



# Global Information Assurance Certification Paper

Copyright SANS Institute  
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

## Interested in learning more?

Check out the list of upcoming events offering  
"Hacker Tools, Techniques, Exploits, and Incident Handling (Security 504)"  
at <http://www.giac.org/registration/gcih>

# **SANS GIAC Certified Intrusion Detection Analyst Extra Work and 10 Detects for Practical**

Prepared by Martin J Seery

© SANS Institute 2000 - 2002, Author retains full rights.

# TABLE OF CONTENTS

<b>SUMMARY .....</b>	<b>1</b>
<b>EXTRA WORK ASSIGNMENT .....</b>	<b>2</b>
2.1 TCP/IP FOR INTRUSION DETECTION AND PERIMETER DEFENSE .....	2
2.2 INTRUSION DETECTION AND PACKET FILTERING: HOW IT REALLY WORKS .....	4
2.3 INTRUSION DETECTION ANALYSIS –SHADOW STYLE .....	7
2.4 NETWORK BASED INTRUSION DETECTION ANALYSIS.....	9
2.5 INTRUSION DETECTION WORKSHOP.....	11
<b>GIAC PRACTICAL FOR MARTIN SEERY.....</b>	<b>14</b>
DETECT 1 .....	14
DETECT 2 .....	17
DETECT 3 .....	20
DETECT 4 .....	22
DETECT 5 .....	26
DETECT 6 .....	33
DETECT 7 .....	35
DETECT 8 .....	41
DETECT 9 .....	48
DETECT 10.....	52
<b>APPENDIX.....</b>	<b>55</b>
OCLM SECRETS - BUBBA'S GUIDE TO THE TELNET INTERFACE .....	A1
PORTS USED BY TROJANS (2000-05-20).....	A17

© SANS Institute 2000 - 2002. Author retains full rights.

## Summary

This document contains the necessary extra work required for GCIA (GIAC Certified Intrusion Detection Analyst) certification and 10 detects for the completion of the practical. The Extra Work section is broken down into five subsections reflecting the SANS GIAC Intrusion Detection course layout. All questions were created using materials taken from their respective section and cross-referenced when possible using TCP/IP Illustrated volume 1. The Practical section contains 10 detects captured by SNORT version 1.6 using 05172kany.rules file from 'http://snort.rapidnet.com'. (Full Current RapidNet Ruleset using 'any' instead of "\$HOME\_NET" Variables). The NIDS monitored traffic on a small subnet using a 3com ISDN Office Connect LAN Modem (OCLM) Router as a gateway to the Internet. The 3com OCLM had NAT disabled to allow easy access to the machines on the subnet. The subnet had three active hosts, a Windows NT 4.0 Workstation with tightened security, a Windows 98 machine with File and Print sharing disabled, and the NIDS, a LINUX machine running snort with an IP address of 0.0.0.0. The NT Workstation was used to visit numerous hacker sites leaving a trail of breadcrumbs for persons of malicious intent. In conjunction with having tightened security, the NT Workstation was running Back Officer Friendly. For those not familiar with BackOfficer Friendly, it is a spoofing server application that runs on your Windows or UNIX system. Basically, it acts like a light-weight host based virtual honey pot pretending to be a Back Orifice server or a server running a variety of services, such as FTP, HTTP, Telnet, POP3, IMAP2, and SMTP. All of these routines were enabled.

NOTE: Due to issues surrounding the legality of providing information about the behavior of another companies IP addresses, both the source and destination IP addresses have been sanitized. The monitored networks IP addresses have the first two octets sanitized. IP addresses categorized as "intruder" will have the last two octets of the IP address sanitized.

Example:

My subnet = x.x.my.net

Intruder = not.me.x.x

This issue was discussed with and cleared with Stephen Northcutt via email correspondence. Questions regarding the validity of this claim should be brought to the attention of Stephen Northcutt.

## Extra Work Assignment

### 2.1 TCP/IP for Intrusion Detection and Perimeter defense

1. Ephemeral port range can be described as:

- a) Server ports < 600
- b) Ports 135-139
- c) Client ports >1023
- d) Server Ports > 1000

Answer is 'c'. Ports 1024 and above. These are often called Ephemeral ports. (Workbook 2.1, pg. 1-21) Ephemeral ports are 'short lived' ports opened by the client and usually range between 1024 and 5000. The port numbers above 5000 are intended for other servers. Solaris 2.2 is a notable exception. By default the ephemeral ports for TCP and UDP start at 32768. (TCP/IP Illustrated Volume 1, pg. 13)

2. Which best represents the class A, B, and C IP Class Addresses respectively?

- a) 1-126, 127-191, 191-254
- b) 0-127, 128-191, 192-223
- c) 0-126, 127-192, 193-254
- d) 1-127, 128-191, 192-254

The answer is 'b'. (See table in Workbook 2.1, pg. 1-16) The ranges for different IP Address are: Class A 0.0.0.0-127.255.255.255, Class B 128.0.0.0-191.255.255.255, Class C 192.0.0.0-223.255.255.255, Class D 224.0.0.0-239.255.255.255, Class E 240.0.0.0-255.255.255.255 (TCP/IP Illustrated Volume 1, pg. 8)

3. IP and MAC pairs can be found in:

- a) ARP cache
- b) MAC cache
- c) Host tables
- d) IP tables

Answer is 'a'. Any hosts on the network that are listening for broadcasts will see initial ARP request and can cache the MAC and IP addresses. (Workbook 2.1, pg. 1-14) ARP tables are held in memory and can be displayed by typing "arp -a" from a command line. The 48-bit Ethernet addresses are displayed as six hexadecimal numbers separated by colons. (TCP/IP Illustrated Volume 1, pg. 56)

4. Which Protocol live in the Network Layer of the IP Model?

- a) IP
- b) ICMP
- c) IGMP
- d) All of the above

Answer is 'd'. The Network Layer is concerned with routing and how to get from our Host to another Host. This is where IP, ICMP, and IGMP do there work. "The Network Layer (sometimes called the Internet layer) handles the movement of packets around the network. (TCP/IP Illustrated Volume 1, pg. 2)

5. In the following example the ending sequence number is incorrect. What should it be?

"04:30:02.210000 my.host.1188 > your.host.23: S 28304100:00210034(64) win 512"

