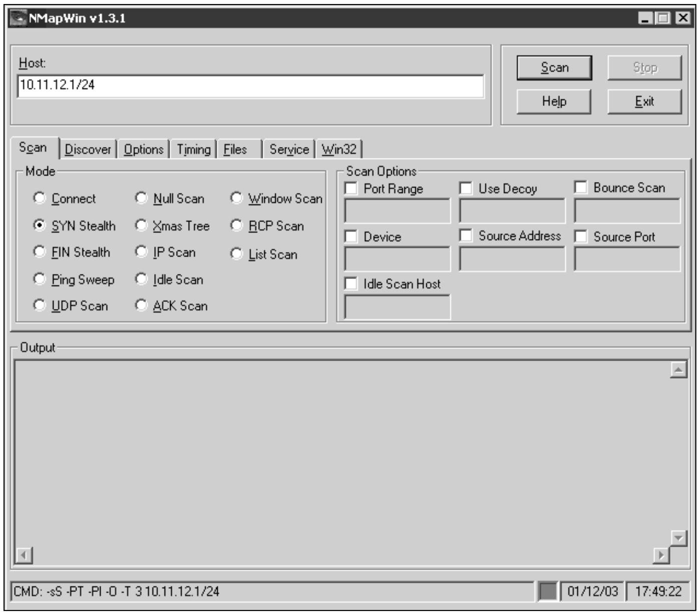
Nmap

After you have a general idea of what hosts are available and what ports are open, you can perform fancier scans to verify that the ports are actually open and not being reported as a false positive. If you wish to do this, Nmap is the perfect tool to use. Nmap allows you to run the following additional scans:

- **Connect:** This basic TCP scan looks for any open TCP ports on the host. You can use this scan to see what's running and determine whether IDSes, firewalls, or other logging devices log the connections.
- **UDP scan:** This basic UDP scan looks for any open UDP ports on the host. You can use this scan to see what's running and determine whether IDSes, firewalls, or other logging devices log the connections.
- **SYN Stealth:** This scan creates a half-open TCP connection with the host possibly evading IDS systems and logging. This is a good scan for testing IDSes, firewalls, and other logging devices.
- **FIN Stealth, Xmas Tree, and Null:** These scans let you mix things up a bit by sending strangely formed packets to your network hosts so you can see how they respond. These scans basically change around the flags in the TCP headers of each packet, which allows you to test how each host handles them to point out weak TCP/IP implementations and patches that may need to be applied.

Be careful when performing these scans. You can create your own DoS attack and potentially crash applications or entire systems. Unfortunately, if you have a host with a weak TCP/IP stack (the software that controls TCP/IP communications on your hosts), there is no good way to prevent your scan from becoming a DoS attack. The best way to reduce the chance of this occurring is to use the slow Nmap timing options — Paranoid, Sneaky, or Polite — when running your scans.

Figure 9-3 shows the NMapWin Scan tab, where you can select all these options. If you're a command-line fan, you see the command-line parameters displayed in the lower-left corner of the NMapWin screen. This helps when you know what you want to do and the command-line help isn't enough.


NMapWin v1.3.1
Scan
Stop
Help
Exit
Scan | Discover | Options | Timing | Files | Service | Win32
Output

Host: 10.11.12.1/24

Mode

○ Connect ○ Null Scan ○ Window Scan
● SYN Stealth ○ Xmas Tree ○ RCP Scan
○ FIN Stealth ○ IP Scan ○ List Scan
○ Ping Sweep ○ Idle Scan
○ UDP Scan ○ ACK Scan

Scan Options

☐ Port Range ☐ Use Decoy ☐ Bounce Scan
☐ Device ☐ Source Address ☐ Source Port
☐ Idle Scan Host

CMD: -sS -PT -PI -O -T 3 10.11.12.1/24 01/12/03 17:49:22

Figure 9-3: In-depth port-scanning options in NMapWin.

If you connect to a single port carefully enough (as opposed to several all at once) without making too much noise, you may be able to evade your IDS/IPS system. This is a good test of your IDS and firewall systems, so assess your logs to see what they saw during this process.

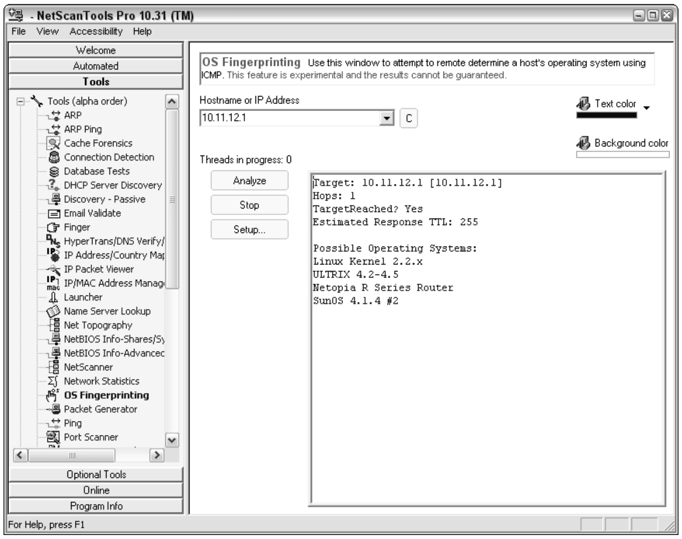
Gathering network information

NetScanTools Pro is a great tool for gathering general network information, such as the number of unique IP addresses, NetBIOS names, and MAC addresses.

The following report is an example of the NetScanner (network scanner) output of NetScanTools Pro 2000:

```
Statistics for NetScanner
Scan completion time = Sat, 7 Feb 2004 14:11:08
Start IP address: 192.168.1.1
End IP address: 192.168.1.254
Number of target IP addresses: 254
Number of IP addresses responding to pings: 13
Number of IP addresses sent pings: 254
Number of intermediate routers responding to pings: 0
Number of successful NetBIOS queries: 13
Number of IP addresses sent NetBIOS queries: 254
Number of MAC addresses obtained by NetBIOS queries: 13
Number of successful Subnet Mask queries: 0
Number of IP addresses sent Subnet Mask queries: 254
Number of successful Whois queries: 254
```

NetScanTools Pro version 10 has a neat feature (although it's experimental) that allows you to fingerprint the operating systems of various hosts. Figure 9-4 shows the OS fingerprint results while scanning a Linksys router/firewall.



OS Fingerprinting Use this window to attempt to remote determine a host's operating system using ICMP. This feature is experimental and the results cannot be guaranteed.

Hostname or IP Address: 10.11.12.1

Threads in progress: 0

```
Target: 10.11.12.1 [10.11.12.1]
Hops: 1
TargetReached? Yes
Estimated Response TTL: 255

Possible Operating Systems:
Linux Kernel 2.2.x
ULTRIX 4.2-4.5
Netopia R Series Router
SunOS 4.1.4 #2
```

Figure 9-4: NetScan Tools Pro OS fingerprinting feature.

Countermeasures against port scanning

You can implement various countermeasures to typical port scanning.

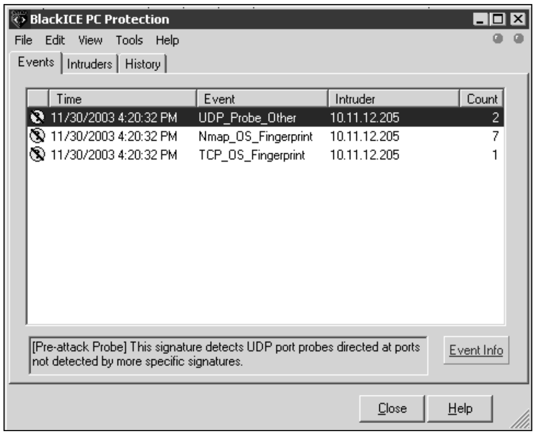
Traffic restriction

Enable only the traffic you need to access internal hosts — preferably as far as possible from the hosts you're trying to protect. You apply these rules in two places:

- External router for inbound traffic
- Firewall for outbound traffic

  Configure firewalls to look for potentially malicious behavior over time (such as the number of packets received in a certain period of time), and have rules in place to cut off attacks if a certain threshold is reached, such as 100 port scans in one minute.

