

FAQ: Firewall Forensics (What am I seeing?)

This document explains what you see in firewall logs, especially what port numbers means. You can use this information to help figure out what hackers are up to.

This document is intended for both security-experts maintaining corporate firewalls as well as home users of **personal firewalls**.

0. Information about this FAQ

Version 0.4.1, June 20, 2000

<http://www.robertgraham.com/pubs/firewall-seen.html>

Copyright 1998-2000 by Robert Graham (firewall-seen@robertgraham.com).

All rights reserved. This document may only be reproduced (whole or in part) for non-commercial purposes. All reproductions must contain this copyright notice and must not be altered, except by permission of the author.

Special thanks to Alan J. Rosenthal (maintainer of FAQs himself) for some really good input.

TOC

[1. What does destination port number ZZZZ mean?](#)

[PORT GUIDE](#) | [source-ports](#) | [many-to-one](#) | [trojans](#) | [DNS](#) | [dial-up](#) | [IRC](#) | [remapping](#) | [still can't figure it out](#)

[2. What does this ICMP info mean?](#)

[0 \(echo reply\)](#) | [3 \(unreachable\)](#) | [4 \(source quench\)](#) | [8 \(ping\)](#) | [11 \(ttl exceeded\)](#) | [12 \(problem\)](#)

[3. What do these IP addresses indicate?](#)

[source-routing](#) | [255.255.255.255](#) | [track owner](#) | [10.x.x.x](#) | [known IP addresses](#) | [0.0.0.0](#) | [directed-broadcasts](#) | [169.254.x.x](#)

[4. Stuff doesn't work](#)

[slow connections](#)

[5. What are some typical signatures of well-known programs?](#)

[traceroute](#) | [sscan](#) | [proxy scanners](#) | [smurf](#) | [fraggle](#)

[7. What do these other logs mean?](#)

[DNS](#) | [HTTP](#) | [RPC](#) | [SMTP](#) | [identd](#)

[8. How do I configure filters?](#)

[ICMP filters](#) | [split DNS](#)

[9. Packet Zen](#)

[IP ID](#) | [TTL](#) | [Resources](#)

[10. What's the deal with NetBIOS \(UDP port 137\)?](#)

[What?](#) | [Why?](#) | [But I'm not Win?](#) | [Statistics](#) | [Signature](#) | [Get rid of them?](#) | [Attacks](#)

[A. Appendix](#)

You'll note that some sections are missing. This is an evolving document; when sections are removed (because the info is moved into other sections), I don't renumber the document.

1. What does destination port number ZZZZ mean?

All the traffic going through the firewall is part of a **connection**. A connection consists of the pair of IP addresses that are talking to each other, as well a pair of **port** numbers. The **destination** port number often indicates the type of service being connected to. When a firewall blocks a connection, it will save the destination port number to its logfile. This section describes some of the meanings of these port numbers.

Port numbers are divided into three ranges:

- The Well Known Ports are those from 0 through 1023. These are tightly bound to services, and usually traffic on this port clearly indicates the protocol for that service. For example, port 80 virtually always indicates HTTP traffic.
- The Registered Ports are those from 1024 through 49151. These are loosely bound to services, which means that while there are numerous services "bound" to these ports, these ports are likewise used for many other purposes. For example, most systems start handing out dynamic ports starting around 1024.
- The Dynamic and/or Private Ports are those from 49152 through 65535. In theory, no service should be assigned to these ports.

In reality, machines start assigning "dynamic" ports starting at 1024. We also see strangeness, such as Sun starting their RPC ports at 32768.

Where to get a more complete list of port info:

<ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers>

"Assigned Numbers" RFC, the official source for port assignments.

<http://advice.networkice.com/advice/Exploits/Ports/>

Database of port numbers, hyper-linked to various exploits on those port numbers.

`/etc/services`

On UNIX systems, the file `/etc/services` contains a list of commonly used UNIX port number assignments. On Windows NT, this file is located in `%systemroot%/system32/drivers/etc/services`.

<http://www.con.wesleyan.edu/~triemer/network/docservs.html>

Links back to the protocol specifications frequently.

<http://www.chebucto.ns.ca/~rakerman/trojan-port-table.html>

Pages describing various ports.

<http://www.tlsecurity.com/trojanh.htm>

TLSecurity's list of Trojans. Rather than a collection of rumors by other people, the maintainers of this list claim to verify each and every port personally.

<http://www.simovits.com/nyheter9902.html>

Trojan Horse probes page.

1.1 What are some common incoming TCP/UDP probes against my firewall?

This section contains a list of common TCP and UDP port scans that people see against their firewalls. Note: there is no such thing as an **ICMP port**. If you are interested in interpreting ICMP data, look in [section 2](#).

0

Commonly used to help determine the operating system. This works because on some systems, port 0 is "invalid" and will generate a different response when you connect to it vs. a normal closed port. One typical scan uses a destination IP address of 0.0.0.0 and sets the ACK bit, with broadcast at the Ethernet layer.

1	tcpmux	Indicates someone searching for SGI Irix machines. Irix is the only major vendor that has implemented tcpmux, and it is enabled by default on Irix machines. Irix machines ship with several default passwordless accounts, such as lp, guest, uucp, nuucp, demos, tutor, diag, EZsetup, OutOfBox, and 4Dgifts. Many administrators forget to close these accounts after installation. Therefore, hackers scan the Internet looking first for tcpmux, then these accounts. [CA-95.15]
7	Echo	<p>You will see lots of these from people looking for fraggle amplifiers sent to addresses of x.x.x.0 and x.x.x.255.</p> <p>A common DoS attack is an <i>echo-loop</i>, where the attacker forges a UDP from one machine and sends it to the other, then both machines bounce packets off each other as fast as they can (see also chargen). [CA-96.01]</p> <p>Another common thing seen is TCP connections to this port by DoubleClick. They use a product called "Resonate Global Dispatch" that connects to this port on DNS servers in order to locate the closest one.</p> <p>Harvest/squid caches will send UDP echoes from port 3130. To quote: <i>If the cache is configured with source_ping on, it also bounces a HIT reply off the original host's UDP echo port. It can generate a lot of these packets.</i></p>
11	sysstat	This is a UNIX service that will list all the running processes on a machine and who started them. This gives an intruder a huge amount of information that might be used to compromise the machine, such as indicating programs with known vulnerabilities or user accounts. It is similar the contents that can be displayed with the UNIX "ps" command. ICMP doesn't have ports; if you see something that says "ICMP port 11", you probably want ICMP type=11 .
19	chargen	This is a service that simply spits out characters. The UDP version will respond with a packet containing garbage characters whenever a UDP packet is received. On a TCP connection, it spits out a stream of garbage characters until the connection is closed. Hackers can take advantage of IP spoofing for denial of service attacks. Forging UDP packets between two chargen servers, or a chargen and echo can overload links as the two servers attempt to infinitely bounce the traffic back and forth. Likewise, the " fraggle " DoS attack broadcasts a packet destined to this port with a forged victim address, and the victim gets overloaded with all the responses. [CA-96.01]
21	FTP	The most common attack you will see are hackers/crackers looking for "open anonymous" FTP servers. These are servers with directories that can be written to and read from. Hackers/crackers use these machines as way-points for transferring warez (pirated programs) and pr0n (intentionally misspelled word to avoid search engines classifying this document).
22	ssh pcAnywhere	<p>TCP connections to this port might indicate a search for ssh, which has a few exploitable features. Many versions using the RSAREF library can be exploited if they are configured in a certain fashion. (Suggestion: run ssh on some other port).</p> <p>Also note that the ssh package comes with a program called <code>make-ssh-known-hosts</code> that will scan a domain for ssh hosts. You will sometimes be scanned from innocent people running this utility.</p> <p>UDP (rather than TCP) packets directed at this port along with port 5632 indicate a scan for pcAnywhere. The number 5632 is (hex) 0x1600, which byte-swapped is 0x0016, which is 22 decimal.</p>
23	Telnet	The intruder is looking for a remote login to UNIX. Most of the time intruders scan for this port simply to find out more about what operating system is being used. In addition, if the intruder finds passwords using some other technique, they will try the passwords here.

25	SMTP	Spammers are looking for SMTP servers that allow them to "relay" spam. Since spammers keep getting their accounts shut down, they use dial-ups to connect to high bandwidth e-mail servers, and then send a single message to the relay with multiple addresses. The relay then forwards to all the victims. SMTP servers (esp. <code>sendmail</code>) are one of the favorite ways to break into systems because they must be exposed to the Internet as a whole and e-mail routing is complex (complexity + exposure = vulnerability).
53	DNS	DNS. Hackers/crackers may be attempting to do zone transfers (TCP), to spoof DNS (UDP), or even hide other traffic since port 53 is frequently neither filtered nor logged by firewalls. An important thing to note is that you will frequently see port 53 used as the <i>source</i> UDP port. Stateless firewalls frequently allow such traffic on the assumption that it is a response to a DNS query. Hackers are increasingly exploiting this to pierce firewalls .
67 and 68	bootp DHCP	Bootp/DHCP over UDP. Firewalls hooked to DSL and cable-modem lines see a ton of these sent to the broadcast address 255.255.255.255 . These machines are asking to for an address assignment from a DHCP server. You could probably hack into them by giving them such an assignment and specifying yourself as the local router, then execute a wide range of man-in-the-middle attacks. The client requests configuration on a broadcast to port 68 (bootps). The server broadcasts back the response to port 67 (bootpc). The response uses some type of broadcast because the client doesn't yet have an IP address that can be sent to.
69	TFTP	(over UDP). Many servers support this protocol in conjunction with BOOTP in order to download boot code to the system. However, they are frequently misconfigured to provide any file from the system, such as password files. They can also be used to write files to the system.
79	finger	Hackers are trying to: <ul style="list-style-type: none"> ○ discover user information ○ fingerprint the operating system ○ exploit known buffer-overflow bugs ○ bounce finger scans through your machine to other machines.
98	linuxconf	The utility " linuxconf " provide easy administration of Linux boxen. It includes a web-enabled interface at port 98 through an integrated HTTP server. It has had a number of security issues. Some versions are setuid root, trust the local network, create world-accessible files in /tmp, and a buffer overflow in the LANG environment variable. Also, because it contains an integrated web server, it may be vulnerable to many of the typical HTTP exploits (buffer overruns, directory traversal using <code>../..</code> , etc.).
109	POP2	POP2 is not nearly as popular as POP3 (see below), but many servers support both (for backwards compatibility). Many of the holes that can be exploited on POP3 can also be exploited via the POP2 port on the same server.
110	POP3	POP3 is used by clients accessing e-mail on their servers. POP3 services have many well-known vulnerabilities. At least 20 implementations are vulnerable to a buffer overflow in the username or password exchange (meaning that hackers can break in at this stage before really logging in). There are other buffer overflows that can be executed after successfully logging in.