- a) 2100001
- b) 28304164
- c) 28304100
- d) 28304101

Answer is 'b'. The ending sequence number is the sum of the initial sequence number plus the number of TCP data bytes sent in the TCP segment. (Workbook 2.1, pg. 2-6) The ending sequence number is equal to the starting sequence number plus the total data size. The sequence number is a 32-bit unassigned number that wraps back around to 0 after reaching 2 to the 32 – 1. (TCP/IP Illustrated Volume 1, pg. 226)

6. What is true about the following output:  
my.host > your.host: icmp: echo request (frag 12345:87@0+  
my.host > your.host: icmp: echo request (frag 12345:40@87

- a) Packet is normal
- b) Packet is abnormal
- c) Fragment is incomplete
- d) Icmp cannot be fragmented

Answer is 'b'. Packet is abnormal. During the 2.1 course, Hal Pomeranz, made a note on Page 3-15 of workbook 2.1 indicating that the data portion of the fragment, with the exception of the last fragment, must be a multiple of eight. (Workbook 2.1, pg. 3-15) Fragmentation requires that the data portion of the generated fragments (that is, everything excluding the IP header) be a multiple of eight bytes for all fragments other than the final one. (TCP/IP Illustrated Volume 1, pg. 150)

7. What happens if the data contained in the DNS datagram response exceeds 484 bytes?  
a) The truncated bit is turned on  
b) The DNS query is reissued using TCP  
c) Port 53 is used for the transfer  
d) All of the above

Answer is 'd'. The Maximum allowable size for a UDP DNS response is 512 bytes. Of the 512 bytes, at least 20 bytes are reserved for the IP header and 8 are used for the UDP header leaving 484 bytes for the DNS message. (Workbook 2.1, pg. 4-19)

Many UDP applications are designed to restrict their application data to 512 bytes or less. (TCP/IP Illustrated Volume 1, pg. 160) When the resolver issues a query and the response comes back with the TC bit set ("truncated") it means the size of the response exceeded 512 bytes, so only the first 512 bytes were returned by the server. The resolver normally issues the request again, using TCP. (TCP/IP Illustrated Volume 1, pg. 206)

The ICMP protocol:

- a) is used for error messages.
- b) uses ephemeral port number when issued from a client.
- c) works in the Link layer of the IP model
- d) all of the above

Answer is 'a'. ICMP has no port numbers as are found in the transport layer protocols. When ICMP error messages are delivered, the receiving host may respond internally, but may not communicate anything back to the informer. (Workbook 2.1, pg. 5-6) ICMP is often considered part of the IP layer. It communicates error messages and other conditions that require attention. (TCP/IP Illustrated Volume 1, pg. 69)

8. Smurf, Winfreeze, TFN, and Loki are all examples of:  
a) Malicious UDP.  
b) Malicious ICMP.  
c) Trojans  
d) WinNuke programs.

Answer is 'b'. Just like many other protocols, ICMP can be used for evil purposes. (Workbook 2.1, pg. 5-29)

9. The following is an example of:

```
1.2.3.4.35955 > 4.3.2.1.21: S 2132546578:2132546578(0)
4.3.2.1.21 > 1.2.3.4.35955: S 9887655432:9887655432(0) ack 2132546579
1.2.3.4.35955 > 4.3.2.1.21: . ack 1
4.3.2.1.21 > 1.2.3.4.35955: P 1:24(23) ack 1
1.2.3.4.35955 > 4.3.2.1.21: . ack 24
```

- a) Port Scan
- b) DNS Query.
- c) FTP session negotiation
- d) Zone transfer.

Answer is 'c'. In the above example, we see that the port FTP connection is established between the client using ephemeral port '35955' and the server port '21'. The three-way handshake is completed and some data, usually a welcome message, is passed between the two. (Workbook 2.1, pg. 6-19)

10. The following is an example of:

```
1.2.3.4 > 4.3.2.1: ICMP: echo reply [tos 0xfc]
1.2.3.4 > 4.3.2.1: ICMP: echo reply [tos 0xfc]
1.2.3.4 > 4.3.2.1: ICMP: echo reply [tos 0xfc]
1.2.3.4 > 4.3.2.1: ICMP: echo reply [tos 0xfc]
1.2.3.4 > 4.3.2.1: ICMP: echo reply [tos 0xfc]
```

- a) ICMP port scan
- b) Crafted Packet
- c) Spoofed address 4.3.2.1
- d) Spoofed address 1.2.3.4

Answer is 'c'. This is an example of all stimulus no response. It is very possible that 1.2.3.4 did not initiate an echo reply without some other stimulus eliciting this activity. It is most likely that someone spoofed the 4.3.2.1 IP address and sent echo requests to 1.2.3.4. (Workbook 2.1, pg. 6-34)

## 2.2 Intrusion Detection and Packet Filtering: How it Really Works

1. Ethernet's Maximum Transmission Unit or MTU is:

- a) 1400
- b) 1500
- c) 1024
- d) 512

Answer is 'b'. The Ethernet spec calls for a maximum frame length (including the Ethernet header) of only 1500 bytes. (Workbook 2.2, pg. 11) Also see, TCP/IP Illustrated Volume 1, pg. 30, Figure 2.5 Typical maximum transmission units (MTUs).

2. How many Flag bits does a TCP header contain?

- a) 6
- b) 7
- c) 8
- d) 9

Answer is 'a'. There are six flag bits that may be turned on in various combinations in the TCP header. . (Workbook 2.2, pg. 18) Also see, TCP/IP Illustrated Volume 1, pg. 227.