Most firewalls, IDSes, and IPSes detect port scanning and cut it off in real time. Figure 9-5 shows an example: A basic Nmap OS fingerprint scan was detected and cut off (hence the black slash) in real time by ISS's BlackICE personal firewall and IPS product.



Time	Event	Intruder	Count
11/30/2003 4:20:32 PM	UDP_Probe_Other	10.11.12.205	2
11/30/2003 4:20:32 PM	Nmap_OS_Fingerprint	10.11.12.205	7
11/30/2003 4:20:32 PM	TCP_OS_Fingerprint	10.11.12.205	1

[Pre-attack Probe] This signature detects UDP port probes directed at ports not detected by more specific signatures.

Figure 9-5: BlackICE log showing how an Nmap scan was cut off.

Traffic denial

Deny ICMP traffic to specific hosts you're trying to protect. Most hosts don't need to have ICMP enabled — especially inbound ICMP requests — unless it's needed for a network management system that monitors hosts using this protocol.

You *can* break applications on your network, so make sure that you analyze what's going on and understand how applications and protocols are working before you disable such network traffic as ICMP.

SNMP scanning

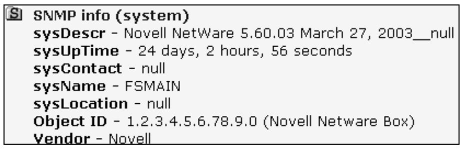
Simple Network Management Protocol (SNMP) is built into virtually every network device. Network management programs (such as HP OpenView and LANDesk Software LANDesk) use SNMP for remote network host management. Unfortunately, SNMP also presents security vulnerabilities.

Vulnerabilities

The problem is that most network hosts run SNMP enabled with the default read/write community strings of public/private. The majority of network devices I come across have SNMP enabled and don't even need it!

If SNMP is compromised, a hacker can gather such network information as ARP tables, usernames, and TCP connections to further attack your systems. If SNMP shows up in port scans, you can bet that a hacker will try to compromise the system. Figure 9-6 shows how GFI LANguard determined the NetWare version running (Version 6, Service Pack 3) by simply querying a host running unprotected SNMP. Here are some other utilities for SNMP enumeration:

Figure 9-6: Information gathered by querying a vulnerable SNMP host.



```
SNMP info (system)
  sysDescr - Novell NetWare 5.60.03 March 27, 2003__null
  sysUpTime - 24 days, 2 hours, 56 seconds
  sysContact - null
  sysName - FSMAIN
  sysLocation - null
  Object ID - 1.2.3.4.5.6.78.9.0 (Novell Netware Box)
  Vendor - Novell
```

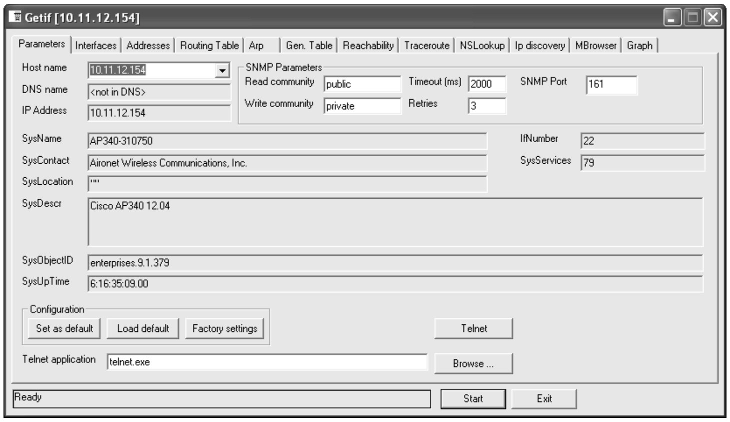
- The commercial tool SolarWinds (www.solarwinds.net) as well as their product SNMP Sweep
- Free Windows GUI-based Getif (www.wtcs.org/snmp4tpc/getif.htm)
- Text-based SNMPUTIL for Windows (www.wtcs.org/snmp4tpc/FILES/Tools/SNMPUTIL/SNMPUTIL.zip)

You can use Getif to enumerate systems with SNMP enabled, as shown in Figure 9-7.

In this test, I was able to glean a lot of information from a wireless access point, including model number, firmware revision, and system uptime. All of this could be used against the host if an attacker wanted to exploit a known vulnerability in this particular system. By digging in further, I was able to discover several management interface usernames on this access point, as shown in Figure 9-8.

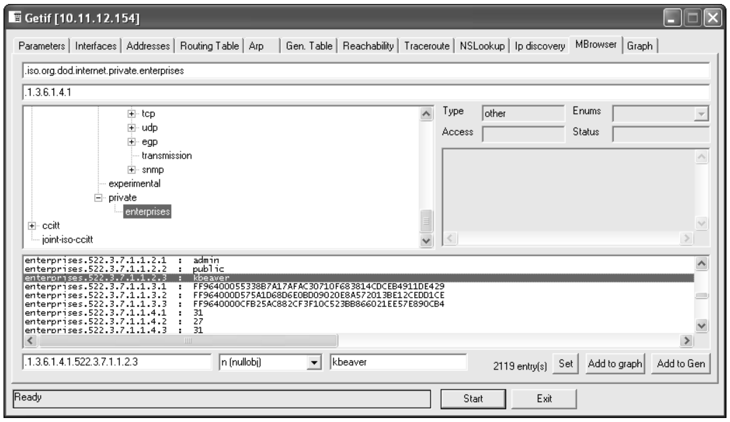
Information such as this is certainly not what you want to be showing off to the world.

For a list of vendors and products affected by the well-known SNMP vulnerabilities, refer to www.cert.org/advisories/CA-2002-03.html.


Getif [10.11.12.154]
Parameters | Interfaces | Addresses | Routing Table | Arp | Gen. Table | Reachability | Traceroute | NSLookup | Ip discovery | MBrowser | Graph
SNMP Parameters
Configuration
Set as default
Load default
Factory settings
Telnet
Browse ...
Start
Exit

Host name: 10.11.12.154
DNS name: <not in DNS>
IP Address: 10.11.12.154
Read community: public
Write community: private
Timeout (ms): 2000
Retries: 3
SNMP Port: 161
SysName: AP340-310750
SysContact: Aironet Wireless Communications, Inc.
SysLocation: ""
SysDescr: Cisco AP340 12.04
IfNumber: 22
SysServices: 79
SysObjectID: enterprises.9.1.379
SysUpTime: 6:16:35:09.00
Telnet application: telnet.exe
Ready

Figure 9-7: General SNMP information gathered using Getif.



.iso.org.dod.internet.private.enterprises

.1.3.6.1.4.1

Type: other

```
enterprises.522.3.7.1.1.2.1  :  admin
enterprises.522.3.7.1.1.2.2  :  public
enterprises.522.3.7.1.1.2.3  :  kbeaver
enterprises.522.3.7.1.1.3.1  :  FF96400055338B7A17AFAC30710F683814CDCEB4911DE429
enterprises.522.3.7.1.1.3.2  :  FF964000D575A1D68D6E0BD09020E8A572013BE12CEDD1CE
enterprises.522.3.7.1.1.3.3  :  FF9640000CFB25AC882CF3F10C523BB866021EE57E890CB4
enterprises.522.3.7.1.1.4.1  :  31
enterprises.522.3.7.1.1.4.2  :  27
enterprises.522.3.7.1.1.4.3  :  31
```

.1.3.6.1.4.1.522.3.7.1.1.2.3 | n (nullobj) | kbeaver | 2119 entry(s)

Ready

Figure 9-8: Management interface user IDs gleaned via Getif's SNMP browsing function.