- | | | |
|-----|--|--|
| 111 | sunrpc
portmap
rpcbind | <p>Sun RPC PortMapper/RPCBIND. Access to portmapper is the first step in scanning a system looking for all the RPC services enabled, such as rpc.mountd, NFS, rpc.statd, rpc.csmd, rpc.ttybd, amd, etc. If the intruder finds the appropriate service enabled, s/he will then run an exploit against the port where the service is running.</p> <p>Note that by putting a logging daemon, IDS, or sniffer on the wire, you can find out what programs the intruder is attempting to access in order to figure out exactly what is going on.</p> |
| 113 | identd
auth | <p>This is a protocol that runs on many machines that identifies the user of a TCP connection. In standard usage this reveals a LOT of information about a machine that hackers can exploit. However, it used by a lot of services by loggers, especially FTP, POP, IMAP, SMTP, and IRC servers. In general, if you have any clients accessing these services through a firewall, you will see incoming connection attempts on this port. Note that if you block this port, clients will perceive slow connections to e-mail servers on the other side of the firewall. Many firewalls support sending back a RST on the TCP connection as part of the blocking procedure, which will stop these slow connections.</p> |
| 119 | NNTP
news | <p>Network News Transfer Protocol, carries USENET traffic. This is the port used when you have a URL like news://comp.security.firewalls. Attempts on this port are usually by people hunting for open USENET servers. Most ISPs restrict access to their news servers to only their customers. Open news servers allow posting and reading from anybody, and are used to access newsgroups blocked by someone's ISP, to post anonymously, or to post spam.</p> <p>Update: @Home has started scanning their subscribers to see if they are running USENET servers. They are doing this in order to find these servers and close them before spammers can take advantage of them.</p> |
| 135 | loc-serv
MS RPC end-point
mapper | <p>Microsoft runs its DCE RPC end-point mapper for its DCOM services at this port.</p> <p>This has much the same functionality as port 111 for UNIX systems. Services that use DCOM and/or RPC register their location with the end-point mapper on the machine. When clients remotely connect to the machine, they query the end-point mapper to find out where the service is. Likewise, hackers can scan the machine on this port in order to find out such things as "is Exchange Server running on this machine, and which version?".</p> <p>This port is often hit in order to scan for services (for example, using the "epdump" utility), but this port may also be attacked directly. Currently, there are a few denial-of-service attacks that can be directed at this port.</p> |
| 137 | NetBIOS
name service
nbtstat | <p>(UDP) This is the most common item seen by firewall administrators and is perfectly normal. Please read the NetBIOS section below for more details.</p> |
| 139 | NetBIOS
File and Print
Sharing | <p>Incoming connections to this port are trying to reach NetBIOS/SMB, the protocols used for Windows "File and Print Sharing" as well as SAMBA. People sharing their hard disks on this port are probably the most common vulnerability on the Internet.</p> <p>Attempts on this port were common at the beginning of 1999, but tapered off near the end. Now at the start of year 2000, attempts on this port have picked up again. Several VBS (IE5 VisualBasic Scripting) worms have appeared that attempt to copy themselves on this port. Therefore, it may be worms attempting to propagate on this port.</p> |

143	IMAP4	<p>Same security idea as POP3 above, numerous IMAP servers have buffer overflows that allow compromise during the login. Note that for awhile, there was a Linux worm (admw0rm) that would spread by compromising port 143, so a lot of scans on this port are actually from innocent people who have already been compromised. IMAP exploits became popular when RedHat enabled the service by default on its distributions. In fact, this may have been the first widely scanned for exploit since the Morris Worm.</p> <p>This port is also used for IMAP2, but that version wasn't very popular.</p> <p>Several people have noted attacks from port 0 to port 143, which appears to be from some attack script.</p>
161	SNMP	<p>(UDP) A very common port that intruders probe for. SNMP allows for remote management of devices. All the configuration and performance information is stored in a database that can be retrieved or set via SNMP. Many managers mistakenly leave this available on the Internet. Crackers will first attempt to use the default passwords "public" and "private" to access the system; they may then attempt to "crack" the password by trying all combinations.</p> <p>SNMP packets may be mistakenly directed at your network. Windows machines running HP JetDirect remote management software uses SNMP, and misconfigured machines are frequent. HP OBJECT IDENTIFIERS will be seen in the packets. Newer versions of Win98 will use SNMP for name resolution; you will see packets broadcast on local subnets (cable modem, DSL) looking up sysName and other info.</p>
162	SNMP trap	Probably a misconfiguration.
177	xdmcp	Numerous hacks may allow access to an X-Window console; it needs port 6000 open as well in order to really succeed.
513	rwho	Probably from UNIX machines on your DSL/cable-modem segment broadcasting who is logged into their servers. These people are kindly giving you really interesting information that you can use to hack into their systems.
535	CORBA IIOP	(UDP) If you are on a cable-modem or DSL VLAN, then you may see broadcasts to this port. CORBA is an object-oriented remote procedure call (RPC) system. It is highly likely that when you see these broadcasts, you can use the information to hack back into the systems generating these broadcasts.
600	pcserver backdoor	<p>See port 1524 for more info.</p> <p><i>Some script kiddies feel they're contributing substantially to the exploit programs by making a minor change from <code>ingreslock</code> to <code>pcserver</code> in constant text... -- Alan J. Rosenthal.</i></p>
635	mountd	Linux mountd bug. This is a popular bug that people are scanning for. Most scans on this port are UDP-based, but they are increasingly TCP-based (mountd runs on both ports simultaneously). Note that mountd can run at any port (for which you must first do a portmap lookup at port 111), it's just that Linux defaulted to port 635 in much the same way that NFS universally runs at port 2049 .
1024	-----	Many people ask the question what this port is used for. The answer is that this is the first port number in the dynamic range of ports. Many applications don't care what port they use for a network connection, so they ask the operating system to assign the "next freely available port". In point of fact, they ask for port 0, but are assigned one starting with port 1024. This means the first application on your system that requests a dynamic port will be assigned port 1024. You can test this fact by booting your computer, then in one window open a Telnet session, and in another window run "netstat -a". You will see that the Telnet application has been assigned port 1024 for its end of the connection. As more applications request more and more dynamic ports, the operating system will assign increasingly higher port numbers. Again, you can watch this effect with 'netstat' as you browse the Internet with your web browser, as each web-page requires a new connection.

1025	-----	See port 1024 .
1026	-----	See port 1024 .
1027	-----	See port 1024 .
1080	SOCKS	This protocol tunnels traffic through firewalls, allowing many people behind the firewall access to the Internet through a single IP address. In theory, it should only tunnel inside traffic out towards the Internet. However, it is frequently misconfigured and allows hackers/crackers to tunnel their attacks inwards, or simply bounce through the system to other Internet machines, masking their attacks as if they were coming from you. WinGate, a popular Windows personal firewall, is frequently misconfigured this way. This is often seen when joining IRC chatrooms .
1114	SQL	This is rarely probed by itself, but is almost always seen as part of the sscan script.
1243	Sub-7	Trojan Horse (TCP). See the section on SubSeven for more details.
1524	ingreslock backdoor	Many attack scripts install a backdoor shell at this port (especially those against Sun systems via holes in sendmail and RPC services like statd, ttdbserver, and cmsd). If you've just installed your firewall and are seeing connection attempts on this port, then this may be the cause. Try telnetting to the attempted machine in order to see if it indeed comes up with a shell. Connections to port 600/pcserver also have this problem. [IN-99-04]
2049	NFS	The NFS program usually runs at this port. Normally, access to portmapper is needed to find which port this service runs on, but since most installations run NFS on this port, hackers/crackers can bypass portmapper and try this port directly.
3128	squid	This is the default port for the "squid" HTTP proxy. An attacker scanning for this port is likely searching for a proxy server they can use to surf the Internet anonymously. You may see scans for other proxies at the same time, such as at port 8000/8001/8080/8888. Another cause of scans at this port, for a similar reason, is when users enter chatrooms. Others users (or the servers themselves) will attempt to check this port to see if the user's machines supports proxying. See section 5.3 for more info.
5632	pcAnywhere	You may see lots of these, depending on the sort of segment you are on. When a user opens pcAnywhere, it scans the local Class C range looking for potential agents. Hackers/crackers also scan looking for open machines, so look at the source address to see which it is. Some scans for pcAnywhere frequently also include a UDP packet to port 22 . See dialup probes for more info.
6776	Sub7 artifact	This port is used separately from the SubSeven main port to transfer data. One example where you might see this is when a master is controlling a slave on a dialup line, then the slave machine hangs up. Therefore, when someone else dials-in at that IP address, they will see a continuous stream of connection attempts at this port. more on dialups
6970	RealAudio	Clients receive incoming audio streams from servers on UDP ports in the range 6970-7170. This is setup by the outgoing control connection on TCP port 7070.
13223	PowWow	The "PowWow" chat program from Tribal Voice. It allows users to open up private chat connections with each other on this port. The program is very aggressive at trying to establish the connection and will "camp" on the TCP port waiting for a response. This causes a connection attempt at regular intervals like a heartbeat. This can be seen by dial-up users who inherit IP addresses from somebody who was chatting with other people: it will appear as if many different people are probing that port. The protocol uses the letters "OPNG" as the first four bytes of its connection attempt. more

17027	Conducent	<p>Outbound: This is seen on outbound connections. It is caused by users inside the corporation who have installed shareware programs using the Conducent "adbot" wrapper. This wrapper shows advertisements to users of the shareware. A popular shareware program that uses this is PKware. Bill Royds mentions that in his experience, you can block this outbound connection with no problem, but if you block the IP addresses themselves, then the adbots can overload the link trying to reach the servers by continually connecting many times per second.</p> <p>The machines will attempt to resolve the DNS name "ads.conducent.com", which resolve to the IP addresses:</p> <p>216.33.210.40 216.33.199.77 216.33.199.80 216.33.199.81 216.33.210.41</p> <p>These addresses are hosted by Exodus.</p>
27374	Sub-7	Trojan Horse (TCP). See the section on SubSeven for more details.
30100	NetSphere	Trojan Horse (TCP). This is a commonly seen scan looking for systems compromised by this trojan.
31337	Back Orifice "elite"	This number means "elite" in hacker/cracker spelling (3=E, 1=L, 7=T). Lots of hacker/cracker backdoors run at this port, but the most important is Back Orifice. At one time, this was by far the most popular scan on the Internet. These days, it's popularity is waning and other remote access trojans are becoming popular.
31789	Hack-a-tack	UDP traffic on this port is currently being seen due to the "Hack-a-tack" RAT (Remote Access Trojan). This trojan includes a built-in scanner that scans from port 31790, so any packets FROM 31789 TO 317890 indicate a possible intrusion. (Port 31789 is the control connection; port 31790 is the file transfer connection).
32770 ~ 32900	RPC services	Sun Solaris puts most of its RPC services in this range. In particular, older versions of Solaris (pre-2.5.1) put a portmapper in this range, allowing hackers access to this even when low ports are blocked by a firewall. Probes in this range might either be for this portmapper, or for known RPC services that can be exploited.
33434 - 33600	traceroute	If you see a series of UDP packets within this port range (and only within this range), then it is probably indicative of traceroute. See traceroute for more info.
41508	Inoculan	Inoculan on UDP. Older versions of Inoculan apparently generate huge quantities of UDP traffic directed at subnets in order to discover each other. More info can be found at http://www.circlemud.org/~jelson/software/udpsend.html and http://www.ccd.bnl.gov/nss/tips/inoculan/index.html . Thanks to Jerry Leslie, NeoNET <leslie at clio dot rice dot edu>

1.2 What do the following source ports mean?

Ports 1-1024 are for reserved services, and almost never appear as the source. There are some exceptions, such as when connections come from NAT machines. See [section 1.9](#) for some more details.

Ports closely after 1024 (i.e. 1024-5000) are the ones most commonly seen. These are the "dynamic" range that are assigned to applications that don't care what port they use for their connection.

Server	Client	Service	Description
1-5/tcp	dynamic	FTP	Ports 1-5 are indicative of a script called ' sscan '
20/tcp	dynamic	FTP	FTP servers usually transfer files from this port.
53	dynamic	FTP	DNS servers will send UDP responses from this port. You may also see TCP connections with source/destination ports of 53.