3. A computer that listens to all traffic on the network regardless of destination IP is said to be in which mode?
- Server
  - Multihomed
  - Promiscuous
  - Active

Answer is 'c'. Sniffers operate by putting the local interface of the Host machine into promiscuous mode. Normally, Hosts only listen for and respond to packets destined for their IP or hardware address. (Workbook 2.2, pg. 21). Most interfaces can be placed in to a promiscuous mode whereby they receive a copy of every frame. (TCP/IP Illustrated Volume 1, pg. 169)

4. In a large ICMP echo request that has been fragmented, which fragment contains the ICMP header information?
- All
  - First
  - Last
  - None

Answer is 'b'. The first fragment contains the ICMP header information, which is encapsulated in the payload of the IP datagram. (Workbook 2.2, pg. 25)

5. The following trace is an example of which attack?
- ```
10:56:32.395383 192.168.101.20.139 > 192.168.101.20.139: S
10:56:33.123456 192.168.101.20.139 > 192.168.101.20.139: S
10:56:34.743256 192.168.101.20.139 > 192.168.101.20.139: S
```
- Smurf
  - WinNuke
  - Land
  - Syn Flood

Answer is 'c'. The packet is spoofed so that the packet appears to have come from the target host itself. (Workbook 2.2, pg. 49)

6. The following trace is an example is most likely an example of which?
- ```
10:56:33.123332 192.168.101.20.1613 > 202.119.20.10.31337: UDP 19
10:56:33.123456 202.119.20.10.31337 > 192.168.101.20.1613: UDP 53
10:56:33.123458 202.119.20.10.31337 > 192.168.101.20.1613: UDP 48
10:56:33.123460 202.119.20.10.31337 > 192.168.101.20.1613: UDP 64
10:56:33.123466 202.119.20.10.31337 > 192.168.101.20.1613: UDP 92
10:56:33.123490 202.119.20.10.31337 > 192.168.101.20.1613: UDP 34
```
- Back Orifice Probe
  - Loki Probe
  - Land Attack
  - Back Orifice Session

Answer is 'd'. This trace is similar to a Loki session in that a single packet from the client stimulates several response packets from the server. The key to identifying Back Orifice is the server port 31337 (ELEET). (Workbook 2.2, pg. 101)

7. If a SYN-ACK is sent to a Win95 host, the Win95 host will:
- Send an ACK
  - Send a RESET
  - Send a SYN-ACK
  - Blue Screen

Answer is 'b'. The host will send a reset regardless of the state of the target port (open or closed). (Workbook 2.2, pg. 119) In general, a reset is sent by TCP whenever a segment arrives that doesn't appear correct for the referenced connection. (TCP/IP Illustrated Volume 1, pg. 246)



8. Which is not an example of an “impossible” flag setting?

- a) FIN-ACK
- b) FIN only
- c) SYS-FIN
- d) RST-FIN

Answer is ‘a’. Impossible packet, these are packets where the following flag combinations are present: no flags, FIN only, SYN-FIN, RST-FIN, SYN-RST. The allowed packets are then: SYN only, SYN-ACK, ACK only, FIN-ACK, RST only, RST-ACK. (Workbook 2.2, pg. 139)

9. The following trace is an example of:

```
10:56:33.123456 202.119.20.10.31337 > 192.168.101.20.80: S
10:56:33.123458 202.119.20.10.31337 > 192.168.101.21.80: S
10:56:33.123460 202.119.20.10.31337 > 192.168.101.22.80: S
10:56:33.123466 202.119.20.10.31337 > 192.168.101.23.80: S
10:56:33.123490 202.119.20.10.31337 > 192.168.101.24.80: S
```

- a) SYN Flood
- b) Web Server Scan
- c) Back Orifice Scan
- d) Smurf Attack

Answer is ‘b’. A network scan (in this case for HTTP Servers) is easily recognized – the attacker sends SYN packets to the same service port (Port 80 HTTP) on many different machines. (Workbook 2.2, pg. 111)

10. The following trace is an example of:

```
10:56:33.123456 202.119.20.10.38758 > ns.my.net.33476: UDP 12
10:56:33.123458 202.119.20.10.38758 > ns.my.net.33477: UDP 12
10:56:33.123460 202.119.20.10.38758 > ns.my.net.33478: UDP 12
10:56:33.123466 202.119.20.10.38758 > ns.my.net.33479: UDP 12
10:56:33.123490 202.119.20.10.38758 > ns.my.net.33480: UDP 12
```

```
10:56:33.883456 223.11.23.11.48412 > ns.my.net.33510: UDP 12
10:56:33.883458 223.11.23.11.48412 > ns.my.net.33512: UDP 12
10:56:33.883460 223.11.23.11.48412 > ns.my.net.33513: UDP 12
10:56:33.883466 223.11.23.11.48412 > ns.my.net.33514: UDP 12
10:56:33.883490 223.11.23.11.48412 > ns.my.net.33515: UDP 12
```

```
10:56:34.423456 202.119.192.53.58853 > ns.my.net.33476: UDP 12
10:56:34.423458 202.119.192.53.58853 > ns.my.net.33477: UDP 12
10:56:34.423460 202.119.192.53.58853 > ns.my.net.33478: UDP 12
10:56:34.423466 202.119.192.53.58853 > ns.my.net.33479: UDP 12
10:56:34.423490 202.119.192.53.58853 > ns.my.net.33480: UDP 12
```

- a) Distributed Denial of Service
- b) Network Mapping
- c) Back Orifice Scan
- d) Loki Scan

Answer is ‘b’. This is an example of simultaneous traceroutes. All of the traceroutes are destined for the same target host on the protected net. By analyzing the traceroute results obtained from the different sources, the attacker may gain some insight into the protected network’s external topology. (Workbook 2.2, pg. 97)

## 2.3 Intrusion Detection Analysis –Shadow Style

1. The Hex character 'e' is equivalent to which binary number?

- a) 0111
- b) 1010
- c) 1110
- d) 1001

Answer is 'c'. If all bits of a 4 bit chunk are turned on or set to 1 the maximum value will be 15 (8+4+2+1). Counting goes from 0 to 9, 10=a, 11=b, 12=c, 13=d, 14=e, 15=f. (Workbook 2.3, pg. 13)

2. If 8 bits equals a byte, then 4 bits equals a:

- a) nibble
- b) word
- c) bytet
- d) octet

Answer is 'a'. If you consider a byte as two hexadecimal characters, each character will be 4 bits long. A Nibble equals 4 bits. (Workbook 2.3, pg. 13)

3. If the first byte of the IP header has a hex value of '45', what mask value would you use verify the IP header length?