Countermeasures against SNMP attacks

Preventing SNMP attacks can be as simple as A-B-C:

- **A**lways disable SNMP on hosts if you're not using it — period.
- **B**lock the SNMP port (UDP port 161 and 162) at the network perimeter.
- **C**hange the default SNMP community read string from *public* and the default community write string from *private* to another long and complex value that's virtually impossible to guess.

Banner grabbing

Banners are the welcome screens that divulge software version numbers and other host information to a network host. This banner information may identify the operating system, the version number, and the specific service packs, so hackers know possible vulnerabilities. You can grab banners by using either plain old telnet, some of the tools I've already mentioned such as nmap and SuperScan, or Netcat.

telnet

You can telnet to hosts on the default telnet port (TCP port 23) to see whether you're presented with a login prompt or any other information. Just enter the following line at the command prompt in Windows or UNIX:

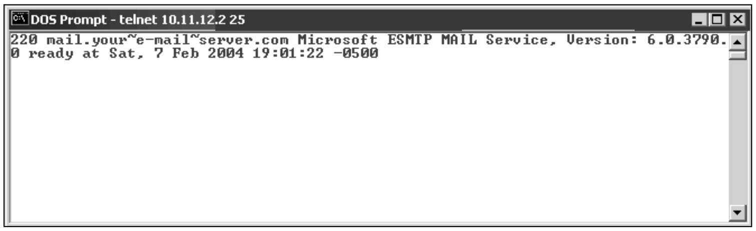
```
telnet ip_address
```

You can telnet to other commonly used ports with these commands:

- **SMTP:** `telnet ip_address 25`
- **HTTP:** `telnet ip_address 80`
- **POP3:** `telnet ip_address 110`

Figure 9-9 shows specific version information about an Exchange 2003 server when telnetting to it on port 25. For help with telnet, simply enter `telnet /?` or `telnet help` for specific guidance on using the program.

Figure 9-9: Information gathered about Exchange 2003 via telnet.



```
220 mail.your~e-mail~server.com Microsoft ESMTP MAIL Service, Version: 6.0.3790.
0 ready at Sat, 7 Feb 2004 19:01:22 -0500
```

Netcat

Netcat, which runs on Linux and Windows, can grab banner information from routers and other network hosts, such as a wireless access point or managed Ethernet switch.

The following steps bring back information about a host that runs a Web server for remote management purposes:

1. **Enter the following line to initiate a connection on port 80:**

   ```
   nc -v ip_address 80
   ```

2. **Wait for the initial connection.**

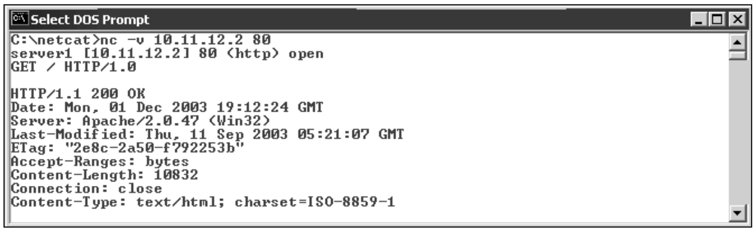
   Netcat returns the message `hostname [ip_address] 80 (http) open`.

3. **Enter the following line to grab the home page of the Web server:**

   ```
   GET / HTTP/1.0
   ```

4. **Press Enter a couple of times to load the page.**

   Figure 9-10 shows some typical results with Netcat.



```
C:\netcat>nc -v 10.11.12.2 80
server1 [10.11.12.2] 80 (http) open
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 01 Dec 2003 19:12:24 GMT
Server: Apache/2.0.47 (Win32)
Last-Modified: Thu, 11 Sep 2003 05:21:07 GMT
ETag: "2e8c-2a50-f792253b"
Accept-Ranges: bytes
Content-Length: 10832
Connection: close
Content-Type: text/html; charset=ISO-8859-1
```

Figure 9-10: A Web-server banner grab using Netcat.

Countermeasures against banner-grabbing attacks

The following steps can reduce the chance of banner-grabbing attacks:

- If there is no business need for services that offer banner information, disable those unused services on the network host.
- If there is no business need for the default banners, or if you can customize the banners displayed, configure the network host's application or operating system to either disable the banners or remove information from the banners that could give an attacker a leg up. Check with your specific vendor or support forum for information on how to do this.

TIP

If you can customize your banners, check with your lawyer about adding a warning message that won't stop banner grabbing but will show that the system is private. Here's an example:

> *Warning!!! This is a private system. All use is monitored and recorded. Any unauthorized use of this system may result in civil and/or criminal prosecution to the fullest extent of the law.*

Firewall rules

As part of your ethical hacking, you can test your firewall rules to make sure they're working like they're supposed to.

Testing

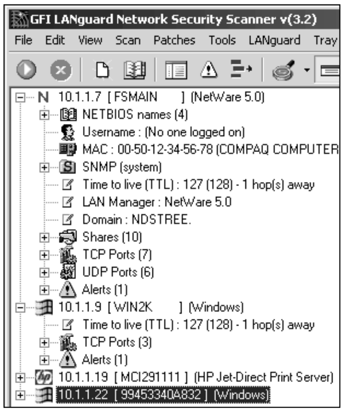
A few tests can verify that your firewall actually does what it says it's doing. You can connect through it on the ports you believe are open, but what about all the other ports that can be open and shouldn't be?

Some security-assessment tools can not only test for open ports, but also determine whether traffic is actually allowed to pass through the firewall.

All-in-one tools

All-in-one tools aren't perfect, but their broad testing capabilities make the network scanning process a lot less painful and can save you tons of time! Their reporting is really nice, too, especially if you will show your test results to upper management.

Nessus, QualysGuard, and GFI LANguard Network Security Scanner provide similar results. Figure 9-11 shows partial output from LANguard. It identifies open ports on the test network and presents information on SNMP, operating-system information, and special alerts to look for.



```
⊟ N 10.1.1.7 [ FSMAIN      ] (NetWare 5.0)
    ⊞ NETBIOS names (4)
      Username : (No one logged on)
      MAC : 00-50-12-34-56-78 (COMPAQ COMPUTER
    ⊞ SNMP (system)
      Time to live (TTL) : 127 (128) - 1 hop(s) away
      LAN Manager : NetWare 5.0
      Domain : NDSTREE.
    ⊞ Shares (10)
    ⊞ TCP Ports (7)
    ⊞ UDP Ports (6)
    ⊞ Alerts (1)
⊟ 10.1.1.9 [ WIN2K      ] (Windows)
      Time to live (TTL) : 127 (128) - 1 hop(s) away
    ⊞ TCP Ports (3)
    ⊞ Alerts (1)
⊞ 10.1.1.19 [ MCI291111 ] (HP Jet-Direct Print Server)
⊞ 10.1.1.22 [ 99453340A832 ] (Windows)
```

Figure 9-11: Information gathered from a network scan using LANguard Network Security Scanner.

TIP

You can use LANguard Network Security Scanner and QualysGuard to find operating system vulnerabilities and patches that need to be applied. Pretty slick! I show you more on this in Chapter 11 when I talk about hacking Windows.

Netcat

Netcat can test certain firewall rules without having to test a production system directly. For example, you can check whether the firewall allows port 23 (telnet) through. Follow these steps to see whether a connection can be made through port 23:

1. **Load Netcat on a client machine *inside* the network.**

   This allows you to test from the inside out.

2. **Load Netcat on a testing computer *outside* the firewall.**

   This allows you to test from the outside in.