123	dynamic S/NTP	The (Simple) Network Time Protocol (S/NTP) servers run at this port. They will also send broadcasts to this port.
27910-27961/udp	dynamic Quake games	Quake (and Quake-derived games) usually run servers at these ports. Therefore, UDP packet from this range (and to this range) will usually be games.
61000+	dynamic FTP	Ports above 61000 might come from machines behind a Linux NAT server called "IP Masquerade".

1.3 I'm seeing attempts on the same set of ports from widely varying sources all over the Internet.

This is due to a "decoy" scan, such as in 'nmap'. One of them is the attacker; the others are not.

Forensics and protocol analysis can be used to track down who this is. For example, if you ping each of the systems, you can match up the TTL fields in those responses with the connection attempts. This will at least point a finger at a decoy scan. (The TTLs should match; if not, then they are being spoofed). [Newer versions of scanner now randomize the attackers own TTL, making it harder to weed them out].

You can also attempt to go back further in your logs, looking for all the decoy addresses or people from the same subnets. You will often see that the attacker has actually connected to you recently, while the decoyed addresses haven't.

1.4 What are Trojan Horse probes?

The first stage of a Trojan Horse attack is to get the program on a user's machine. Typical techniques are:

- post the program to newsgroups claiming to be some other program
- spam mailing lists with the attached program
- post program to websites
- send via instant messenger programs and chat systems (ICQ, AIM, IRC, etc.)
- forge e-mail from the ISP (like AOL) with a hoax message asking somebody to run a program (such as a software update).
- copy to startup folder via "File and Print Sharing".

The next stage of the attack is to scan the Internet looking for machines that might be compromised. The problem is that most of the techniques outlined above don't tell the cracker/hacker where their victim machine is. Therefore, the cracker/hacker must scan the Internet looking for the machines they might have compromised.

This leads the condition where owners of firewalls (including personal firewalls) regularly see "probes" directed at their machines from crackers/hackers looking for these machines. However, if the machine hasn't been compromised, then these probes are not a problem. The probes cannot compromise the machine by themselves. Administrators can usually ignore these "attacks".

Typical ports used by these probes are listed below. In order to tell if your machine might be running one of these trojans, run the program "netstat -an" on your machine. Look for the ports that might be "listening" for incoming connections.

Port	Trojan
555	phAse zero
1243	Sub-7, SubSeven
3129	Masters Paradise
6670	DeepThroat
6711	Sub-7, SubSeven
6969	GateCrasher
21544	GirlFriend

12345 NetBus

23456 EvilFtp

27374 Sub-7, [SubSeven](#)

30100 NetSphere

31789 Hack'a'Tack

31337 BackOrifice, and many others

50505 Sockets de Troie

Resources: <http://www.commodon.com/threat/threat-ports.htm>

1.4.1 What is SubSeven (Sub-7) ?

Sub7 has become the most popular remote access trojan. At this time, it is the easiest-to-use and most powerful trojan. The reasons for this are:

- It is actively maintained/updated. Most other Trojans were created once then development stopped except for a couple of bug fixes.
- The program not only includes a scanner, but also can tell a slave machine to scan as well.
- The creator has a contest for cracked sites using Sub7.
- Supports "port redirection", so that any attack can be funneled through a victim's machines.
- Contains extensive tricks to play with ICQ, AOL IM, MSN Messenger, and Yahoo messenger, including password sniffing, posting messages, and other features.
- Extensive UI tricks, such as flipping the screen, talking through the victim's speaker, and spying on the victim's screen.

In short, it not only is an excellent hacking tool, the little "magic" tricks are designed to scare the <bleep> out of victims.

Sub7 is written by a hacker who calls himself "Mobman". His site can be reached at <http://subseven.slak.org/>.

Sub7 might use the following ports:

1243

The default connection port for older versions.

2772

Screen capture port

2773

Key logger port

6711

???

6776

I'm not sure what this port is for, but it has been claimed that this can serve as a "backdoor" in some versions. (Yes, a backdoor program with a backdoor to avoid password prompts).

7215

Port for the "matrix" chat program

27374

Another default port appearing in v2.0

54283

Spy port

1.9 DNS packets from low numbered ports

Q: I've seen many DNS requests from many low port numbers below 1024. Aren't they supposed to be reserved? Aren't they supposed to use 1024-65535 range?

A: These are coming from machines behind NAT firewalls. A NAT doesn't necessarily have the concept of reserved port numbers. *thanks to Ryan Russell Ryan.Russell at sybase dot com*

Q: My filters reject incoming packets with source ports below 1024, so the DNS lookups are failing.

A: Don't filter that way. Lots of firewalls have similar rules, but this is somewhat "misguided" since hackers/crackers can forge whatever ports they want.

Q: Are these NAT firewalls doing it incorrectly?

A: Not in theory, but in practice it will result in failures. The "correct" way would be more strictly control DNS traffic in any case (such as essentially "proxying" DNS and forcing out through port 53).

Q: I thought DNS lookup was supposed to use a random source port above 1024?

A: In practice, your average DNS client will use a non-reserved port. However, a lot of implementations use a source port of 53. In any case, the NAT issue is completely separate because it completely changes the entire 'socket' (IP address + port combo).

1.10 Immediately upon dialing up to my ISP, my personal firewall starts alarming me about probes against port X.

This is very common. The cause is that somebody hung up just before you dialed in and your ISP assigned you the same IP address. You are now seeing the remnants of communication with the previous person.

A typical example is [chat](#) programs. If someone simply hangs up, then everyone who was chatting with that person will attempt to still send traffic to them. Some programs take a long time to timeout. Typical programs that show this behavior are PowWow and ICQ.

Another example is on-line, multiple games. You might see such traffic from gaming providers like MPlayer, or maybe from unknown servers (Quake servers litter the Internet). These games are typically UDP based, so there is no concept of a connection that can be dropped. They also are quite aggressive at maintaining connections, in order to make a good user experience. Some game ports that you might see are:

7777	Unreal, Klingon Honor Guard
7778	Unreal Tournament
22450	Sin
26000	Quake
26900	Hexen 2
26950	HexenWorld
27015	Half-life, Team Fortress Classic (TFC)
27500	QuakeWorld
27910	Quake 2
28000-28008	Starsiege TRIBES (TRIBES.DYNAMIX.COM)
28910	Heretic 2

Another example is multimedia audio/visual. For example, [RealAudio](#) uses UDP ports in the range of **6970-7170** for clients to receive audio streams.

Make sure that you carefully figure out the correct side of the connection. For example, an ICQ server runs on port 4000, and the client chooses a random high-numbered port. That means you will see UDP packets from port 4000 going to the random port. In other words, don't go looking in a port database trying to figure what that random, high-numbered port means. The significant port is the source.

The [SubSeven](#) trojan has a similar problem. It uses separate TCP connections for different services. If the slave agent goes away, it will continue to create connection attempts to the slave ports, especially at port [6776](#).

1.11 IRC servers are probing me.

One of the most popular applications is "chat", like IRC. One feature of chat programs is that they reveal the IP address of the people you are chatting with. One problem with chatrooms is that people enter the rooms "anonymously" and play around, either by disrupting conversations with offtopic comments and flamebait, or by "flooding" the servers or other clients in an attempt to kicked them off.

Therefore, both servers and clients are implementing measures to stop "anonymous" use of chatrooms. In particular, they check people entering chatrooms in order to see if they are "proxying" through some other connection. The most popular of such probes is SOCKS. The assumption is that if the IP address of where you are coming from supports SOCKS, then it is possible that you have a completely separate machine and are only going through the indicated machine in order to hide your true identity. Undernet's policy on this can be found at <http://help.undernet.org/proxyscan>.

At the same time, crackers/hackers will scan people's machines in order to determine if they are running some sort of server that can be bounced through. Again, by checking for SOCKS, the attacker hopes to find somebody that has left SOCKS open, such as a home user implementing connection sharing using SOCKS, but accidentally configured it so that anybody on the Internet has access to it.

1.12 What are "remapped" ports?

A common technique is to remap ports to some other address. For example, whereas the default port for HTTP is 80, many people remap it to another port, such as 8080 (hence, this document could reside at <http://www.robertgraham.com:8080/pubs/firewall-seen.html> if I were to remap the port).

Remapping is done under the theory that making the port harder to find will make it more difficult for a hacker to exploit. Instead of simply exploiting a well-known service at a well-known port, the hacker will have to port scan the machine.

Most port remapping is done at some variation of the original port. Therefore, most HTTP ports are based upon a variation of the theme "[80](#)": 81, 88, 8000, 8080, 8888, and so forth. POP, which is originally at port [110](#) can often be found at port 1100.

There are other statistically significant chosen numbers, like 12345, 23456, 34567, etc. Many people also choose numbers that are well known for other reasons; 42, 69, 666, 31337, and so on. The recent proliferation of Remote Access Trojans (RATs) has resulted in hackers/crackers choosing the same defaults for their programs. For example, NetBus defaults to port 12345.

Blake R. Swopes points out that remapping is also done because on UNIX machines, your server needs root privileges to listen on ports below 1024. If you don't have root level access and want to run a web service, you will need to install it on a high-numbered port. Likewise, some ISPs might firewall low-numbered ports, forcing you to remap even when you own the entire machine.

1.13 I still can't figure out what somebody is trying to connect to a port, what can I do?

Use netcat in order to setup a listening process. For port '1234', use:

```
netcat -L -p 1234
```

A lot of protocols will send data as the first part of the connection. By setting up netcat listening on the port, you might be able to figure out what protocol that are using. If you are lucky, the protocol in question will be HTTP, which will give you a wealth of information that you can use to track down what is happening.

The "-L" option means to listen continuously. Normally, netcat would accept a single connection, dump the contents, then exit. By adding this option, it will remain running for multiple connections.

2. ICMP

Whereas TCP and UDP carry data, ICMP contains purely *control messages*. Therefore, ICMP messages cannot really be used to break into your machine. Hackers use ICMP messages to attempt to scan networks, [DoS](#) machines, or redirect traffic.

Some firewalls incorrectly label ICMP fields as "ports". ICMP has no ports like TCP or UDP, but it does have two fields called "type" and "code". While these fields serve completely unrelated purposes, the fact that there are two of them have led to firewalls mislabeling them. For more on ICMP, please read my Infosec Lexicon entry on [ICMP](#).

The official reference for what ICMP Type/Code fields mean is found at <http://www.isi.edu/in-notes/iana/assignments/icmp-parameters>. While that document describes the official meanings, this section describes what hackers are trying to do. This section contains a brief summary at top, then more details descriptions down below.

Type	Code	Name	Summary
0	*	Echo Reply	A response to a ping. [more]
3	*	Destination Unreachable	An indication back from a host or router that some packet did not reach its destination. [more]
	0	Net Unreachable	Route configuration problem or incorrectly specified IP address. [more]
	1	Host Unreachable	It means that the router one hop before the desired host could not ARP the host.
	3	Port unreachable	The server tells the client that nobody is listening at the port the client attempted to contact. [more]
	4	Fragmentation Needed but DF set	Important: If you are seeing these in your firewall reject logs, then you've misconfigured your firewall. You should allow this packet to pass through, otherwise your clients will see their TCP connections mysteriously hang. [more]
4	*	Source Quench	Congestion on the Internet. [more]
5	*	Redirect	Somebody is trying to redirect your default router. This could be from a hacker trying to execute a man-in-the-middle against you by causing you to route through their own machine.
8	*	Echo Request	Ping. [more]
9	*	Router Advertisement	There is exists a hack against Win9x and Solaris such that a hacker can DoS you by redirecting your default router. A neighboring hacker can also do a man-in-the-middle attack by directing you through his/her router.
11	*	Time Exceeded In Transit	It means that a packet never reached its target because something timed out.
	0	TTL Exceeded	Router dropped the packet either because of a <i>routing loop</i> or maybe because of a traceroute. [more]

1	Fragment reassembly timeout	The host dropped the packet because it didn't receive all the fragments. [more]
12	* Parameter Problem	Something unusual is going on, and probably indicates an attack. [more]

2.0 Type = 0 (Echo Reply)

The sender is responding to a ping from your address. This could be because:

Someone's ping that person

Somebody behind the firewall is sending pings to the target.