- a) 00001111
- b) 11110000
- c) 11111111
- d) 00000000

Answer is 'a'. The low order bit (nibble value 4) is what we want to verify. A 1 in the mask bit preserves a corresponding value bit, a 0 in the mask bit discards a corresponding value bit. (Workbook 2.3, pg. 15)

4. Which TCPdump filter would check to see if an IP datagram has options set?

- a) `ip[0] & 0x0f > 20`
- b) `ip[0] & 0x0f > 5`
- c) `ip[0] & 0x0e = 20`
- d) `ip[0] & 0x0e < 5`

Answer is 'b'. A way to test whether an IP datagram has options is to test if the IP header length is greater than 5 (this is five 32 bit "words" – or 4 bytes). The filter would become `ip[0]x0f > 5` (Workbook 2.3, pg. 15)

5. Which filter would detect a TCP packet with the SIN-FIN flag set?

- a) `tcp[13] & 0x0f = 0`
- b) `tcp[13] & 0x03 = 3`
- c) `tcp[13] & 0xf0 != 0`
- d) `tcp[13] & 0x0c = 12`

Answer is 'b'. `tcp[13]` indicates the 13<sup>th</sup> byte in the tcp header (counting from 0), this is the location of the tcp flag bits. Once the mask, "`tcp[13] & 0x03 = 3`", is superimposed over the `tcp[13]` byte we can test for a condition equal to the value 3 (0011), SIN-FIN. (Workbook 2.3, pg. 17,18)

6. The following tcpdump sample is an example of which kind of protocol record?

09:32:43.910000 4.3.2.1.1117 > 1.2.3.4.139: S 9887655432:9887655432(0) win 512

- a) UDP
- b) ICMP
- c) IGRP
- d) TCP

Answer is 'd'. Different protocols will have different representations in tcpdump output. One of the first challenges is to identify the protocol (TCP, UDP, ICMP). Flag bits, sequence, and acknowledgement numbers are clues that will identify tcp. (Workbook 2.3, pg. 23)

7. The following filter would log which type of anomalous condition?

(ip[19] =0xff)

- a) Land Attacks
- b) Any broadcast traffic to x.x.x.255
- c) Any traffic with ip options
- d) More fragments bit set

Answer is 'b'. Byte 19 of the IP datagram is the last octet of the destination IP address. The hex value ff indicates that the filter is looking for all 1's set (x.x.x.255) in the last octet of the destination IP. (Workbook 2.3, pg. 80)

8. The following trace is an example of:

4.3.2.1.0 > 1.2.3.4.110: SF 537067520:537067520(0) win 512  
4.3.2.1.0 > 1.2.3.55.110: SF 537067520:537067520(0) win 512  
4.3.2.1.0 > 1.2.3.69.110: SF 537067520:537067520(0) win 512  
4.3.2.1.0 > 1.2.3.24.110: SF 537067520:537067520(0) win 512  
4.3.2.1.0 > 1.2.3.124.110: SF 537067520:537067520(0) win 512

- a) Bootp request
- b) Denial of Service
- c) Pop-3 scan
- d) Imap scan

Answer is 'c'. This is a pop-3 scan; there is also a known vulnerability associated with pop-2. In this scan, we see three well know signatures of anomalous behavior. The three signatures are: Source port 0, SIN-FIN flag set, and unchanging sequence numbers. (Workbook 2.3, pg. 177)

9. The following trace is an example of:

4.3.2.1.34488 > 1.2.3.4.139: FP 0:3(3) ack 1 win 8760 urg 3 (DF)  
4.3.2.1.34488 > 1.2.3.4.139: FP 0:3(3) ack 1 win 8760 urg 3 (DF)  
4.3.2.1.34488 > 1.2.3.4.139: FP 0:3(3) ack 1 win 8760 urg 3 (DF)  
4.3.2.1.34488 > 1.2.3.4.139: FP 0:3(3) ack 1 win 8760 urg 3 (DF)  
4.3.2.1.34488 > 1.2.3.4.139: FP 0:3(3) ack 1 win 8760 urg 3 (DF)

- a) WinNuke
- b) Smurf attack
- c) OS fingerprinting
- d) FIN scan

Answer is 'a'. This attack is also know as the OOBNUKE; the OOB is out of band which refers to a TCP flag bit know as the urgent bit. (Workbook 2.3, pg. 193)

10. Identify the embedded protocol in the following IP header.

4500 0038 d838 0000 7601 1e19 0101 0101

- a) TCP
- b) IP
- c) ICMP
- d) UDP

Answer is 'c'. The embedded protocol can be identified by the value of the 9<sup>th</sup> byte (counting from 0). 1 = ICMP, 6 = TCP, 17 = UDP. (Workbook 2.3, pg. 228)

## 2.4 Network Based Intrusion Detection Analysis

1. The following log was most likely pulled from which device?  
Oct 12 01:04:26 ucc3.edu 45725: 8w5d^]: %SEC-6-IPACCESSLOGP: list  
190 denied tcp 202.159.12.192(2235) -> 172.20.8.233(3128), 1 packet
- a) Raptor Firewall
  - b) Snort IDS
  - c) Cisco Router
  - d) Squid Proxy

Answer is 'c'. The key to identifying Cisco Logs is the reference to the ACL, "IPACCESSLOGP". (Workbook 2.4, pg. 25)

2. When calculating severity an attack blocked at the firewall would indicate?
- a) no risk
  - b) low risk
  - c) moderate risk
  - d) high risk

Answer is 'c'. As you calculate severity try to avoid the mental trap of "the firewall blocked it so there is no risk". There are a number of ways through a firewall such as every system in the building with a modem and also tunneling through http and sending files as email attachments. (Workbook 2.4, pg. 30)

3. CIDF refers to:
- a) Cert Incident Database Format
  - b) Common Intrusion Detection Framework
  - c) Certified Intrusion Detection Fundamentals
  - d) Common Incident Definition Format

Answer is 'b'. The Common Intrusion Detection Framework is a proposed standard to allow interoperability between intrusion detection components. (Workbook 2.4, pg. 33)

4. Traffic Analysis is also known as?
- a) header analysis
  - b) correlation
  - c) sequencing
  - d) all of the above

Answer is 'a'. Traffic analysis of data collected by ID sensors, firewalls, system logs and other sources of information is focused on the fact that a message was passed, not the content. Correlation and profiling are two of the most powerful TA tools. (Workbook 2.4, pg. 52)

5. In CIDF-speak, there are two types of event generators, Push and Pull. Which of the following is true of Pull technology?
- a) The paging feature is often used.
  - b) Email is the preferred method
  - c) Everything is history
  - d) All of the above