3. **Enter the Netcat listener command on the client (internal) machine with the port number you're testing.**

   For example, if you're testing port 23, enter this command:

   ```
   nc –l –p 23 cmd.exe
   ```

4. **Enter the Netcat command to initiate an inbound session on the testing (external) machine. You must include the following information:**

   - The IP address of the internal machine you're testing
   - The port number you're testing

   For example, if the IP address of the internal (client) machine is 10.11.12.2 and the port is 23, enter this command:

   ```
   nc –v 10.11.12.2 23
   ```

If Netcat presents you with a new command prompt (that's what the `cmd.exe` is for in Step 3) on the external machine, it means that you connected and are now executing commands on the internal machine! This can serve several purposes, including testing firewall rules and — well, uhhhmmm — executing commands on a remote system!

A neat commercial tool that specializes in evaluating the performance of packet filtering devices, such as firewalls, is Traffic IQ Pro by Karalon (`www.karalon.com`). With this tool, shown in Figure 9-12, you can connect one NIC on your testing machine to your firewall's internal segment and a second NIC to your firewall's external segment or DMZ and generate generic and/or malicious traffic see if your firewall is doing what it says it's doing. Such a test is great for those annual firewall "rulebase audits" mandated in many organizations.

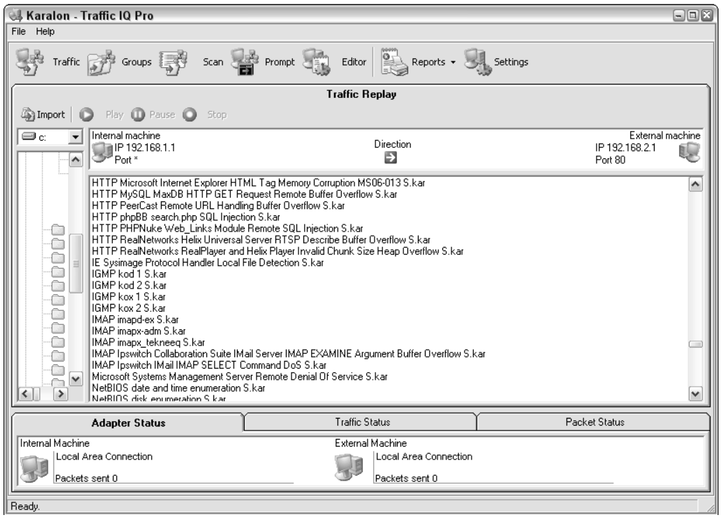
An alternative firewall rulebase testing tool for the UNIX platform is Firewalk (`www.packetfactory.net/firewalk`).

Countermeasures against firewall attacks

The following countermeasures can prevent a hacker from testing your firewall:

- **Limit traffic to what's needed.**

  Set rules on your firewall (and router, if needed) to pass only traffic that absolutely must pass. For example, have rules in place that allow HTTP inbound to an internal Web server and outbound for external Web access.



Internal machine
IP 192.168.1.1
Port *

Direction

External machine
IP 192.168.2.1
Port 80

```
HTTP Microsoft Internet Explorer HTML Tag Memory Corruption MS06-013 S.kar
HTTP MySQL MaxDB HTTP GET Request Remote Buffer Overflow S.kar
HTTP PeerCast Remote URL Handling Buffer Overflow S.kar
HTTP phpBB search.php SQL Injection S.kar
HTTP PHPNuke Web_Links Module Remote SQL Injection S.kar
HTTP RealNetworks Helix Universal Server RTSP Describe Buffer Overflow S.kar
HTTP RealNetworks RealPlayer and Helix Player Invalid Chunk Size Heap Overflow S.kar
IE Sysimage Protocol Handler Local File Detection S.kar
IGMP kod 1 S.kar
IGMP kod 2 S.kar
IGMP kox 1 S.kar
IGMP kox 2 S.kar
IMAP imapd-ex S.kar
IMAP imapx-adm S.kar
IMAP imapx_tekneeq S.kar
IMAP Ipswitch Collaboration Suite IMail Server IMAP EXAMINE Argument Buffer Overflow S.kar
IMAP Ipswitch IMail IMAP SELECT Command DoS S.kar
Microsoft Systems Management Server Remote Denial Of Service S.kar
NetBIOS date and time enumeration S.kar
NetBIOS disk enumeration S.kar
```

Internal Machine
Local Area Connection
Packets sent 0

External Machine
Local Area Connection
Packets sent 0

Figure 9-12: Traffic IQ Pro for generating packets and analyzing a firewall's capabilities.

This is the best defense against someone poking at your firewall.

- **Block ICMP to help prevent abuse from some automated tools, such as Firewalk.**
- **Enable stateful packet inspection on the firewall, if you can. It can block unsolicited requests.**

Network analyzers

A *network analyzer* is a tool that allows you to look into a network and analyze data going across the wire for network optimization, security, and/or troubleshooting purposes. Like a microscope for a lab scientist, a network analyzer is a must-have tool for any security professional.

Network analyzers are often generically referred to as *sniffers,* though that's actually the name and trademark of a specific product from Network Associates, *Sniffer* (the original commercial network analysis tool).

A network analyzer is handy for *sniffing* packets off the wire. Watch for the following network traffic behavior when using a network analyzer:

- What do packet replies look like? Are they coming from the host you're testing or from an intermediary device?
- Do packets appear to traverse a network host or security device, such as a router, a firewall, or a proxy server?

When assessing security and responding to security incidents, a network analyzer can help you

- View anomalous network traffic and even track down an intruder.
- Develop a baseline of network activity and performance, such as protocols in use, usage trends, and MAC addresses, before a security incident occurs.

When your network behaves erratically, a network analyzer can help you

- Track and isolate malicious network usage
- Detect malicious Trojan-horse applications
- Monitor and track down DoS attacks

Network analyzer programs

You can use one of the following programs for network analysis:

- **WildPackets EtherPeek** (`www.wildpackets.com/products/etherpeek/overview`) is my favorite network analyzer. It does everything I need and more and is very simple to use. EtherPeek is available for the Windows operating systems.

  If you're going to be doing a lot of network analysis on both wired and wireless networks that may require the decoding of Gigabit Ethernet, WAN protocols, voice over IP, and other advanced systems, you should check out WildPackets OmniPeek product line (`www.wildpackets.com/products/omni/overview/omnipeek_analyzers`). OmniPeek offers an all-in-one solution to help you keep your network analysis costs down plus you get the benefit of being able to use one tool for everything.
- **TamoSoft's CommView** (`www.tamos.com/products/commview`) and **Sunbelt Software's LanHound** (`www.sunbelt-software.com/LanHound.cfm`) are low-cost, Windows-based alternatives.
- **Cain and Abel** (`www.oxid.it/cain.html`) is a free alternative for performing network analysis, ARP poisoning, Voice over IP capture/replay, password cracking, and more.
- **Ethereal** (`www.ethereal.org`) is a free alternative. I download and use this tool if I need a quick fix and don't have my laptop nearby. It's not as user-friendly as most of the commercial products, but it is very powerful if you're willing to learn its ins and outs. Ethereal is available for both Windows and UNIX-based operating systems.
- **ettercap** (`http://ettercap.sourceforge.net`) is another powerful (and free) utility for performing network analysis and much more on both Windows and UNIX-based operating systems.