Automated ping

Lots of applications use pings for various purposes, such as to see if their communication partner is alive, or to measure the response time. A big cause of this is VitalSign's Net.Medic, which sends pings of various sizes in order to measure link speed.

Decoy Ping Sweep

Somebody is using your IP address as a decoy in a ping sweep, so you are seeing the responses.

Covert-channel communications

Most network managers block incoming pings (type=8), but allow ping responses (type=0). Therefore, hackers have begun using ping replies as ways of bypassing firewalls. For example, in the massive DDoS attacks against Internet sites, commands could be imbedded in ping responses, and floods of responses were directed against the sites in order to clog their Internet connections.

2.3 Type = 3 (Destination Unreachable)

The exact code is important in the Unreachable packet.

Note that Unreachables sometimes play a part in defeating SYN floods. This means that if a host you are talking to is under SYN flood attack, you will not be able to reach them if you block incoming Unreachables.

In some cases, you will receive destination unreachable packets from hosts you have never heard of. The most common cause of this is a "decoy scan". An attacker is sending spoofed packets a target using possibly hundreds of source addresses, including one that is the real address. The hacker's theory is that the victim won't wade through all the decoys in order to pin them down.

The best way to solve this is to examine the actual packets as described [below](#). Try to discover if the pattern looks like what one would see in a decoy scan. For example, look for alternating port numbers in TCP or UDP headers contained within the ICMP portion of the packet.

2.3.0 Type = 3, Code = 0 (Destination Net Unreachable)

No route to host A router tells the client that it does not know how to route to anything at all in the network range that includes the host the client is talking to. This indicates either the client chose the wrong IP address, or that routing tables are misconfigured somewhere. Note that sometimes you see the message "No route to host" on your own UNIX machine when your own routing tables are messed up, which is especially common when configuring point-to-point links.

2.3.3 Type = 3, Code = 3 (Destination Port Unreachable)

This packet is sent by a SERVER when a CLIENT tries to connect to a UDP port that isn't running. For example, if you try to send an SNMP packet to port 161, but the machine doesn't support the SNMP service, you will get back an ICMP Destination Port Unreachable packet.

Protocol Decode

The first thing to debug this problem is to check the port numbers within the packet. You probably need to use a [sniffing](#) utility as firewalls tend not to log the information. This technique relies upon the fact that ICMP messages include the IP and UDP headers of the original packet. Here is a hex dump of an ICMP unreachable:

```
00 00 BA 5E BA 11 00 60 97 07 C0 FF 08 00 45 00
00 38 6F DF 00 00 80 01 B4 12 0A 00 01 0B 0A 00
01 C9 03 03 C2 D2 00 00 00 00 45 00 00 47 07 F0
00 00 80 11 1B E3 0A 00 01 C9 0A 00 01 0B 08 A7
79 19 00 33 B8 36
```

Where the bytes **03 03** are the type/code for the ICMP packet. The last 8 bytes of the packet are the original UDP header, which decodes as:

08A7

UDP Source Port = 2215

May be dynamically allocated, so no always important.

7919

UDP Destination Port = 31001

This is very important, it meant the person was originally attempting to contact a service on port 31001.

0033

UDP Length = 51

The length of the original UDP data might be important.

B836

UDP Checksum = 0xB836

The checksum may or may not be important

Analysis

Here are some reasons why you may be seeing this:

Decoy UDP Scans

Somebody may be scanning the person who sent you the ICMP packet. They are forging the source as one of your IP addresses. They will in reality forge lots of different source addresses so that they victim can't be sure who it really is. If you receive large numbers of these packets from the same source in a short time frame, then this is a likely bet. Check the *UDP Destination Port* field. If it is constantly changing, then this is a very likely scenario.

Stale DNS

A client may send a DNS request to your server, which takes a long time to resolve. By the time your DNS server responds, the client has already forgotten about you and closed the UDP port assigned to receive your response. Check the *UDP Source Port* field to see if it equals 53. If so, then this is a likely occurrence. Why does this happen? The server may be resolving a recursive query, but its own query packet was lost, so it had to time out and try again. By the time it gets back to the client, it has timed out. Many client applications (especially on Windows) do their own DNS resolution, meaning that they must create their own socket to do so. If they passed the request onto the OS, it is likely the OS would simply have left the socket open.

Multi-response DNS

Another variation is when the client receives multiple responses to the same request. It receives the first response, then closes the socket. Subsequent responses will be dropped. There other variations on this problem. A Sun machine connected with multiple NICs on the same Ethernet will assign both NICs the same MAC address, causing it to receive two copies of every frame, then send multiple responses. Likewise, a poorly written client program (it has been claimed that some DNS resolvers are multi-threaded, but not thread safe) sometimes send out multiple requests, then close the socket on the first response. However, there may be an attempt at **DNS spoofing**, where a hacker is attempting to corrupt the resolver's cache by sending both a recursive query and a response.

NetBIOS Resolution

If the receiver of the ICMP packets is a Windows machine, look to see if the *UDP Destination Port* is 137. In this case, the cause of this is the Windows system trying to execute the 'gethostbyaddr()' function, which attempts to resolve the IP address into a name using both DNS and NetBIOS. The DNS request gets sent to a DNS server somewhere (and not sent to the target), but the NetBIOS request gets sent directly to the target. If the target doesn't support NetBIOS, then it will send back an ICMP unreachable.

Traceroute

Most traceroute programs (with the exception of Windows tracert.exe) send UDP packets to closed ports. This causes a sequence of back-to-back ICMP Port Unreachable packets to be sent back to the machine doing the traceroute. Thus, if you are seeing these ICMP packets on your firewall, then somebody inside might be doing a traceroute. You may also see TTL exceeded as well.

2.3.4 Type = 3, Code = 4 (Fragmentation Needed and Don't Fragment was Set)

These are sent by routers attempting to forward IP datagrams that are marked "DF" (Don't Fragment).

Why? Both IP and TCP fragment data, but in different ways. TCP is vastly more efficient at fragmentation than IP. Therefore, stacks attempt to find the "Path MTU (Maximum Transmission Unit)". This ICMP message is sent during that process.

Let's consider ALICE talking to BOB. Both are on Ethernets (max frame size = 1500 bytes), but some intervening link limits the maximum IP packet size to 600 bytes. This means all IP packets sent will be fragmented by the routers on that link into 3 fragments. Since it is much more efficient to fragment at the TCP layer, the TCP stack will attempt to discover the MTU. It does this by setting the "DF" (Don't Fragment) bit in all its packets. As soon as it hits a router that cannot forward a packet that large, the router will send back this ICMP error message. From that, the TCP stack will know how to fragment correctly.

You should probably let these packets through the firewall. Otherwise, the intended recipient will have a hung connection as small packets get through to set up the connection, but the large packets are mysteriously dropped. A common result from this are people who see web pages that are only halfway returned.

Path MTU Discovery is becoming more and more integrated into communication. For example, IPsec needs this functionality.

2.4 Type = 4 (Source Quench)

These packets are supposed to be transmitted by routers/destination when traffic level exceeds a certain threshold. Many systems today, however, do not generate them. The reason is that we now believe that simple packet loss is the best indication of congestions (since the only reason packets are dropped, in practice, is congestion).

In general, the rules for source quenches are now ([RFC 1122](#)):

- Routers SHOULD NOT generate them.
- Hosts MAY generate them.
- Hosts SHOULD honor them.
- Firewalls SHOULD discard them.

However, hosts still react to Source Quenches by slowing communication, so they can be used as a denial of service. Firewalls should filter these out. If a DoS is suspected, the source address of the packets will be meaningless, because the IP addresses are spoofed.

Source quenches have been known to be sent by some SMTP servers.

2.8 Type = 8 (Echo aka PING)

These are ping request packets. They are used all over the place; it may indicate hostile intent of someone trying to scan your computer, but it may be part of the normal network functionality. See Type = 0 (Echo Response) above for more info.

Lots of network management "scanners" will precede a scan using a special ping packet. These include ISS scanner, WhatsUp monitor, and others. This will be visible in the payload of the scanner. Most firewalls don't log this payload, so you may need to use some sort of [sniffer](#) to capture them or some time of [Intrusion Detection System](#) to flag them.

Note that blocking incoming PINGs does not mean a hacker can't scan the network. There are many other ways of doing this. For example, TCP ACK scanning becoming popular -- they usually get through the firewall, and they illicit a response from the target system.

Pings sent to broadcast IP addresses like `x.x.x.0` or `x.x.x.255` are probably attempts to use your network as a [smurf](#) amplifier.

2.11 Type = 11 (Time Exceeded In Transit)

This probably doesn't indicate an attack from a hacker/cracker.

2.11.0 Type = 11, Code = 0 (TTL Exceeded In Transit)

This can be caused by a number of things. If somebody from your site is doing traceroutes out to the Internet, you will see lots of TTL exceeded responses from routers. This is how traceroute works: forces the routers to generate TTL exceeded messages in order to find them.

Another common reason firewall administrators see this is due to routing loops developing in the Internet. Route flapping (constant route changes) is a common problem, and will often briefly result in a loop. This means that while a IP packet is heading towards it destination, the packet gets misrouted to a router that it previously visited it. The packet then gets routed in a circle infinitely -- or it would be, if the routers didn't decrement the TTL field each time and discard the packet once that value hit zero.

Another cause of this is distance. Many machines start with a default TTL of 127 (Windows) or even lower. Routers will often decrement the TTL more than by one in order to reflect slow lines like dialups or transcontinental links. Therefore, a site might not be reachable with a low initial TTL. In addition, some hackers/crackers like to make their site unreachable through this method.

2.11.1 Type = 11, Code = 1 (Fragment Reassembly Time Exceeded)

When sending fragmented IP datagrams, the sender of this message never received all the fragments. Normally, most TCP/IP traffic shouldn't even be fragmented. You will only see this if the traffic is both fragmented AND there congestion somewhere between you and the target.

2.12 Type = 12 (Parameter Problem)

This probably indicates an attack. There are a number of [fingerprinting](#) techniques that will generate these packets.

3. IP

3.1 What are source routed packets?

Source route is an option in the IP header that allows the sender to override some or all of the routing decisions. Normally, routers between the source and destination decide how to route the packet.

There are a couple of network management uses of this packet, such as testing to see if two computers can talk to each other. A network manager at point A may send a packet to B through point C. This tells A if B & C can talk to each other.

The same technique can be used to evade firewalls, subvert trust relationships, and communicate with machines using "private" address (10.x.x.x, 192.168.x.x, 172.[16-31].x.x).

Let's say you are a hacker/cracker on the Internet and you want to talk to some machines behind a firewall who use 10.x.x.x as their IP addresses. Since the routers on the Internet do not know where this subnet is located, they will drop your packets. However, you put a loose source route option in the IP packet and tell all the Internet routers to first forward to the firewall. Since the firewall straddles both the Internet and the private network, it will know how to forward the packet appropriately. Thus, you can carry on a conversation with the victim by bouncing all packets through the firewall.

This can be used with IP spoofing. You pretend to be a router (like the firewall mentioned above) and pretend that somebody else is bouncing packets through you. Thus, pick some random machine on the Internet (ALICE) as the spoofer, then send packets from ALICE to your victim BOB. BOB will think the packets are coming from ALICE, but in reality they are coming from you. This masks the real source of the attack.