Answer is 'c'. A pull-based generator will remain passive until queried by the analyst. In a push-based scenario the analyst receives alerts, usually by email or pager. (Workbook 2.4, pg. 63)

6. A TCP connection is commonly known as?

- a) a three-way handshake
- b) connectionless
- c) resource intensive
- d) a message protocol

Answer is 'a'. TCP requires a three-way handshake between the client and the server before a connection can be established and data transferred. (Workbook 2.4, pg. 81) "When you connect to the Internet the Internet connects to you." *Stephen Northcutt*

7. The following output was most likely generated by?

Active Connections

Proto	Local Address	Foreign Address	State
TCP	hostname:135	0.0.0.0:0	LISTENING
TCP	hostname:135	0.0.0.0:0	LISTENING
TCP	hostname:1033	0.0.0.0:0	LISTENING
TCP	hostname:1034	0.0.0.0:0	LISTENING
TCP	hostname:1035	0.0.0.0:0	LISTENING
TCP	hostname:1044	0.0.0.0:0	LISTENING
TCP	hostname:1046	0.0.0.0:0	LISTENING
TCP	hostname:1761	0.0.0.0:0	LISTENING
TCP	hostname:1762	0.0.0.0:0	LISTENING
TCP	hostname:1027	0.0.0.0:0	LISTENING
TCP	hostname:1027	localhost:1035	ESTABLISHED
TCP	hostname:1032	0.0.0.0:0	LISTENING
TCP	hostname:1032	localhost:1034	ESTABLISHED
TCP	hostname:1034	localhost:1032	ESTABLISHED
TCP	hostname:1035	localhost:1027	ESTABLISHED
TCP	hostname:137	0.0.0.0:0	LISTENING
TCP	hostname:138	0.0.0.0:0	LISTENING

- a) tcpdump -r
- b) netstat -a
- c) snort
- d) windump

Answer is 'b'. The output of netstat -a shows that a connected system may have multiple active connections at one time, each requiring memory. . (Workbook 2.4, pg. 85)

8. What matches the physical address to the IP address?

- a) MAC
- b) DLC
- c) ICMP
- d) ARP

Answer is 'd'. ARP (Address Resolution Protocol) matches the MAC address to the IP address. (Workbook 2.4, pg. 96)

9. Which of the following is not in the TCP header?

- a) Checksum
- b) Window
- c) IP Address
- d) Urg Pointer

Answer is 'c'. The IP Address is not in the TCP header. (Workbook 2.4, pg. 100)

10. Which formula would you use to calculate the severity of an attack?

- a) (Critical + Lethal) – (System + Net Countermeasures)
- b) (Critical + Lethal + System) – (Net Countermeasures)
- c) (Critical - Lethal - System) + (Net Countermeasures)
- d) (Critical - Lethal) + (System - Net Countermeasures)

Answer is 'a'. Severity is best viewed from the target(s) of interest POV. (Workbook 2.4, pg. 118)

## 2.5 Intrusion Detection Workshop

1. A high performance caching proxy server for web clients?

- a) IIS
- b) BIND
- c) Squid
- d) IRC

Answer is 'c'. Squid is a high performance caching proxy server for web clients. It supports FTP, gopher, and HTTP protocols. (Workbook 2.5, pg. 153)

2. The primary key to maintaining situational awareness and to help scope the size and intensity of an attack is?

- a) Correlation
- b) TCP Wrappers
- c) Source enumeration
- d) Attack forensics

Answer is 'a'. Within a site, correlating ID sensors and system logs is a powerful tool. As we correlate logs from multiple sources we get a bigger and better picture of what is happening. . (Workbook 2.5, pg. 157)

3. Which system trace correlates with the snort trace?

\*\*\*\*\*

**TRACE 1** - Feb 21 18:59:03 dns3 rcpbind: refused connect from 159.226.8.2 to dump()

**TRACE 2** - Feb 21 18:59:09 dns3 rcpbind: refused connect from 159.226.8.2 to getport(1000d)

**TRACE 3** - Feb 21 19:50:15 dns3 rcpbind: refused connect from 159.226.8.2 to getport(1000d)

\*\*\*\*\*

### SNORT LOG

```
[**] RCP – portmap-request-cmsd [**]  
02/21-18:59:09.518264 159.226.8.2:33553 -> x.x.x.z:111  
UDP TTL:242 TOS:0x0 ID:60474 DF  
Len: 64  
38 B4 8B 69 00 00 00 00 00 00 02 00 01 86 A0 8..i.....  
00 00 00 02 00 00 00 03 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 01 86 E4 00 00 00 04 .....  
00 00 00 11 00 00 00 00 .....
```

- a) Trace 1
- b) Trace 2
- c) Trace 3
- d) All of the above

Answer is 'b'. If nothing else match up the time stamps. (Workbook 2.5, pg. 175)

4. Which statement about CIDF is true?
- a) Created by IANA
  - b) Used by Cisco IOS
  - c) Has a language for describing attacks
  - d) All of the above

Answer is 'c'. The Common Intrusion Detection Framework (CIDF) has a language for describing attacks, which is crucial for correlation. (Workbook 2.5, pg. 179)

5. The following trace is an example of:
- ```
host.2822 > fw.53: S 37007:37007(0) win 512
fw.53 > host.2822: S 12000:12000(0) ack 37008 win 32768 (DF)
host.2822 > fw.53: . ack 1 win 16060 (DF)
```
- a) crafted packet
  - b) window size negotiation
  - c) DNS query
  - d) Zone transfer

Answer is 'b'. Window size can be negotiated during the TCP conversation. This is an example of normal traffic. (Workbook 2.5, pg. 208)

6. FTP has a high number of?
- a) states
  - b) ports
  - c) exploits
  - d) all of the above

Answer is 'a'. FTP is rich in patterns caused by the large number of states in the protocol, as well as some curious implementations. (Workbook 2.5, pg. 215)

7. As a general rule, SYN floods are considered to be?
- a) High Risk
  - b) Low Risk
  - c) Evidence of a Trojan
  - d) A threat to Windows hosts

Answer is 'b'. Most commercial intrusion detection systems false positive on SYN floods so often that you have to set their counters to a very high number. SYN floods are becoming less of a problem. (Workbook 2.5, pg. 242)