A network analyzer is simply software running on a computer with a network card. It works by placing the network card in *promiscuous mode,* which enables the card to see all the traffic on the network, even traffic not destined for the network analyzer's host. The network analyzer performs the following functions:

- Captures all network traffic
- Interprets or decodes what is found into a human-readable format
- Displays it all in chronological order

Here are a few caveats for using a network analyzer:

- To capture all traffic, you must connect the analyzer to either
  - A hub on the network
  - A monitor/span/mirror port on a switch
  - A switch that you've performed an ARP poisoning attack on
- You should connect the network analyzer to a hub on the outside of the firewall, as shown in Figure 9-13, as part of your testing so you can see traffic similar to what a network-based IDS sees:
  - What's entering your network *before* the firewall filters eliminate the junk traffic
  - What's leaving your network *after* the traffic goes past the firewall

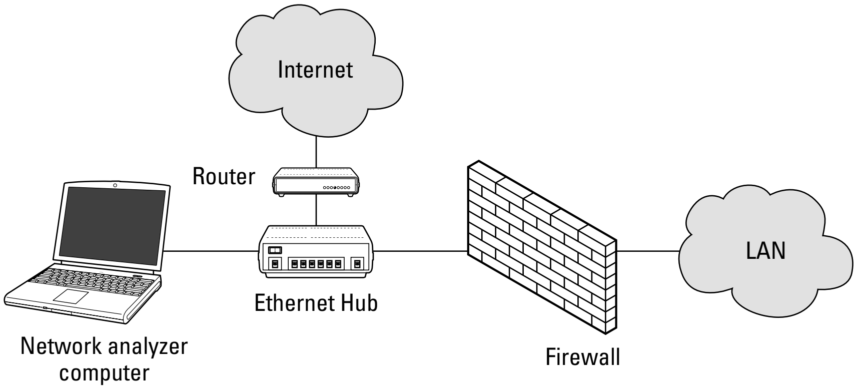


Figure 9-13: Connecting a network analyzer outside the firewall.

Whether you connect your network analyzer inside or outside your firewall, you see immediate results. It can be an overwhelming amount of information, but you can look for these issues first:

- **Odd traffic,** such as
  - An unusual amount of ICMP packets
  - Excessive amounts of multicast or broadcast traffic
  - Packet types that don't belong, such as NetBIOS in a NetWare environment

- **Internet usage habits,** which can help point out malicious behavior of a rogue insider or system that has been compromised, such as
  - Web surfing
  - E-mail
  - IM and other P2P software
- **Questionable usage,** such as
  - Many lost or oversized packets
  - High bandwidth consumption that may point to a Web or FTP server that doesn't belong
- **Reconnaissance probes and system profiling from port scanners and vulnerability assessment tools,** such as a significant amount of inbound traffic from unknown hosts — especially over ports that are not used very much, such as FTP or telnet.
- **Hacking in progress,** such as tons of inbound UDP or ICMP echo requests, SYN floods, or excessive broadcasts.
- **Nonstandard hostnames on your network.** For example, if your systems are named `Computer1`, `Computer2`, and so on, a computer named `GEEKz4evUR` should raise a red flag.
- **Hidden servers** (especially Web, SMTP, FTP, and DHCP) that may be eating network bandwidth or serving illegal software or even access into your network hosts.
- **Attacks on specific applications** that show such commands as `/bin/rm`, `/bin/ls`, `echo`, and `cmd.exe`.

You may need to let your network analyzer run for quite a while — several hours to several days, depending on what you're looking for. Before getting started, configure your network analyzer to capture and store the most relevant data:

- If your network analyzer permits it, configure your network analyzer software to use a first-in, first-out buffer.

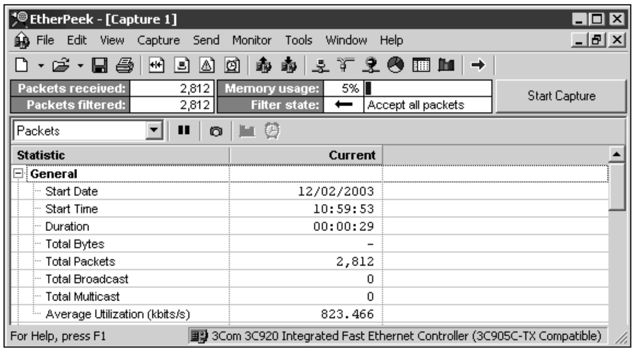
  This overwrites the oldest data when the buffer fills up, but it may be your only option if memory and hard drive space are limited on your network-analysis computer.
- If your network analyzer permits it, record all the traffic into a capture file and save it to the hard drive. This is the ideal scenario — especially if you have a large hard drive, such as 50GB or more.

  You can easily fill a several-gigabyte hard drive in a short period of time. I highly recommend running your network analyzer in what EtherPeek calls *monitor mode*. This allows the analyzer to keep track of what's going on but not capture every single packet. Monitor mode — if supported by your analyzer — is very beneficial and is often all you need.

- When network traffic doesn't look right in a network analyzer, it probably isn't. It's better to be safe than sorry.

  Run a baseline when your network is working normally. When you have a baseline, you can see any obvious abnormalities when an attack occurs.

Figure 9-14 shows what the well-known Smurf DoS attack (`http://en.wikipedia.org/wiki/Smurf_attack`) can do to a network in just 30 seconds. (I created this attack with BLADE Software's IDS Informer, but you can use other tools.) On a small network with very little traffic, the utilization number is 823 Kbps — not too large a number for a 100 Mbps Ethernet network. However, on a busy network with a lot more traffic, the number would be staggering.



Packets received: 2,812
Packets filtered: 2,812
Memory usage: 5%
Filter state: ⟵ Accept all packets

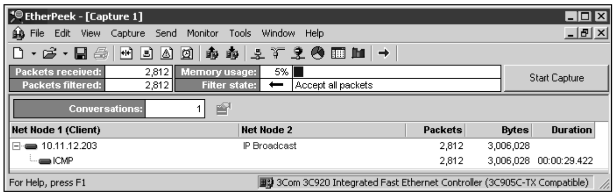
Packets

Statistic	Current
⊟ General	
Start Date	12/02/2003
Start Time	10:59:53
Duration	00:00:29
Total Bytes	-
Total Packets	2,812
Total Broadcast	0
Total Multicast	0
Average Utilization (kbits/s)	823.466

For Help, press F1
3Com 3C920 Integrated Fast Ethernet Controller (3C905C-TX Compatible)

Figure 9-14: What a Smurf DoS attack looks like through a network analyzer.

Figure 9-15 shows the Smurf DoS attack on EtherPeek's conversation monitor. Three million bytes were transmitted in this short period of time — all from one host!



Packets received: 2,812
Packets filtered: 2,812
Memory usage: 5%
Filter state: ⟵ Accept all packets

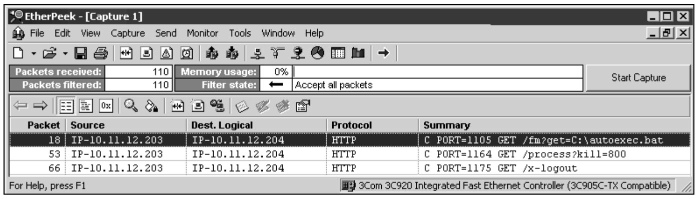
Conversations: 1

Net Node 1 (Client)	Net Node 2	Packets	Bytes	Duration
⊟ 10.11.12.203	IP Broadcast	2,812	3,006,028	
ICMP		2,812	3,006,028	00:00:29.422

For Help, press F1
3Com 3C920 Integrated Fast Ethernet Controller (3C905C-TX Compatible)

Figure 9-15: A Smurf DoS conversation via EtherPeek.

Figure 9-16 shows what the backdoor tool, WANRemote (`www.megasecurity.org/trojans/w/wanremote/Wanremote3.0.html`) remote administration tool (RAT) looks like across the network via EtherPeek. It shows the commands sent to get files from the local C: drive, to kill UNIX processes, and to unload X-Window.