This is even better if you know that BOB trusts ALICE. IP addresses are often used as part of authentication. Let's say the firewall has a rule allowing all traffic from ALICE into the network. By forging all IP packets to be from ALICE (but being source routed through your own machine), then you get free access to the victim network.

More and more core Internet routers are disabling source routed packets. They slow down routing anyway, but they are a huge security risk. There is also no real need for them. Managers should do the same and disable source routing everywhere: on firewalls, on routers, and even on end-nodes so that they won't even accept incoming source routed packets.

See Microsoft Knowledge Base article Q217336 for setting the "DisableIPSourceRouting" on WinNT SP5 systems

3.2 I'm seeing the IP address 255.255.255.255 in my reject log

This is happening a lot these days as more and more people use DSL or cable-modem connections. The reason is that unlike point-to-point connections (like T-1, frame relay, etc.), these new high-speed technologies drop you onto an ATM VLAN, which is a single broadcast domains. In fact, many cable-modem users are seeing multiple megabytes of traffic per day simply from such broadcasts.

You must remember that such packets MUST be local. Routers (generally) refuse to forward packets with the IP address of 255.255.255.255. This address is known as a "local broadcast" for this reason: it never travels past the local segment (or these days, the local "virtual" segment).

What are these packets for?

Check the list of ports at the top of this document. If it is not listed there, then the only way to figure this out is to capture them with a [sniffer](#) and view their contents.

For example, a common service that runs with a random port number is CORBA IIOP packets. Many services run at [port 535](#), but it is frequently reconfigured to broadcast on other ports. If you look at the hex dump in the sniffer, you will see the letters "IIOP" somewhere in the contents.

In any case, this is rarely something to be concerned about. In fact, it advertises something about the person sending the traffic that can be used to hack them. Hackers rarely attack their own neighborhoods (because it is easy to detect), so it probably is accidental, not malicious.

It should be noted that with today's ATM networks, the source of the broadcast may not even be in the same state as you are; they may be hundreds of miles away. The word "local" means in terms of the network topology, not distance.

3.3 How do I track down the owner of an IP address?

Remember that IP addresses can be spoofed, so that the "owner" of an IP address may be innocent. Increasingly, attacks are coming from compromised machines. The owner of the IP may actually be grateful! Both of these statements come to the same conclusion: be polite and professional.

Many companies have established the e-mail address "abuse@example.com" (replace "example" with the proper company). This e-mail role is for both e-mail abuse (such as spam) as well as for network abuse. When you find the owner of the IP address, you should probably compose a message including the evidence of the attack.

Registrar Databases

In the past, all the IP address owners were kept by the Internic. A database built from that information is at <http://ipindex.dragonstar.net/>. There are now 3 official registrars for North America, Asia, and Europe. Unfortunately, you will have to query each individual database. However, if you start with the North America registrar, it will tell you if the address belongs to one of the other three. **Warning:** The returned information is fragile; so don't send flames to these people because you have only about 30% chance of reaching the right people.

America	http://www.arin.net/whois/ ARIN (American Registry for Internet Numbers)
Europe	http://www.ripe.net/db/whois.html RIPE (Reseaux IP Europeens)
Asia and Pacific	http://www.apnic.net/apnic-bin/whois.pl APNIC (Asia Pacific Network Information Centre) [more]

Japan	http://www.nic.ad.jp/cgi-bin/whois_gw JPNIC	Japanese	English
--------------	--	----------	---------

[traceroute](#)

Running traceroute will often find at least the ISP who is hosting the IP address. A reverse DNS lookup on the actual IP address is easy to spoof, but the route to the machine will reveal who is hosting the possible intruder.

Common IP addresses

Many attacks are now coming from cable-modem subscribers in the 24.x.x.x range. These are probably from machines who have been compromised by a Remote Access Trojan (RAT). (While hackers/crackers frequently use dial-up lines because they don't care if their account gets canceled, few users want to have their cable-modem accounts canceled).

Another important range is the "private address" ranges of 10.x.x.x, 192.168.x.x, and 172.16.x.x-172.31.x.x. See [3.4](#) below.

Addresses like 127.x.x.x indicate "localhost" and should never be seen on the Internet.

The address range 192.0.2.x has been designated for "examples", like "example.com".

3.4 I'm seeing packets from "private" addresses (10.x.x.x et al.) on the Internet side of my firewall

The "private address" ranges are 10.x.x.x, 192.168.x.x, and 172.16.x.x-172.31.x.x.

I've been seeing these in three cases:

traceroutes

Core routers on the Internet are increasingly being assigned IP address in this range. There is really no need for a router to be reached from the Internet. The "forwarding" function really is independent from "sending/receiving" packets. When a router drops a packet and sends back a "ICMP TTL Exceeded" message, it will use the private address. Note that some routers are multi-homed with both private and non-private addresses. Other routers have only private addresses.

cable-modem, DSL

Most cable-modem and DSL connections are on virtual LANs over ATM. You will often see broadcast packets from neighboring machines with these private addresses.

hackers

Very rarely, you may see an address from a hackers who are spoofing addresses in this range.

3.5 What kind of scans should I expect to see from quasi-legitimate sources?

You will often see scans from somewhat legitimate sources. What I mean by this is that people will scan you without the intention of actually hacking you. For example, search engines will index your site, but it isn't an attack.

[Doubleclick](#)

Sends [echos](#) to people in order to redirect them to a nearer server for their advertising.

<http://www.cyveillance.com/>

Scans websites looking for illegal activities, such as copyrighted items.

3.6 I'm seeing source IP address of 0.0.0.0?

If the port is also 0, then this is probably an attempt to [fingerprint](#) your system.

3.7 What are directed broadcasts and what do they mean?

TODO:

- Often indicate people scanning your subnet
- Hackers looking for smurf amplifiers

3.8 I'm seeing strange addresses like 169.254.x.x?

From a [draft](#) document on auto-configuration of IP addresses when DHCP fails:

```
Once a DHCP Client has determined it must auto-configure an IP
address, it chooses an address.  The algorithm for choosing an
address is implementation dependant.  The address range to use MUST
be "169.254/16", which is registered with the IANA as the LINKLOCAL
net.
```

This only happens when the normal DHCP process fails.

This new technique was introduced with Microsoft Win98 and Apple MacOS 8.5.

Also see: <http://www.unixreview.com/archives/articles/1999/july/9907dd.shtml>

4. Stuff doesn't work

4.1 Installing a firewall causes slow connections to POP and SMTP services

This is because the POP and SMTP servers are trying to establish an [identd/AUTH](#) connection back to the client. These reverse-connections are blocked, and it takes a while before the servers timeout and continue.

The identd/AUTH service identifies the user of the TCP connection (user name, process id, etc.). When the e-mail server accepts the incoming TCP connection, before sending the greetings, it will first attempt to gather information via the identd protocol. This consists of a TCP connection in the reverse direction. In other words, when I connect to my e-mail server, my e-mail server attempts to connect back to me on port 113, the identd port. My e-mail connection just sits there until the e-mail server resolves the identd information.

The problem comes about because the firewall silently drops the SYN packet. The e-mail server is expecting an immediate SYN-ACK (identd supported) or RST (identd not supported), but when the firewall drops the packet it keeps trying until the connection times out.

Note that the e-mail server doesn't care if I don't support identd, and indeed most people don't on their clients. It just wants an immediate response one way or the other. The firewall blocks that. This is why some personal firewalls for Windows (like BlackICE Defender from my company) contain default rules that allow identd/AUTH to pass through. Windows doesn't reveal the information that UNIX does, and opening it up gives the immediate response these servers are looking for.

To solve this problem:

- reconfigure the e-mail server to stop querying identd info
- reconfigure the firewall to RST all those connections
- reconfigure the firewall to allow this protocol, but this would be a **BAD IDEA** because identd/AUTH reveals a HUGE amount of information about your UNIX machines.

Note that this means you should be seeing lots of dropped incoming connection attempts at [port 113](#) in your log files because of this.

5. What are some typical signatures of well-known programs?

5.1 traceroute

The program "traceroute" is based upon a very intelligent hack by Van Jacobson (also famous for other nifty kludges). Every IP packet has a **time-to-live (TTL)** field that indicates how many hops the packet can travel before being dropped. This field is needed because routers sometimes get misconfigured and will forward packets in a continuous: i.e. Alice forwards the packet to Bob who forwards it to Charlene who mistakenly forwards it back to Alice.

Therefore, each router decrements (subtracts 1) from the TTL field. When each reaches zero, the router who currently has the packet will simply "drop" it (not forward it on). When a router drops a packet, it sends a message back to the sender informing for this. This message is called an ICMP "TLL Exceeded in Transit".

The nifty thing about this is that the router uses its own IP address as the source address of the ICMP message. Therefore, if you send a packet to a target but with a TTL of only 1, the first router will receive the packet, decrement the field to 0, drop it, then send back the ICMP notification. This informs you of the first router along the route (which you probably knew anyway).

The same goes for an initial TTL of 2. The first router gets it, decrements to 1, then forwards to the second router along the route. This router then decrements to 0, drops the packet, and sends back an error ICMP message.

By continuing this process, you eventually end up with the list of routers between yourself and the target.

Versions of traceroute

There are various versions of the traceroute program. In particular, the Windows program "tracert.exe" uses pings as the packet it sends to the target. Therefore, you might see ICMP Echoes on your firewall.

The most popular "traceroute" program for UNIX programs sends UDP datagrams to port [33434](#) for the first packet sent, then increases this port number by one for each successive packet. This means that you will never see port 33434 on your firewall, but you will start to see successive ones starting at higher port numbers. Traceroute programs typically send 3 packets for each hop (in case some get dropped). Therefore, if somebody is 10 hops away, the first port you will see is $33434 + 3 * 10 = 33464$.

Symptoms

Firewall administrators should learn the symptoms of traceroute activity.

port scans in [33434-33600](#)

A brief sequential "port scan" in this range usually indicates a traceroute for a UNIX machine, as explained in this section.

[incoming TTL exceeded](#)

If someone inside the network is attempting a traceroute, then you'll see these incoming packets. Many admins allow these through the firewall.

outgoing TTL exceeded

This indicates that somebody is tracerouting you. This doesn't necessarily indicate hostile activity, but somebody is scanning you. These should be blocked by the firewall.

[outgoing ICMP port unreachable](#)

When a traceroute successfully hits a target, it will generate back-to-back "ICMP port unreachable" messages (probably 3 in a row).

Other

Some traceroutes are designed to bypass firewalls. See

<http://www.packetfactory.net/Projects/Firewalk/firewalk-final.html> for more information.

5.2 sscan

The 'sscan' tool has become a popular scanning tool on the Internet. It not only "port scans" but attempts to discover some common vulnerabilities. There are several versions of sscan, and it is very configurable, so matching an exact signature to this program may be difficult. The 'sscan' program is derived from the older 'mscan' tool.

A sscan goes through several phases:

TCP ACK pings

The program will attempt to see if the host is reachable by scanning for the most common services, namely ports [23/telnet](#), [25/smtp](#), [110/pop3](#), [143/imap4](#), [80/http](#). This phase is easily detected because both the source and destination port are the same.

connection attempts

Connection attempts are made to several services in order to see if they are available. This is highly configurable. Typically configured probes are those above, as well as [111/rpc](#), [6000/x-windows](#), [79/finger](#), [53/dns](#), [31337/elite](#), [139/netbios,smb](#), [21/ftp](#), [1114/msql](#), [1/tcpmux](#)

OS [fingerprint](#)

sscan contains a basic OS fingerprinting technique, easily detected because it uses source ports 1-5. The fingerprinting is not as complete as the techniques used by Queso or nmap.

vulnerability assessment

It then looks at the ports that are open and checks the banners that might indicate a vulnerable version of one of the services. It also scans for a range of known vulnerable CGI scripts.

script execution

Depending upon what it finds, it can further launch configured scripts against the system.