8. The following trace is an example of:
- ```
16:34:59 sm.org 137 > 192.168.130.55 137:udp
16:34:59 sm.org 137 > 192.168.108.171 137:udp
16:34:59 sm.org 137 > 192.168.162.231 137:udp
16:34:59 sm.org 137 > 192.168.14.209 137:udp
16:34:59 sm.org 137 > 192.168.145.45 137:udp
```
- a) DoS
  - b) WinNuke
  - c) Trojan Trolling
  - d) Information Gathering

Answer is 'd'. Port 137 provides a lot of information to an attacker; if at all possible this should be blocked. (Workbook 2.5, pg. 293)

```
netbios-ns 137/tcp NETBIOS Name Service
netbios-ns 137/udp NETBIOS Name Service
```

The Well-Known Ports are assigned by the IANA and on most systems can only be used by system (or root) processes or by programs executed by privileged users.

9. In a Windows network, the following command would:

```
net use \\172.20.244.164\IPC$ "" /USER:""
```

- a) blue screen un-patched Windows machines
- b) connect to the Windows 9x registry directory
- c) establish a null session to Windows NT machine
- d) establish a null session to any Windows machine

Answer is 'c'. If a null session or anonymous login is allowed, it is possible to collect a tremendous amount of information about the system including the names of the Local Administrators. (Workbook 2.5, pg. 298) Windows NT has a feature that allows non-authenticated users to enumerate users on a Windows NT domain. If you do not want this functionality, set the following in the Registry:

Hive: HKEY\_LOCAL\_MACHINE\SYSTEM

Key: CurrentControlSet\Control\LSA

Name: RestrictAnonymous

Type: REG\_DWORD

Value: 1

(Microsoft Internet Information Server 4.0 Security Checklist)

10. The following trace is an example of:

```
00:59:17 m4trix> 172.20.179.41: icmp: echo reply
00:59:17 router> m4trix: icmp: host unreachable
02:11:50 m4trix> 172.20.54.94: icmp: echo reply
02:11:50 router> m4trix: icmp: host unreachable
02:55:14 m4trix> 172.20.135.88: icmp: echo reply
03:45:36 m4trix> 172.20.54.116: icmp: echo reply
03:45:36 router> m4trix: icmp: host unreachable
04:09:11 m4trix> 172.20.9.57: icmp: echo reply
04:49:12 m4trix> 172.20.205.83: icmp: echo reply
05:00:17 m4trix> 172.20.134.25: icmp: echo reply
```

- a) inverse scan
- b) spoofed IP address
- c) fragmented ICMP
- d) DoS

Answer is 'a'. The router will reply to hosts that do not exist allowing an attacker to map out the network behind the router. Consider the slow scan rate: it will probably evade scan detection software. Also, most sites do not block incoming echo replies. . (Workbook 2.5, pg. 307)



## GIAC Practical for Martin Seery

NOTE: Due to issues surrounding the legality of providing information about the behavior of another companies IP addresses, both the source and destination IP addresses have been sanitized. The monitored networks IP addresses have the first two octets sanitized. IP addresses categorized as “intruder” have the last two octets of the IP address sanitized.

Example:

My subnet = x.x.my.net

Intruder = not.me.x.x

This issue was discussed with and cleared with Stephen Northcutt via email correspondence. Questions regarding the validity of this claim Should be brought to the attention of Stephen Northcutt.

### **Detect 1**

*Time Stamp in all traces is GMT*

#### Snort Alert:

```
[**] spp_portsan: PORTSCAN DETECTED from x.x.9.1 [**]
05/22-20:45:57.689299
[**] spp_portsan: portsan status from x.x.9.1: 4 connections across 2
hosts: TCP(0), UDP(4) [**]
05/22-20:46:06.445113
[**] spp_portsan: portsan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:46:16.445198
[**] spp_portsan: portsan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:46:26.444554
[**] spp_portsan: portsan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:46:36.444277
[**] spp_portsan: portsan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:46:46.444430
[**] spp_portsan: portsan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:46:56.443808
[**] spp_portsan: portsan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:47:06.443519
[**] spp_portsan: portsan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:47:16.443705
[**] spp_portsan: portsan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:47:26.442978
[**] spp_portsan: portsan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:47:36.442757
[**] spp_portsan: portsan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:47:46.442909
[**] spp_portsan: portsan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:47:56.442251
```

Prepared by Martin J Seery

```
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:48:06.442029
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:48:16.442156
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:48:26.441494
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:48:36.441307
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:48:46.441403
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:48:56.440733
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:49:06.440480
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:49:16.440633
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:49:26.440054
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:49:36.439721
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:49:46.439877
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:49:56.439213
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:50:06.438968
[**] spp_portscan: portscan status from x.x.9.1: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/22-20:50:16.439160
[**] spp_portscan: End of portscan from x.x.9.1 [**]
05/22-20:50:24.008197
```

### Supporting Data from Snort PortScan Log:

```
May 22 20:46:06 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:46:16 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:46:26 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:46:36 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:46:46 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:46:56 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:47:06 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:47:16 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:47:26 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:47:36 x.x.9.1:1025 -> x.x.9.31:2071 UDP
```

```

May 22 20:47:46 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:47:56 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:48:06 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:48:16 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:48:26 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:48:36 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:48:46 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:48:56 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:49:06 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:49:16 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:49:26 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:49:36 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:49:46 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:49:56 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:50:06 x.x.9.1:1025 -> x.x.9.31:2071 UDP
May 22 20:50:16 x.x.9.1:1025 -> x.x.9.31:2071 UDP

```

**Supporting Data from Total log:**

```

05/22-20:46:06.444757 x.x.9.1:1025 -> x.x.9.31:2071 UDP
UDP TTL:64 TOS:0x0 ID:19947
Len: 101
00 59 FC 67 01 01 02 02 01 04 03 06 2B 45 72 6F .Y.g.....+Ero
6C 73 04 0C 39 2C 32 31 35 34 33 37 38 31 31 31 ls..9,215437xxxx
05 00 06 01 01 07 04 00 00 01 91 08 01 00 09 01 .....
01 0A 01 00 16 01 00 17 00 18 00 19 00 1A 01 00 .....
1B 04 00 00 00 00 1C 01 FF 1D 01 00 1E 01 FF 28 ..... (
06 2B 45 72 6F 6C 73 29 00 2A 00 2B 00 .+Erols).*+.