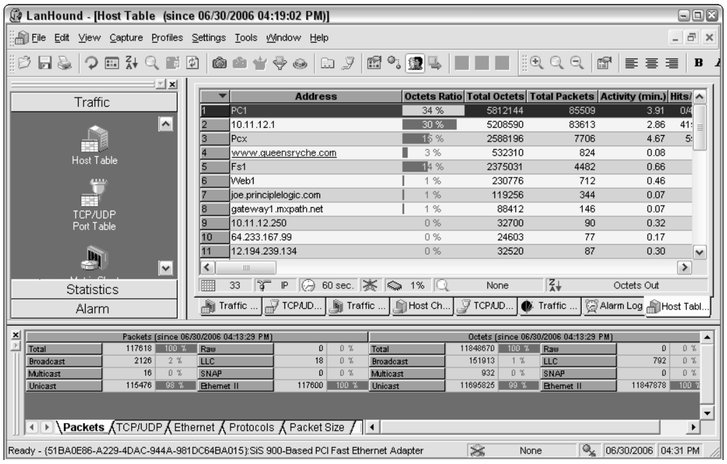
Packets received: 110 Memory usage: 0%
Packets filtered: 110 Filter state: ⟵ Accept all packets

Packet	Source	Dest. Logical	Protocol	Summary
18	IP-10.11.12.203	IP-10.11.12.204	HTTP	C PORT=1105 GET /fm?get=C:\autoexec.bat
53	IP-10.11.12.203	IP-10.11.12.204	HTTP	C PORT=1164 GET /process?kill=800
66	IP-10.11.12.203	IP-10.11.12.204	HTTP	C PORT=1175 GET /x-logout

For Help, press F1 — 3Com 3C920 Integrated Fast Ethernet Controller (3C905C-TX Compatible)

Figure 9-16: WAN Remote RAT-attack traffic.

If one workstation consumes considerably more bandwidth than the others — such as the 10.11.12.1 host highlighted in the LanHound capture in Figure 9-17 — dig deeper to see what's going on. (Network hosts, such as servers, often send and receive more traffic than other hosts.)



	Address	Octets Ratio	Total Octets	Total Packets	Activity (min.)	Hits/
1	PC1	34 %	5812144	85509	3.91	0/4
2	10.11.12.1	30 %	5208590	83613	2.86	41:
3	Pcx	15 %	2588196	7706	4.67	5:
4	www.queensryche.com	3 %	532310	824	0.08	
5	Fs1	14 %	2375031	4482	0.66	
6	Web1	1 %	230776	712	0.46	
7	joe.principlelogic.com	1 %	119256	344	0.07	
8	gateway1.mxpath.net	1 %	88412	146	0.07	
9	10.11.12.250	0 %	32700	90	0.32	
10	64.233.167.99	0 %	24603	77	0.17	
11	12.194.239.134	0 %	32520	87	0.30	

Packets (since 06/30/2006 04:13:29 PM)						Octets (since 06/30/2006 04:13:29 PM)					
Total	117618	100 %	Raw	0	0 %	Total	11848670	100 %	Raw	0	0 %
Broadcast	2126	2 %	LLC	18	0 %	Broadcast	151913	1 %	LLC	792	0 %
Multicast	16	0 %	SNAP	0	0 %	Multicast	932	0 %	SNAP	0	0 %
Unicast	115476	98 %	Ethernet II	117600	100 %	Unicast	11695825	99 %	Ethernet II	11847878	100 %

Ready - {51BA0E86-A229-4DAC-944A-981DC64BA015}:SiS 900-Based PCI Fast Ethernet Adapter — None — 06/30/2006 04:31 PM

Figure 9-17: Higher-than-normal network usage discovered by LanHound.

Check your network for a high number of ARP requests and ICMP echo requests proportionate to your overall traffic, as shown in Figure 9-18.

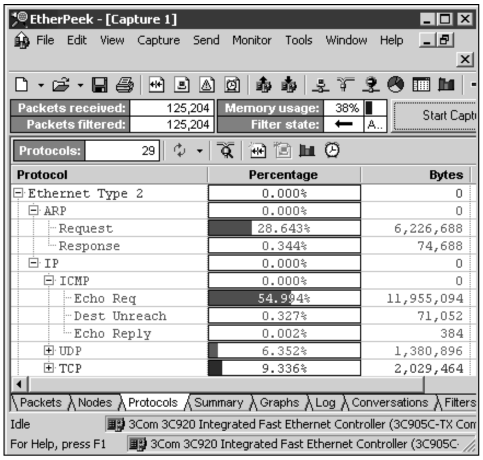
Figure 9-19 shows CommView indicating that a port scan or other malicious attack is being carried out on the network. It shows all the different protocols and the small number of packets this analysis found, including Gnutella, telnet, and rlogin.

Countermeasures against network analyzer attacks

A network analyzer can be used for good or evil. All these tests can be used against you, too. A few countermeasures can help prevent someone from using an unauthorized network analyzer, but there's no way to completely prevent it.

WARNING!

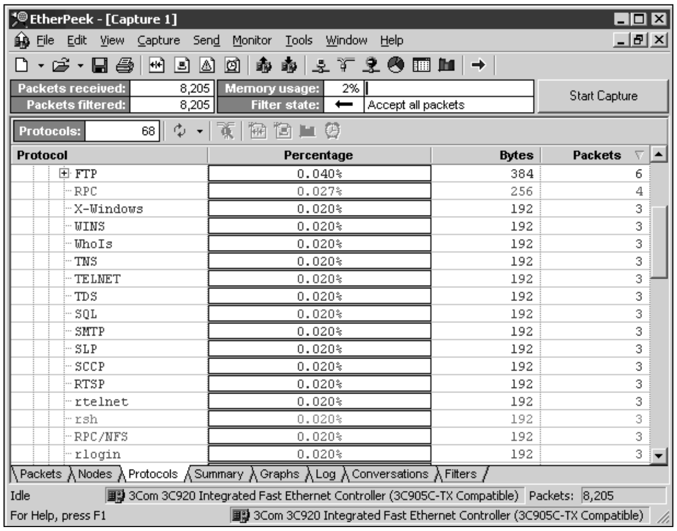
If hackers can connect to your network (physically or wirelessly), they can capture packets on the network, even if you're using a switch.


EtherPeek - [Capture 1]
File Edit View Capture Send Monitor Tools Window Help
Start Captu
Packets Nodes Protocols Summary Graphs Log Conversations Filters
Idle 3Com 3C920 Integrated Fast Ethernet Controller (3C905C-TX Com
For Help, press F1 3Com 3C920 Integrated Fast Ethernet Controller (3C905C-

Packets received: 125,204
Packets filtered: 125,204
Memory usage: 38%
Filter state: ⟵ A..
Protocols: 29

Protocol	Percentage	Bytes
⊟ Ethernet Type 2	0.000%	0
⊟ ARP	0.000%	0
Request	28.643%	6,226,688
Response	0.344%	74,688
⊟ IP	0.000%	0
⊟ ICMP	0.000%	0
Echo Req	54.994%	11,955,094
Dest Unreach	0.327%	71,052
Echo Reply	0.002%	384
⊞ UDP	6.352%	1,380,896
⊞ TCP	9.336%	2,029,464

Figure 9-18: Abnormally high ICMP and ARP requests show potential malicious behavior.


EtherPeek - [Capture 1]
File Edit View Capture Send Monitor Tools Window Help
Start Capture
Packets Nodes Protocols Summary Graphs Log Conversations Filters
Idle 3Com 3C920 Integrated Fast Ethernet Controller (3C905C-TX Compatible) Packets: 8,205
For Help, press F1 3Com 3C920 Integrated Fast Ethernet Controller (3C905C-TX Compatible)

Packets received: 8,205
Packets filtered: 8,205
Memory usage: 2%
Filter state: ⟵ Accept all packets
Protocols: 68

Protocol	Percentage	Bytes	Packets
⊞ FTP	0.040%	384	6
RPC	0.027%	256	4
X-Windows	0.020%	192	3
WINS	0.020%	192	3
WhoIs	0.020%	192	3
TNS	0.020%	192	3
TELNET	0.020%	192	3
TDS	0.020%	192	3
SQL	0.020%	192	3
SMTP	0.020%	192	3
SLP	0.020%	192	3
SCCP	0.020%	192	3
RTSP	0.020%	192	3
rtelnet	0.020%	192	3
rsh	0.020%	192	3
RPC/NFS	0.020%	192	3
rlogin	0.020%	192	3

Figure 9-19: Non-standard protocols can indicate a port scan or other malicious use.

Physical security

Ensure that adequate physical security is in place to prevent a hacker from plugging into your network:

WARNING!