Example

The following is a record pulled from an intrusion detection system.

```
ports=1 22 23 25 53 79 110 111 143 1114 2766 6000 31337
```

Unfortunately, the system consolidates alerts, discards duplicates, and keeps the port numbers in sort order. In a real scan, several of the ports would have duplicate connection attempts, and port 1/tcpmux would be one of the last probes, not one of the first.

More info

[\[IN-99-01\]](#)

5.3 Proxy scanners

One of the most common scans on the Internet looks for HTTP proxy servers. Normally, the hackers aren't looking to compromise systems, they simply want the ability to "anonymize" their connections. For example, most anonymous e-mail services ([HotMail](#), [Yahoo mail](#), etc.) will store the IP address in the e-mail headers, making them not so anonymous (many people have been caught this way). By bouncing HTTP traffic through a proxy server, the hacker can complete erase his/her tracks.

In late summer of 1999, probes for ports 80/8080/3128 were particularly noticed. These came from all over the Internet and were fairly disjoint. These came from a Trojan Horse called "Ring0" (RingZero). It would infect PCs, then scan random IP addresses for proxy servers. The SANS Institute (a security training/conference organization) coordinated an effort to track down exactly what was happening from reports from many of their customers. A common symptom of this Trojan is 3 probes spaced within a minute from the same IP address from this Trojan. More information can be found at: <http://www.sans.org/newlook/resources/ringzero.htm>. A news article by CMP can be found at: <http://www.techweb.com/wire/story/TWB19991013S0018>

A list of open proxies can be found at: <http://freebooks.hypermart.net/proxy/proxies.htm>

Ports with variations of the "80" them (81, 88, 8000, 8080, 8888, etc) are most commonly used for proxies. In addition, a popular free proxy server called "squid" runs at port 3128.

5.4 smurf/fraggle

Smurf/fraggle programs send packets to broadcast addresses with a [spoofed](#) source address of the victim. Everybody on that subnet then sends responses back to that address, flooding it.

A **smurf** is a ping ([ICMP Echo Request](#)) whereas a **fraggle** is a [UDP port 7/echo](#). These are named after the programs/scripts that first implemented them.

These packets are sent to *broadcast* addresses. In IP, a *directed broadcast* has all the "host" bits set to either one or zero. This means an address that looks something like 192.0.2.0 or 192.0.2.255 is likely a broadcast. The key thing to remember is that such addresses are only broadcasts if the router on that subnet chooses to interpret it as a broadcast. If that router has this configured as a broadcast in its routing tables, it will forward the single IP packet as broadcast on that (Ethernet) segment, causing all systems on that (Ethernet) segment to receive the packet.

Therefore, there are two configuration problems:

- Routers forwarding directed broadcasts.
- Systems responding to broadcasts.

Both can be fixed.

5.4.1 fraggle signature

Somebody saw the following incident with millions of incoming packets. Below are some examples of these packets:

source	destination	sport	dport	protocol
212.187.65.86	192.0.3.63	7744	7	17
212.187.65.86	192.0.2.128	6537	7	17
212.187.65.86	192.0.2.63	29432	7	17
212.187.65.86	192.0.2.128	15793	7	17
212.187.65.86	192.0.2.191	17367	7	17
212.187.65.86	192.0.3.63	29210	7	17
212.187.65.86	192.0.3.127	351	7	17
212.187.65.86	192.0.2.127	17330	7	17

Some questions that have been asked about this are:

Q: Why are these only aimed at strategic points like broadcast addresses?

A: Because if a single packet is sent to a broadcast, then it generates lots of responses to the spoofed address of the victim.

Q: I monitor multiple networks. Why is only this network being attacked this way?

A: Your network isn't being attacked; instead it is the third party in a fraggle attack. Your network is being used to attack somebody else (the source address of the packets, which is spoofed). Either your other networks aren't nearly as effective as fraggle amplifiers, or they have been registered in smurf/fraggle registries yet. Hackers rarely look for their own amplifiers, but instead simply look up good amplifiers in such directories. If you get registered, then multiple hackers will use/abuse your network.

Q: Why port UDP [7](#) only?

A: There are a number of reasons. The first is that script-kiddies aren't too bright. If they only scripts available use port 7, then that is all they can use. Secondly, the service has to respond to broadcast requests. Therefore, you cannot use TCP (which will only respond to directed queries). Many other UDP services only respond to directed queries. Finally, when fraggle was first developed, many firewalls allowed Echos to pass through (because they were used for performance monitoring). More dangerous protocols like NetBIOS (port 137) are already blocked by firewalls.

7. What do these other logs mean?

The following information helps interpret the meaning of events generated by logging systems, not necessarily from a firewall. They might come from the service itself, [intrusion detection systems](#), or really smart firewalls.

7.1 What do the following DNS errors mean?

Response from unexpected source

A DNS server might report this when it receives an incoming response with a different IP address than the corresponding request. There are several causes of this.

Remember that DNS servers will "recursively" send out queries when resolving names on behalf of clients. Each outgoing request is given a unique **transaction identifier**; incoming responses contain the same transaction identifier.

Therefore, if a server sends request #45689 to server 192.0.2.131, but gets response #45689 back from server 192.0.2.3, then it triggers this alert.

The most common cause of this is due to proxying, caching, and dual-homed hosts. For example, the DNS server might have two IP addresses: [192.0.2.131] and [192.0.2.3]. The typical way of writing a DNS server is to not bind the sockets to individual IP addresses. What this means is that the DNS server does not know which IP address the request was received on, nor does it tell the underlying TCP/IP stack which IP address to use when sending the response. Therefore, when the DNS server sends the response, the underlying stack uses one of the IP addresses at random (which can be the wrong one).

Various errors with 127.0.0.1

Some servers are misconfigured to map this address. On the other hand, it is also a hacker technique to cause names within the hacker domain to resolve to addresses within a company (including localhost/127.0.0.1).

Zone transfers (AXFR)

A hacker is attempting to list all the DNS names within a domain. This is an attempt to "map" your network. Managers should consider using *split DNS* aka *shadow domains*, whereby the public DNS contains only those records that must be accessed publicly, but use a separate (and distinct) DNS server for internal machines. Note that some people are fairly benign. If the transfers are coming from the IP addresses 128.9.160.57 and 198.32.4.13, you might want to let them through. <http://www.isi.edu/~bmanning/in-addr-audit.html>.

VERSION.BIND

Hackers are scouring the Internet querying the version of BIND (Berkeley Internet Name Daemon). It is a TXT record of CHAOS class that contains the version string for your server. If you have a server on the Internet, you will see such queries. You might see something like:

```
May 12 04:33:01 ns1 named[171]: unapproved query from [192.0.2.71].35687 for "VERSION.BIND"
```

7.2 What do the following URL's mean in weblogs?

A lot of these pop up in logs as "404 Not Found" errors:

favicon.ico

In MSIE5 (Microsoft Internet Explorer v.50), when a user adds a link to his/her "Favorites" (Bookmarks) or drags the link to the desktop, the browser attempts to retrieve an icon for it. It first searches in the same directory as the file being linked to, then walks up the directory structure until it hits the root. A lot of sites (example: Yahoo!) now supply icons for their sites.

robots.txt

Whenever a search engine (like AltaVista, Infoseek, Excite, etc.) attempts to index your site, it will first get the file

"/robots.txt". If you don't want parts of your website indexed, you can put rules here. On the other hand, hackers will sometimes grab this file as well on the assumption that if you tell a search engine not to index some directories, they might be something interesting to look at. Indeed, network managers do believe that putting directories in "robots.txt" hides them, when in reality it exposes them more.

URL's beginning with `http://`

People occasionally see the following type of line in their webserver log:

```
14:03:00 192.0.2.243 GET /index.html - 200 Mozilla/4.0 - -
14:03:03 192.0.2.243 GET http://www.example.com/ - 200 - - -
```

The first is a normal line, but what is that complete URL starting with "HTTP"? This is an attempt to see if the machine supports proxying. This is how pretty much all HTTP proxies work -- they receive a complete URL, then fetch that URL for the user.

See section [5.3](#) for more info.

7.3 What do the following mean in my RPC portmapper logs?

Clients lookup an RPC program in [portmapper/rpcbind](#) in order to find out which port number the service runs on. A hacker will either **dump** all the listings (using `rpcinfo -p <host>`) or lookup the mapping (using **getport**) for the particular RPC he/she wants to exploits.

As always, these attempts are usually from scans against thousands/millions of machines rather than against you in particular. Every few months, a new exploit script is published for Linux or Solaris services, and script kiddies start scanning the Internet for that service. Most of the vulnerabilities in the services listed are [buffer overflows](#).

Note that on Sun Solaris machines, these services usually have port numbers in the range starting at [port 32770](#). Many other times, RPC services will have ports below 1024, on the assumption that it provides a little better security because

More info on RPC can be found in [RFC1833.txt](#).

7.3.1 What do the following RPC portmapper commands mean?

The portmapper service has six commands (numbered 0-5).

0	NULL	This is a "ping" style command -- it just verifies that the service is running. You see these almost never.
1	SET	If you see this go across the wire, then it is an intrusion attempt. This should be used only internally as RPC-based programs register themselves with portmapper.
2	UNSET	If you see this go across the wire, then it is an intrusion attempt. This should be used only internally as RPC-based programs unregister themselves with portmapper. It is sometimes used as a DoS attack in order to kill your services. Such attacks are frequently spoofed .
3	GETPORT	This is the normal use of portmapper that you should see 99.9% of the time going across the wire. An external client looks up the corresponding port number for the desired service. When reviewing logs, if you see requests to strange services, you can lookup the program number in the table below .
4	DUMP	This dumps all the mappings in the portmapper database. The UNIX command " <code>rpcinfo -p</code> " carries out this command. This is a common reconnaissance technique for hackers.

- 5 CALLIT This may be an attempt to compromise the system. The *callit* feature was created for RPC broadcasts. Because a desired service runs on different ports on different systems, one cannot simply broadcast to it. Therefore, portmapper will accept incoming broadcasts on port 111, then forward them to the appropriate program. However, some even protocols that don't support broadcasts can be compromised by sending the requests through this service.

7.3.2 What do the following RPC program numbers mean?

An RPC program number is assigned by Sun (rpc@sun.com).

I've put an astrisk * next to the ones that have been seen to use the *callit* feature.