```

\*\*\*\*\* Above data repeats to match time stamps of previous logs. \*\*\*\*\*

**1. Source of trace**

Small subnet, using a 3com ISDN Office Connect LAN Modem (OCLM) Router as a gateway to the Internet.

**2. Detect was generated by:**

SNORT version 1.6 using 05172kany.rules.

**3. Probability the source address was spoofed:**

The source IP in this trace was verified to be from the 3com OCLM (Office Connect LAN Modem), that was being used as a gateway to the Internet. The IP was not spoofed.

**4. Description of attack:**

spp\_portscan: PORTSCAN DETECTED - UDP packets were being broadcast from the router to subnet x.x.9.0/27, causing snort to generate an spp\_portscan alert. This was the first alert generated by Snort (very exciting). In order to confirm what was happening the total log output, with UDP data, was analyzed. The data portion of the UDP packet contained information about the ISP. This appears to be a broadcast of service from the OCLM. (It's a very noisy unit.)

**5. Attack mechanism:**

The attack ended up being a false positive. The subnet being monitored is x.x.9.0/27. A 27-bit prefix represents a contiguous block of 25 (32) individual IP addresses. However, since the all-0s and all-1s host addresses cannot be allocated, there are 30 (25 -2) assignable host addresses on the subnet. The OCLM router was broadcasting service to network x.x.9.0/27 using the x.x.9.31 broadcast address for this subnet. During the initial investigation it was discovered that telnet was available for this router even though this feature was not in the vendor documentation. "OCLM SECRETS - Bubba's Guide To The Telnet Interface" is an unofficial summary of undocumented and unsupported commands available in the Telnet Interface of the 3Com Office Connect LAN Modem (OCLM). It

proved to be a valuable resource in understanding the configuration limitations of the OCLM and is included in the Appendix of this document.

#### 6. Correlation's:

This is a legitimate broadcast. No correlation was needed.

#### 7. Evidence of active targeting:

The OCLM router was targeting the entire x.x.9.0/27 network for broadcast of service.

#### 8. Severity:

Severity = (criticality + lethality) – (system + net) = (3+2) – (5+2) = -2

Criticality = 3: Target was entire subnet.

Lethality = 2: Broadcast to subnet.

System = 5 All systems on subnet were modern OS/hardware, with SPs and patches applied.

Net = 2 Router was wide open with limited logging capabilities. Router was secured with a strong password consisting of upper and lower case characters, numbers, and special characters.

#### 9. Defensive recommendation:

The attack is a false positive. No defensive measures required.

#### 10. Multiple choice test question:

Identifying the source of false positives can be used for?

- a) Severity analysis
- b) identifying Trojan trolling
- c) network analysis
- d) identifying DoS attacks

Answer is 'c'. Identifying the source of false positives can help resolve anomalous behavior on your network which could free up valuable bandwidth.

### Detect 2

*Time Stamp in all traces is GMT*

#### Snort Alert:

```
[**] BACKDOOR SIGNATURE - NetMetro File List [**]
05/23-15:48:09.777455 209.166.x.x:80 -> x.x.9.10:5032
TCP TTL:250 TOS:0x0 ID:30790 DF
*****PA* Seq: 0x8B918BF Ack: 0x28855 Win: 0xFAF0
```

```
[**] BACKDOOR SIGNATURE - NetMetro File List [**]
05/23-15:48:13.659312 209.166.x.x:80 -> x.x.9.10:5032
TCP TTL:250 TOS:0x0 ID:30800 DF
*****PA* Seq: 0x8B91D2A Ack: 0x28855 Win: 0xFAF0
```

```
[**] BACKDOOR SIGNATURE - NetMetro File List [**]
05/23-15:48:19.732697 209.166.x.x:80 -> x.x.9.10:5032
TCP TTL:250 TOS:0x0 ID:30801 DF
*****PA* Seq: 0x8B92492 Ack: 0x28855 Win: 0xFAF0
```

#### Supporting Data from Total log:

```
05/23-15:47:40.025008 x.x.9.10:5032 -> 209.166.x.x:80
TCP TTL:128 TOS:0x10 ID:64684 DF
**S***** Seq: 0x286C4 Ack: 0x0 Win: 0x2000
TCP Options => MSS: 1460
```

SANS GIAC – GIAC Practical

4A 46

JF

05/23-15:47:40.070583 209.166.x.x:80 -> x.x.9.10:5032  
TCP TTL:250 TOS:0x0 ID:30788 DF  
\*\*S\*\*\*A\* Seq: 0x8B918BE Ack: 0x286C5 Win: 0xFAF0  
TCP Options => MSS: 1460  
00 00 ..

05/23-15:47:40.070897 x.x.9.10:5032 -> 209.166.x.x:80  
TCP TTL:128 TOS:0x10 ID:65196 DF  
\*\*\*\*\*A\* Seq: 0x286C5 Ack: 0x8B918BF Win: 0x2238  
02 04 05 B4 4A 46 ....JF