- **Keep the bad guys out of your server room and wiring closet.**

  A special monitor port on a switch where a hacker can plug in a network analyzer is especially sensitive. Make sure it's extra secure.

- **Make sure that unsupervised areas, such as unoccupied desks, don't have live network connections.**

Network analyzer detection

You can use a network- or host-based utility to determine whether someone is running an unauthorized network analyzer on your network:

- **Sniffdet** (`http://sniffdet.sourceforge.net`) for UNIX-based systems
- **PromiscDetect** (`http://ntsecurity.nu/toolbox/promiscdetect`) for Windows

These tools enable you to monitor the network for Ethernet cards that are running in promiscuous mode. You simply load the programs on your computer, and the programs alert you if they see promiscuous behaviors on the network (Sniffdet) or local system (PromiscDetect).

The MAC-daddy attack

Attackers can use ARP (Address Resolution Protocol) running on your network to make their systems appear to be either your system or another authorized host on your network.

ARP spoofing

An excessive number of ARP requests can be a sign of an *ARP spoofing* attack (also called *ARP poisoning*) on your network.

A client running a program such as the UNIX-based dsniff or the UNIX- and Windows-based Cain and Abel can change the ARP tables — the tables that store IP addresses to *media access control (MAC)* address mappings — on network hosts. This causes the victim computers to think they need to send traffic to the attacker's computer rather than to the true destination computer when communicating on the network. This is often referred to as a Man-in-the-Middle (MITM) attack.

Spoofed ARP replies can be sent to a switch very quickly, which can crash an Ethernet switch or (hopefully) make it revert to *broadcast mode,* which essentially turns it into a hub. When this occurs, an attacker can sniff every packet going through the switch without bothering with ARP spoofing.

REMEMBER

This security vulnerability is inherent in how TCP/IP communications are handled.

Here's a typical ARP spoofing attack with a hacker's computer (Hacky) and two legitimate network users' computers (Joe and Bob):

1. Hacky poisons the ARP caches of victims Joe and Bob by using dsniff, ettercap, or a utility he wrote.
2. Joe associates Hacky's MAC address with Bob's IP address.

3. Bob associates Hacky's MAC address with Joe's IP address.
4. Joe's traffic and Bob's traffic are sent to Hacky's IP address first.
5. Hacky's network analyzer captures Joe's and Bob's traffic.

   If Hacky is configured to act like a router and forward packets, it forwards the traffic to its original destination. The original sender and receiver never know the difference!

Using Cain and Abel for ARP poisoning

You can perform ARP poisoning on your switched Ethernet network to test your IDS/IPS or to see how easy it is to turn a switch into a hub and capture anything and everything with a network analyzer.

ARP poisoning can be hazardous to your network's hardware and health, causing downtime and more. So be careful!

Perform the following steps to use Cain and Abel for ARP poisoning:

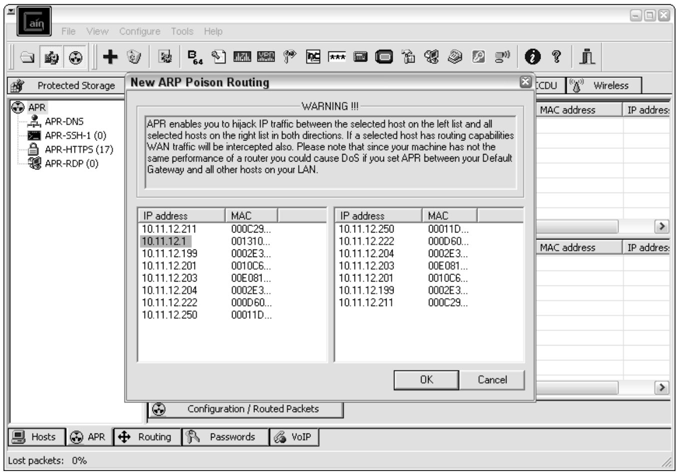
1. **Load Cain and Abel and click the Sniffer tab at the top to get into the network analyzer mode. It defaults to the Hosts page.**
2. **Click the Start/Stop APR icon (the yellow and black circle).**

   This starts the ARP poison routing (how Cain and Abel refers to ARP poisoning) process and also enables the built-in sniffer.
3. **If prompted, select the network adapter in the window that displays and click OK.**
4. **Click the blue + icon to add hosts to perform ARP poisoning on.**
5. **On the MAC Address Scanner window that comes up, ensure the All Hosts in My Subnet option is selected and click OK.**
6. **Click the APR tab (the one with the yellow and black circle icon) at the bottom to load the APR page.**
7. **Click in the white space under the uppermost Status column heading (just under the Sniffer tab).**

   This re-enables the blue + icon.
8. **Click the blue + icon, and the New ARP Poison Routing window comes up showing the hosts discovered in Step 3 above.**
9. **Select your default route (in my case, 10.11.12.1).**

   This will then fill the right-hand column with all the remaining hosts, as shown in Figure 9-20.
10. **Ctrl+click all the hosts in the right column that you want to poison.**
11. **Click OK, and the ARP poisoning process starts.**

This process can take anywhere from a few seconds to a few minutes depending on your network hardware and each hosts' local TCP/IP stack. The results of ARP poisoning on my test network are shown in Figure 9-21.



New ARP Poison Routing

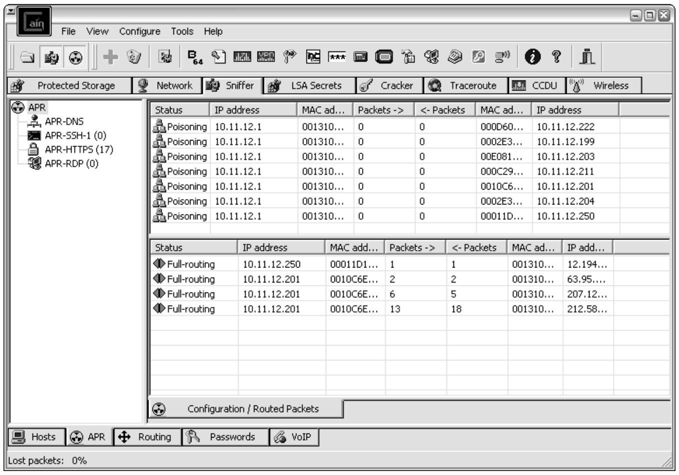
WARNING !!!

APR enables you to hijack IP traffic between the selected host on the left list and all selected hosts on the right list in both directions. If a selected host has routing capabilities WAN traffic will be intercepted also. Please note that since your machine has not the same performance of a router you could cause DoS if you set APR between your Default Gateway and all other hosts on your LAN.

IP address	MAC
10.11.12.211	000C29...
10.11.12.1	001310...
10.11.12.199	0002E3...
10.11.12.201	0010C6...
10.11.12.203	00E081...
10.11.12.204	0002E3...
10.11.12.222	000D60...
10.11.12.250	00011D...

IP address	MAC
10.11.12.250	00011D...
10.11.12.222	000D60...
10.11.12.204	0002E3...
10.11.12.203	00E081...
10.11.12.201	0010C6...
10.11.12.199	0002E3...
10.11.12.211	000C29...

Figure 9-20: Selecting your victim hosts for ARP poisoning in Cain and Abel.



Status	IP address	MAC ad...	Packets ->	<- Packets	MAC ad...	IP address
Poisoning	10.11.12.1	001310...	0	0	000D60...	10.11.12.222
Poisoning	10.11.12.1	001310...	0	0	0002E3...	10.11.12.199
Poisoning	10.11.12.1	001310...	0	0	00E081...	10.11.12.203
Poisoning	10.11.12.1	001310...	0	0	000C29...	10.11.12.211
Poisoning	10.11.12.1	001310...	0	0	0010C6...	10.11.12.201
Poisoning	10.11.12.1	001310...	0	0	0002E3...	10.11.12.204
Poisoning	10.11.12.1	001310...	0	0	00011D...	10.11.12.250

Status	IP address	MAC add...	Packets ->	<- Packets	MAC ad...	IP add...
Full-routing	10.11.12.250	00011D1...	1	1	001310...	12.194...
Full-routing	10.11.12.201	0010C6E...	2	2	001310...	63.95....
Full-routing	10.11.12.201	0010C6E...	6	5	001310...	207.12...
Full-routing	10.11.12.201	0010C6E...	13	18	001310...	212.58...