100001	rstatd *rup	Allows CPU, network traffic, and disk statistics to be remotely monitored. Hackers may use this as part of recon.
100002	rusersd *rusers	Lists the users on a machine, which reveals lots of info to hackers.
100005	NFS mountd	In late 1998, the RedHat Linux distribution contained a buffer overflow bug in the mountd service running at port 635 . The popularity of RedHat and the fact that the service ran at a common port number resulting in popularity among hackers. Not only did hackers scour the Internet for such machines, but a worm was created to spread via this service. [CA-98.12]
100008	walld *rwall	The program walld, which sends messages to users from the system administrator (such as notifying them the system is about to be rebooted, so they had better save their work). Messages are frequently sent via <i>callit</i> broadcasts.
100068	rpc.cmsd	Solaris <i>Calender Messaging Service</i> In the middle of 1999, a buffer-overflow was found in this service. Immediately after this discovering, hackers started doing extensive scans for this service, resulting in thousands of hacks against web-sites using Solaris. [CA-99-08]
100083	ToolTalk	ToolTalk (<code>rpc.ttdbserverd</code>) [CA-98.11]
100232	rpc.sadmind	Sun <i>Solstice Adminsuite</i> , installed by default on Solaris systems 2.5 and above (2.4 and below installed a similar service called <i>rpc.admind</i>). [CA-99-16]
300019	rpc.amd	Linux <i>Automounter</i> In late 1999, a buffer overflow bug was found in the logging service. While any code based upon the original BSD sources is vulnerable, hackers are probably scanning for the Linux implementation includes in many distros. [CA-99-12]
300055	unixware *	I'm not sure what this service is, but UnixWare sends <i>callit</i> broadcasts across this program number.
300214	FrameMaker *	This number has been assigned to FrameMaker for UNIX. You can download an evaluation copy of this program at: http://www.adobe.com/support/downloads/fmunix.htm . Apparently, the license manager supports <i>callit</i> broadcasts. This license manager supports a "roving" license whereby many people can have it installed, but only a few can use the product.
390109	nsrstat *	<i>Legato NetWorker Server Remote Status</i> . This is a backup service (also OEMed as Solstice Backup). Status updates are broadcast via <i>callit</i> .
200100001	netinfobind *	Part of NeXTstep replacement for YP. When a child netinfo process starts up, it searches for a parent by broadcasting a NIBIND_BIND procedure (function=8) on the local subnet.

7.4 What do the following mean in my SMTP (e-mail) logs?

While not your classic packet filtering firewall, SMTP (e-mail) are important gateways between the outside world and your internal network. They should be considered along the same lines as your firewall.

7.4.1 What is this message about "relay" attempts?

A *relay* is where somebody sends your e-mail server not destined for anybody who you serve e-mail for. For example, I might connect to your e-mail server and attempt to send mail to "test@example.com". Your e-mail server should not accept the e-mail ("relay not allowed"). Your e-mail server should only accept incoming e-mail to your users (or outgoing e-mail from your users).

The problem is that many administrators simply install servers without taking these simple precautions. Spammers take advantage of this fact. They give a single e-mail to the mail server and a recipient list containing hundreds of unrelated recipients. This allows them to send huge quantities of e-mail using a slow dialup connection. This is important because once the ISPs get enough complaints, they will terminate the user's account, so they must continually get new dialup connections. It also has the effect of partially hiding the true source of the spam.

If you get error messages about relaying, that is a good thing: you've configured your server correctly. If you don't get such messages, this is a bad thing. This means that you are probably not rejecting relayed messages. Has your server seemed slow lately?

Not only do spammers hunt for open relays, anti-spam organizations do the same in an attempt to "blacklist" open relays. Some of the good guys are:

[IMC](#)

The Internet Mail Consortium reports that in 1999, roughly 17% of e-mail systems had open relays.

[MAPS RBL](#)

The MAPS RBL (Realtime Blackhole List) allows you to configure your e-mail server to blackball known open relays that send out bulk spam. It is used by a huge percentage of e-mail servers on the Internet.

[ORBS](#)

Scans the Internet looking for open relays. ORBS uses relay tests from New Zealand (e.g. manawatu.co.nz).

Not only do you receive relay attempts from spammers, you also get attempts from anti-spam organizations. There are several organizations that regularly scan the Internet looking for open relays. The most common is from "manawatu.co.nz"; don't get too upset -- they

7.4.2 What are these messages about rejected EXPN and VRFY attempts?

The "expand" and "verify" commands will expand mailing lists or verify user names (respectively).

If you do the command "VRFY root", you might be able to find out the postmaster's e-mail address. This is good reconnaissance technique.

By doing a "VRFY decode" or "VRFY uudecode", you might be able to find out some security holes in the system related to these subsystems. Other commonly scanned user names are "bbs", "lp", "demo", "guest", and "debug".

Some systems have buffer overflows in this command, either in the command itself or in the logging system behind the command. You might see entries for very long strings like "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx".

If you see a bunch of these in a row, you are probably being scanned by a vulnerability scanner (ISS/CyberCop/Nessus). They will generate a bunch of other junk in your logs as well.

7.5 What are these identd/auth messages?

The UNIX `identd` service identifies which of the logged on users owns a particular TCP connection.

7.5.1 What does No Ident response mean?

Some IRC servers spit this out. It means that the ident service at port [113](#) isn't available. Either the firewall is blocking it or it isn't running. Most IRC clients come with an ident service.

8. How do I configure filters?

Many of the logged packets on your firewall result from incorrect configuration. This section doesn't describe how to configure your firewall, but instead helps describe some common configuration steps you might want to take when you see rejects pop up in your firewall logs.

8.1 What ICMP traffic should I deny?

The "correct" configuration of ICMP filters in a firewall is hotly debated. The problem is that ICMP are the "control messages" for TCP/IP. If you block some incoming ICMP, then you will break communication.

The absolute minimum ICMP traffic to allow is the packets dealing with TCP path MTU discovery. Fragmenting a stream is more efficient at the TCP layer rather than the IP layer, so the TCP layer will try to discover when IP packets are being inadvertently fragmented. They do this by setting the "[DF](#)" (Don't Fragment) on all outgoing packets. When a router cannot forward the packet because it is too big, rather than fragmenting it, it sends back a "fragmentation needed" ICMP packet (type=3/code=4). The TCP stack then starts sending smaller IP packets, segmenting the data at the TCP layer rather than allow routers to fragment at the IP layer. Therefore, firewalls must be configured to allow incoming ICMP type=3, code=4 packets.

Another issue is **Host unreachable** and **Destination Unreachable** packets. Allowing these to come in through your firewall will allow connections to timeout faster, but they can also be used as a denial of service attack (by disconnecting clients from servers).

Users will constantly ask for the ability to ping and traceroute machines on the Internet. Most firewall administrators will eventually give into these demands. Nobody really needs to ping/traceroute, but they really want to. It should be remembered, however, that ICMP ping responses are often used as a [covert-channel](#). (The massive DDoS attacks against Internet portals used this as a covert channel).

For more information on this, you may want to consult "Protect and Survive Using IBM Firewall 3.1 for AIX", IBM publication SG24-2577-02. See <http://www.redbooks.ibm.com> for more info. I disagree with it, though.

Another good document is <http://www.worldgate.com/~marcs/mtu/>.

8.2 split DNS

Keep a separate primary DNS server for internal use vs. external use. An external DNS server should only have entries for publicly available servers, such as web servers, FTP servers, e-mail servers, and so forth.

9. Packet Zen

You can deduce a lot of information by examining fields within the TCP/IP headers. What seems like random or meaningless numbers to most people can in fact reveal a lot of information.

9.1 How do I interpret the IP identification fields? (IP ID Zen)

The IP [identification \(ID\)](#) field is a two-byte field contained within the packet. Its sole purpose in life is allow IP packets to be fragmented: all fragments should contain the same ID as the original packet so that they can be pasted back together again. Most systems use the concept of a *monotonically increasing* ID: for each packet sent, the field is increased by one.

There is a little twist to this scenario. A little-endian machine (like Intel processors) stores numbers in reverse byte-order than how numbers are represented on the wire. This means that a monotonically increasing integer from a Wintel box will increment the high-order byte first, whereas a Sun SPARC box will increment the low-order byte first. Therefore, lets say that you are being pinged steadily from both a Sun SPARC and a Wintel, you will see the following sort of progression in the IP ID field:

SPARC Wintel

0x01FD 0xFD01

0x01FE 0xFE01

0x01FF 0xFF01

0x0200 0x0002

0x0201 0x0102

The above numbers are shown in [hexadecimal](#), which shows the byte-order problem. However, many firewall logs (stupidly) show these numbers in decimal. If a firewall system assumes the number is big-endian but the incoming packets are little endian, then the progression of the numbers is hidden. For example:

IP ID Big-endian Little-endian

01 FD	509	64769
-------	-----	-------

01 FE	510	65025
-------	-----	-------

01 FF	511	65281
-------	-----	-------

02 00	512	2
-------	-----	---

02 01	513	258
-------	-----	-----

This entire issue is complicated by the fact that a firewall running on a platform doesn't have to base its decimal calculation of the IP ID field on the underlying CPU. What I mean by this is that the C code that interprets the IP ID could be written in two ways;

```
/* ID field is a 2-byte number at offset 4 within the IP header */
int ipid_cpu = *(unsigned short*)(iphdr+4);
int ipid_be = iphdr[4] * 256 + iphdr[5];
```

The first example is CPU dependent: x86 CPUs will pull it out as a little-endian number, but SPARC CPUs will pull it out as a big-endian number. The second form is **CPU independent**: it tells all CPUs to interpret the field as a big-endian number. Note: `ntohs(*(unsigned short*)(iphdr+4))` will crash a SPARC CPU and is not a good solution

Therefore, if you are running a Linux-based x86 firewall that interprets the IP ID field as a little-endian number, then a string of packets from a Wintel box will demonstrate a monotonically increasing number. However, a stream from a SPARC box will show skipping numbers. Conversely, if the Linux-based firewall uses the (correct) field parsing method, you'll see the reverse.

Moral of the story: Find out the byte order interpretation of the IP ID field used within your firewall. Also send your firewall vendor flames telling them to get with the program and represent the field in hex in the first place.

9.2 How do I interpret the TTL fields? (TTL Zen)

The [Time-to-Live \(TTL\)](#) field is decremented by one every time a router forwards a packet. When it reaches zero, the router discards the packet. *Routing loops* are a frequent occurrence on the Internet as routers get confused as to the proper direction in which to forward packets. The TTL mechanism assures that packet eventually "die" when and don't get

routed in loops forever.

It also means that you can tell how far away a person is from the TTL field, and sometimes what kind of platform they are running. Most Windows machines send packets with a starting TTL of 128. This means that if your firewall log shows a TTL=112, then you can make the guess that the sender is 16 hops away, and that they are using a Windows machine.

Conversely, UNIX machines typically choose 64 as the starting TTL, so a packet when the TTL is 51, then it probably isn't Windows, but it is probably 13 hops away.

This technique was once used to find the source of [nmap](#) decoy scans. The decoys were given random TTLs, but the real scans were given normal TTLs. This allowed the astute observer the ability to sift through the incoming decoys and find the real scan. The *nmap* program was soon fixed to randomize the TTL of the real scan as well.

9.x Other resources

Passive [fingerprinting](#) of people is a common topic when sniffing packets. Some articles that describe this are:

Max Vision's "Passive Host Fingerprinting"

<http://dev.whitehats.com/papers/passive/index.html>

Lanz Spitzner's "Passive Fingerprinting"

<http://www.enteract.com/~lspitz/finger.html>

10. What's the deal with NetBIOS (UDP port 137)?

NetBIOS requests to UDP port 137 are the most common item you will see in your firewall reject logs. This comes about from a *feature* in Microsoft's Windows: when a program resolves an **IP address** into a **name**, it *may* send a NetBIOS query to IP address. This is part of the *background radiation* of the Internet, and is nothing to be concerned about.

The discussion of these NetBIOS packets crops up over and over again on firewall/incident mailing lists. In this section, I've tried to come up with the "definitive" answer to this question.

Note that you will see NetBIOS scans, such as from hackers running the *Legion* NetBIOS scanner or other scanners. In this case, you'll likely see a scan of your entire address range. The important thing to remember is that few NetBIOS packets are from hostile intent.

10.1 What does it mean to resolve an IP address to a name?

You are familiar with the **normal** DNS resolution. You type into your web browser an address like <http://www.robertgraham.com>, but it looks up the web site's name with DNS in order to find IP address. Underneath, it is really IP addresses that are used for communication.

We call DNS a **directory service**, where the word *directory* has the same meaning as in phone networks. In the U.S., we can dial *directory assistance* at 411 rather than looking up a name in the phone book. Either way, the goal is to lookup a **name**, and receive a **number**.

In a similar manner, sometimes you have a number, and you want to find the name. Let's say that you have caller ID and somebody calls you with the phone number (212) 555-1038. This doesn't tell you who this is, so you want to do the reverse lookup and discover the person's name.

In much the same fashion, the Internet provides a number of capabilities to resolve an IP address into a name.

10.2 Where do the NetBIOS packets come from? Why does Windows send them?

On virtually all systems (UNIX, Macintosh, Windows), programs call the function `gethostbyaddr()` with the desired address. This function will then do the appropriate lookup, and return the name. This function is part of the [sockets](#) API.

The key thing to remember about `gethostbyaddr()` is that it is **virtual**. It doesn't specify *how* it resolves an address into a name. In practice, it will use all available mechanisms.

If we look at UNIX, Windows, and Macintosh systems, we see the following techniques:

- DNS in-addr.arpa PTR queries sent to the DNS server
- NetBIOS NodeStatus queries sent to the IP address
- lookups in the [/etc/hosts](#) file
- AppleTalk over IP name query sent to the IP address
- RPC query sent to the UNIX NIS server
- NetBIOS lookup sent to the WINS server
- etc.

Windows systems do the [/etc/hosts](#), DNS, WINS, and NodeStatus techniques.

In more excruciating detail, Microsoft has a generic system component called a *naming service*. All the protocol stacks in the system (NetBIOS, TCP/IP, Novel IPX, AppleTalk, Banyan, etc.) register the kinds of name resolutions they can perform. Some RPC products will likewise register an NIS naming service. When a program requests to resolve an address, this address gets passed onto the generic naming service. Windows will try each registered name resolution subsystem sequentially until it gets an answer. (Side note: User's sometimes complained that accessing Windows servers is slow. This is caused by installing unneeded protocol stacks that must timeout first before the real protocol stack is queried for the server name.)

The order in which it performs these resolution steps for IP addresses can be configured under the Windows registry key `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\ServiceProvider`. Of course, that doesn't help you the firewall admin.

10.3 But my network doesn't run any Windows machines. Why am I being sent these packets?

It has nothing to do whether you run Windows, NetBIOS, or Samba on your machines.

The process is simply that a program requests the name for an IP address, and sends this request to all the protocol stacks. If the NetBIOS stack receives such a request, it always sends a NetBIOS query to the IP address. It doesn't matter if you have (or haven't) an existing NetBIOS connection to the machine.

In other words, the only requirement necessary in order to receive such packets is that you have an IP address.

10.4 Why are reverse resolutions so common?

One would think that a reverse query would be rare. They are instead very common. Here are some reasons why programs might do reverse lookups.

[ping.exe](#)

If the user executes a `ping -a 192.0.2.168`, then Windows will attempt to find the name for that address. This doesn't happen so often.

[tracert.exe](#)

The traceroute program finds all the hops between the client and the server. Users sometimes do this from the command-line. The most common source of this is from programs that automatically traceroute the servers the user visits. Note that if they are tracing through several hops, you will get separate queries for each hop.

Microsoft's IIS web server

Microsoft's webserver has the option to log the machine name of the client accessing the web site. Each time one of your users behind your firewall browses an IIS-based server, you'll get a query for the name of the user's machine.

Logfile analyzers

Even if name resolution is disabled on the webserver, the site administrator may run the webserver logfiles through a reporting tool like Webtrends. Most of these tools have the ability to resolve IP addresses to names. At this stage, you will see a flurry of port 137 packets from the address the tool is run from (which may be different from the original webserver). This is especially a problem because they request such a huge amount of DNS PTR queries that they overwhelm the DNS server. Thus, even though DNS queries would normally be resolved, they might fail during analysis of a log file, thereby generating NetBIOS queries. Since these logfile analyzers are often run on a scheduled (i.e. nightly) basis, you may see such activity from the same host during the same period of the day.

Client apps

Beyond web browsing, reverse IP name resolution is a fixture in many Windows client apps like IRC, ICQ, and so forth.

Personal firewalls

Personal firewalls will attempt reverse resolution of the IP addresses. The "auto-learning" personal firewalls that prompt the user for each outgoing connection can be particular offenders in this regard. If BlackICE Defender sees an intrusions attempt, it may also do its own NetBIOS lookups independently from the underlying Windows system.

Note that starting in late 1999, desktop security tools like personal firewalls have exploded. This means that the number of NetBIOS queries have dramatically risen.

Also, see section [10.6](#) for an explanation of how a simple configuration error in DNS can cause you to be suddenly flooded with such requests.

10.5 What is the exact signature I can expect to see?

Windows machines use both a source port of 137 as well as a destination port of 137. In contrast, if UNIX machines attempt to resolve NetBIOS names (via SAMBA), they will use dynamic ports above [1024](#).

If the Windows box is trying to find the name for the IP address 192.0.2.21, it will do the following steps:

- Lookup the DNS "PTR" record for `21.2.0.192.in-addr.arpa`; this request is sent to the local DNS server, which recursively forwards the query to the appropriate DNS server as required.
- If the DNS answer comes back, it *won't* query NetBIOS. If a negative response comes back, it will immediately query NetBIOS. If the DNS server times-out, it will wait 14-seconds, then query NetBIOS.
- When resolving with NetBIOS, it will send out a "NodeStatus" query that is sent to the 192.0.2.12:137 from `x.x.x.x:137`. (I.e. the query is sent to the IP address being resolved to its port 137, and is sent from the Windows machine port 137).
- The NetBIOS request is a "NodeStatus" query that looks up the name "*". It is 50 bytes worth of data (58 including the UDP header, 78 including the IP header, 92 including an Ethernet header).
- Three NetBIOS queries are sent with a 1.5 second timeout.

The personal firewall *BlackICE Defender* will may do its own NetBIOS queries separate from the underlying Windows OS. These will look like UNIX queries from dynamic ports, and have longer, progressive timeouts of 15-seconds, 30-seconds, and 1-minute.

Packet Dump

The packet looks something like the example below. For more information about interpreting this, please see my sniffing FAQ at <http://www.robertgraham.com/pubs/sniffing-faq.html>.

```
ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
IP: ID = 0x3E16; Proto = UDP; Len: 78
UDP: Src Port: NETBIOS (137); Dst Port: NETBIOS (137); Length = 58
NBT: NS: Query req. for *<00...(15)>
      NBT: Transaction ID = 57032 (0xDEC8)
      + NBT: Flags Summary = 0x0000 - Req.; Query; Success
      NBT: Question Count = 1 (0x1)
      NBT: Answer Count = 0 (0x0)
      NBT: Name Service Count = 0 (0x0)
      NBT: Additional Record Count = 0 (0x0)
      NBT: Question Name = *<00...(15)>
      NBT: Question Type = Node Status Request
      NBT: Question Class = Internet Class
```

```
00000:  00 E0 18 E0 0C E7 00 40 05 A4 79 32 08 00 45 00  .....@..y2..E.
00010:  00 4E 3E 16 00 00 80 11 2F CE 0A 0A 00 09 C0 00  .N>...../.....
00020:  02 A8 00 89 00 89 00 3A 14 AB DE C8 00 00 00 01  .....:.....
00030:  00 00 00 00 00 00 20 43 4B 41 41 41 41 41 41 41  ..... CKAAAAAAAA
00040:  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAAAA
00050:  41 41 41 41 41 41 41 41 00 00 21 00 01          AAAAAAA..!..
```

10.6 How do I reduce this traffic so I don't get so many?

Windows will not send a NetBIOS query if the initial DNS query comes back in a timely manner. **Let me repeat: Windows only sends NetBIOS queries when the DNS lookup fails. Therefore, the proximate cause of NetBIOS queries is a fault in the DNS system. The first thing you should hunt down is the DNS fault causing the DNS PTR queries to fail.**

If you are seeing a lot of these requests, it probably means you have one of the following DNS issues.

- Your DNS servers are slow; the Windows machine needs a response within 14 seconds.
- Your link is unreliable/congested, causing the DNS queries to be dropped.
- You haven't configured the PTR entries within your DNS server.
- Your ISP doesn't forward the PTR queries to your DNS server.
- The client's ISP cannot handle CNAME -> PTR indirection for CIDR addresses.
-

Note that in this day/age with CIDR and address blocks smaller than 255 members, a lot of ISPs don't know *how* to forward DNS PTR requests to your server.

No matter what you do, you will still get requests because of configuration errors on the client's ISP. However, making sure the issues above are resolved on your own DNS servers will be an important first step.

10.6.1 What is a DNS PTR query?

For reasons of historical irrelevance, a normal DNS query is called an **A** record. A reverse query is called a **PTR** (pointer) query. The names A and PTR don't really mean anything; remember that a lot of such things come about because some engineer created "temporary" names from the top of his head, meaning to change them later, but they sort of just stick around.

The thing to remember is that **A and PTR queries are unrelated.**

When you register your domain name (example.com) you go to the owner of .com (Network Solutions) and purchase the address. As part of your registration, you tell Network Solutions something to effect "Please pass any DNS queries for the domain example.com to my DNS server ns1.example.com which is located at the IP address 192.0.2.168".

Thus, when resolving www.example.com, you first ask .com for the DNS server for example.com, which is ns1.example.com/192.0.2.168. You then ask that server for the *A record* for www.

Now going the reverse direction is a bit tougher. When trying to figure out who owns the IP address 192.0.2.3, you've got a problem. What is the first step? The solution was to query for a PTR record with the pseudo-name that looks something like "3.2.0.192.in-addr.arpa". Like the .com domain, the .arpa domain is run by a special company. It forwards the requests to the backbone ISPs, which then forward the request to the smaller ISPs and customers.

This forwarding mechanism is easily broken due to CIDR addresses. An ISP may assign 192.0.2.[0.127] to one customer, and 192.0.2.[128-255] to another customer. In order to fix this issue, the ISP must support special CNAME records that delegate lookups. For the network 192.0.2.128/25, then the CNAME record would look like 128/25.0.2.192.IN-ADDR.ARPA. This is kinda complex, easy to get wrong, and the administrators at ISPs often don't know how to do it right.

Please see CNAME -> PTR indirection described in [RFC 2317](#) for more details on this. Also see <http://www.dns.net/dnsrd/> for extensive DNS resources.

Conclusion

The upshot is that you probably have a CIDR allocation that breaks PTR queries causing NetBIOS queries. Harangue your ISP until they fix this.

10.7 What attacks can people go through NetBIOS/137?

Legion scans

There are numerous tools that scan for open shares. The first popular tool for this was called "Legion" from Rhino9. Since then, numerous other tools have been created. Some of these tools will do a lookup on port 137 before connecting to TCP port 139.

NetBIOS worms

Starting in 1999, numerous NetBIOS worms have been seen. These include ExploreZip virus/worm, Network.VBS VisualBasic script, and the 911 worm (which also calls 911 out your modem). All of these worms will attempt connection to you machine.

NetBIOS scans

Sometimes people just scan the Internet looking for people's names. Since most people leave port 137 open, it is pretty fun.

A. Appendix

http://www.cert.org/tech_tips/intruder_detection_checklist.html

CERT's **Intruder Detection Checklist**. If you believe you've been compromised, this document describes how to go through your UNIX system and find signs of this intrusion.

<http://www.usdoj.gov/criminal/cybercrime/reporting.htm>

If you have evidence of a cybercrime that you believe warrants the attention of the FBI, this is a good place to start. Note that you can't simply hand it off to them and say "you take care of it". They are only willing to take part of you are willing to spend the necessary time in gathering evidence. For example, you may have to ship your compromised machine to them.

[fin]