Figure 9-21: ARP poisoning end results in Cain and Abel.

12. **You can use Cain and Abel's built-in passwords feature to capture passwords traversing the network to and from various hosts simply by clicking the Passwords tab at the bottom of the screen.**

The preceding steps show how easy it is to exploit a vulnerability and prove that Ethernet switches aren't all they're cracked up to be from a security perspective.

MAC address spoofing

MAC address spoofing tricks the *switch* into thinking your computer is something else. You simply change your computer's MAC address and masquerade as another user.

TIP

You can use this trick to test access control systems, like your IDS, firewall, and even operating system login controls that check for specific MAC addresses.

UNIX-based systems

In UNIX and Linux, you can spoof MAC addresses with the ifconfig utility. Follow these steps:

1. **While logged in as root, use ifconfig to enter a command that disables the network interface. Insert the network interface number that you want to disable (usually, eth0) into the command, like this:**

   ```
   [root@localhost root]# ifconfig eth0 down
   ```

2. **Enter a command for the MAC address you want to use.**

   Insert the fake MAC address and the network interface number (eth0) into the command again, like this:

   ```
   [root@localhost root]# ifconfig eth0 hw ether new_mac_address
   ```

TIP

You can use a more feature-rich utility called GNU MAC Changer (`www.alobbs.com/macchanger`) for Linux systems.

Windows

You can use regedit to edit the Windows Registry, but I like using a neat Windows utility called SMAC (`www.klcconsulting.net/smac`), which makes MAC spoofing a simple process. Follow these steps to use SMAC:

1. **Load the program.**
2. **Select the adapter for which you want to change the MAC address.**
3. **Enter the new MAC address in the New Spoofed MAC Address fields and click the Update MAC button.**
4. **Stop and restart the network card with these steps:**
   a. *Right-click the network card in Network and Dialup Connections and select Disable.*
   b. *Right-click again and select Enable for the change to take effect.*

   TIP

   You may have to reboot for this to work properly.
5. **Click the Refresh button in the SMAC interface.**

To reverse Registry changes with SMAC, follow these steps:

1. **Select the adapter for which you want to change the MAC address.**
2. **Click the Remove MAC button.**
3. **Stop and restart the network card with these steps:**
   a. *Right-click the network card in Network and Dialup Connections and select Disable.*
   b. *Right-click again and select Enable for the change to take effect.*

      You may have to reboot for this to work properly.
4. **Click the Refresh button in the SMAC interface.**

   You should see your original MAC address again.

Countermeasures against ARP poisoning and MAC address spoofing attacks

A few countermeasures on your network can minimize the effects of a hacker attack against ARP and MAC addresses on your network.

Prevention

You can prevent MAC address spoofing if your switches can enable port security to prevent automatic changes to the switch MAC address tables.

No realistic countermeasures for ARP poisoning exist. The only way to prevent ARP poisoning is to create and maintain static ARP entries in your switches for every host on the network. This is definitely something that no network administrator has time to do!

Detection

You can detect these two types of hacks through either an IDS, IPS, or a standalone MAC address monitoring utility.

Arpwatch (`http://linux.maruhn.com/sec/arpwatch.html`) is a UNIX-based program that alerts you via e-mail if it detects changes in MAC addresses associated with specific IP addresses on the network.

Denial of service

Denial-of-service (DoS) attacks are among the most common hacker attacks. A hacker initiates so many invalid requests to a network host that it uses all its resources responding to them and ignores legitimate requests.

DoS attacks

The following types of DoS attacks are possible against your network and hosts and can cause systems to crash, data to be lost, and every user to jump on your case wondering when Internet access will be restored.

Individual attacks

Here are some common DoS attacks:

- **SYN floods:** The attacker floods a host with TCP SYN packets.
- **Ping of Death:** The attacker sends IP packets that exceed the maximum length of 65,535 bytes, which can ultimately crash the TCP/IP stack on many operating systems.
- **WinNuke:** This attack can disable networking on older Windows 95 and NT computers.

Distributed attacks

Distributed DoS (DDoS) attacks have an exponentially greater impact on their victims. The most famous was the DDoS attack against eBay, Yahoo!, CNN, and dozens of other Web sites by a hacker known as MafiaBoy. These are some common distributed attacks:

- **Smurf attack:** An attacker spoofs the victim's address and sends ICMP echo requests (ping packets) to the broadcast address. The victim computer gets deluged with tons of packets in response to those echo requests.
- **Trinoo and Tribe Flood Network (TFN) attacks:** Sets of client- and server-based programs launch packet floods against a victim machine, effectively overloading it and causing it to crash.

DoS and DDoS attacks can be carried out with tools that the hacker either writes or downloads off the Internet. These are good tools to test your network's IDS/IPS and firewalls. You can find programs that allow actual attacks and programs, such as Karalon's Traffic IQ Pro, that let you send controlled attacks.

Testing

Your first DoS test should be a search for DoS vulnerabilities from a port-scanning and network analysis perspective.

WARNING!

Don't test for DoS unless you have test systems or can perform controlled tests with the proper tools. Poorly planned DoS testing is a job search in the making. It's like trying to delete data from a network share remotely and hoping that the access controls in place are going to prevent it.

Countermeasures against DoS attacks

Most DoS attacks are difficult to predict, but they can be easy to prevent:

- **Test and apply security patches as soon as possible** for network hosts such as routers and firewalls, as well as for server and workstation operating systems.
- **Use an IDS or IPS to monitor regularly for DoS attacks.**

You can run a network analyzer in *continuous capture* mode if you can't justify the cost of an all-out IDS or IPS solution.

- **Configure firewalls and routers to block malformed traffic.** You can do this only if your systems support it, so refer to your administrator's guide for details.
- **Minimize IP spoofing** by either
    - Using authentication and encryption, such as a Public Key Infrastructure (PKI).
    - Filtering out external packets that appear to come from an internal address, the local host (127.0.0.1), or any other private and non-routable address, such as 10.x.x.x, 172.16.x.x–172.31.x.x, or 192.168.x.x.
- **Block all ICMP traffic inbound to your network unless you specifically need it.** Even then, you should allow it to come in only to specific hosts.
- **Disable all unneeded TCP/UDP small services,** such as echo and chargen.

Establish a baseline of your network protocols and traffic patterns before a DoS attack occurs. That way, you know what to look for. And periodically scan for such potential DoS vulnerabilities as rogue DoS software installed on network hosts.

Work with a *minimum necessary* mentality (not to be confused with having too many beers) when configuring your network devices, such as firewalls and routers:

- **Identify traffic that is necessary for approved network usage.**
- **Allow the traffic that's needed.**
- **Deny all other traffic.**

General Network Defenses

Regardless of the specific attacks against your system, a few good practices can help prevent many network problems:

- **Use stateful inspection rules that monitors traffic sessions for firewalls.** This can help ensure that all traffic traversing the firewall is legitimate and can prevent DoS attacks and other spoofing attacks.
- **Implement rules to perform packet filtering** based on traffic type, TCP/UDP ports, IP addresses, and even specific interfaces on your routers before the traffic is ever allowed to enter your network.

- **Use proxy filtering and Network Address Translation (NAT).**
- **Find and eliminate fragmented packets entering your network** (from Fraggle or another type of attack) via an IDS or IPS system.
- **Segment the network and use a firewall on**
  - The internal network in general.
  - Critical departments, such as accounting, finance, HR, and research.