05/23-15:48:09.777455 209.166.x.x:80 -> x.x.9.10:5032  
TCP TTL:250 TOS:0x0 ID:30790 DF  
\*\*\*\*PA\* Seq: 0x8B918BF Ack: 0x28855 Win: 0xFAF0  
48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D HTTP/1.1 200 OK.  
0A 44 61 74 65 3A 20 54 75 65 2C 20 32 33 20 4D .Date: Tue, 23 M  
61 79 20 32 30 30 30 20 31 34 3A 35 38 3A 32 34 ay 2000 14:58:24  
20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 41 70 GMT..Server: Ap  
61 63 68 65 2F 31 2E 33 2E 36 0D 0A 43 6F 6E 74 ache/1.3.6..Cont  
65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F 68 ent-Type: text/h  
74 6D 6C 0D 0A 41 67 65 3A 20 37 0D 0A 43 6F 6E tml..Age: 7..Con  
6E 65 63 74 69 6F 6E 3A 20 63 6C 6F 73 65 0D 0A nnection: close..  
56 69 61 3A 20 48 54 54 50 2F 31 2E 31 20 63 6C Via: HTTP/1.1 cl  
75 73 74 65 72 2E 6C 6E 68 2E 6D 64 20 28 54 72 uster.lnh.md (Tr  
61 66 66 69 63 2D 53 65 72 76 65 72 2F 33 2E 30 affic-Server/3.0  
2E 33 20 5B 75 53 63 4D 73 53 66 57 70 53 65 4E .3 [uScMsSfWpSeN  
3A 74 20 63 20 4D 69 20 70 20 73 53 5D 29 0D 0A :t c Mi p sS)]..  
0D 0A 3C 68 74 6D 6C 3E 0A 3C 68 65 61 64 3E 0A ..<html>.<head>.  
3C 74 69 74 6C 65 3E 4C 69 76 65 20 48 61 63 6B <title>Live Hack  
20 41 74 74 65 6D 70 74 73 20 41 67 61 69 6E 73 Attempts Agains  
74 20 54 68 65 20 41 6E 74 69 4F 6E 6C 69 6E 65 t The AntiOnline  
20 4E 65 74 77 6F 72 6B 3C 2F 74 69 74 6C 65 3E Network</title>  
0A 3C 2F 68 65 61 64 3E 0A 3C 62 6F 64 79 20 62 .</head>.<body b  
67 63 6F 6C 6F 72 3D 22 23 66 66 66 66 66 66 22 gcolor="#ffffff"  
20 6C 69 6E 6B 3D 22 23 30 30 30 30 39 39 22 20 link="#000099"  
61 6C 69 6E 6B 3D 22 23 30 30 30 30 39 39 22 20 alink="#000099"  
76 6C 69 6E 6B 3D 22 23 30 30 30 30 39 39 22 3E vlink="#000099">  
0A 3C 21 2D 2D 20 54 6F 70 20 41 64 2C 20 4C 6F .<!-- Top Ad, Lo  
67 6F 2C 20 41 6E 64 20 53 65 61 72 63 68 20 54 go, And Search T  
61 62 6C 65 20 53 74 61 72 74 20 2D 2D 3E 0A 3C able Start -->.<  
74 61 62 6C 65 20 77 69 64 74 68 3D 36 30 30 20 table width=600  
63 65 6C 6C 73 70 61 63 69 6E 67 3D 30 20 63 65 cellpadding=0 ce  
6C 6C 70 61 64 64 69 6E 67 3D 30 20 62 6F 72 64 llpadding=0 bord  
65 72 3D 30 3E 0A 3C 74 72 20 76 61 6C 69 67 6E er=0>.<tr valign  
3D 22 62 6F 74 74 6F 6D 22 3E 0A 3C 74 64 20 77 ="bottom">.<td w  
69 64 74 68 3D 22 36 30 30 22 20 76 61 6C 69 67 idth="600" valig  
6E 3D 22 62 6F 74 74 6F 6D 22 20 63 6F 6C 73 70 n="bottom" colsp  
61 6E 3D 32 3E 0A 3C 21 2D 2D 20 44 79 6E 61 6D an=2>.<!-- Dynam  
69 63 20 49 6E 73 65 72 74 69 6F 6E 20 4F 66 20 ic Insertion Of  
41 64 20 42 61 6E 6E 65 72 20 2D 2D 3E 0A 3C 41 Ad Banner -->.<A  
20 48 52 45 46 3D 22 68 74 74 70 3A 2F 2F 77 77 HREF="http://ww  
77 2E 41 6E 74 69 4F 6E 6C 69 6E 65 2E 63 6F 6D w.AntiOnline.com  
2F 63 67 69 2D 62 69 6E 2F 61 64 73 2F 34 36 38 /cgi-bin/ads/468  
78 36 30 2D 6E 6F 73 73 69 2E 70 6C 3F 62 61 6E x60-nossi.pl?ban  
6E 65 72 3D 4E 6F 6E 53 53 49 3B 70 61 67 65 3D ner=NonSSI;page=

```

39 33 22 3E 3C 49 4D 47 20 53 52 43 3D 22 68 74 93"><IMG SRC="ht
74 70 3A 2F 2F 77 77 77 2E 41 6E 74 69 4F 6E 6C tp://www.AntiOnl
69 6E 65 2E 63 6F 6D 2F 63 67 69 2D 62 69 6E 2F ine.com/cgi-bin/
61 64 73 2F 34 36 38 78 36 30 2D 6E 6F 73 73 69 ads/468x60-nossi
2E 70 6C 3F 70 61 67 65 3D 39 33 22 20 68 65 69 .pl?page=93" hei
67 68 74 3D 36 30 20 77 69 64 74 68 3D 34 36 38 ght=60 width=468
20 62 6F 72 64 65 72 3D 30 3E 3C 2F 41 3E 0A 3C border=0></A>.<
21 2D 2D 20 45 6E 64 20 41 64 20 42 61 6E 6E 65 !-- End Ad Banne
72 20 43 6F 64 65 20 2D 2D 3E 0A 3C 2F 74 64 3E r Code -->.</td>
0A 3C 2F 74 72 3E 0A 3C 74 72 20 76 61 6C 69 67 .</tr>.<tr valig
6E 3D 22 62 6F 74 74 6F 6D 22 3E 0A 3C 74 64 20 n="bottom">.<td
77 69 64 74 68 3D 36 30 30 20 63 6F 6C 73 70 61 width=600 colspa
6E 3D 32 20 76 61 6C 69 67 6E 3D 22 62 6F 74 74 n=2 valign="bott
6F 6D 22 3E 0A 3C 69 6D 67 20 73 72 63 3D 22 2F om">..</td>.</tr>.<
3C 74 72 20 76 61 6C 69 67 6E 3D 22 62 6F 74 74 <tr valign="bott
6F 6D 22 3E 0A 3C 74 64 20 76 61 6C 69 67 6E 3D om">.<td valign=
22 62 6F 74 74 6F 6D 22 20 61 6C 69 67 6E 3D 22 "bottom" align="
6C 65 66 74 22 20 77 69 64 74 68 3D 22 32 39 34 left" width="294
22 3E 0A 3C 69 6D 67 20 73 72 63 3D 22 2F 69 6D ">. 209.166.x.x:80
TCP TTL:128 TOS:0x10 ID:64684 DF
**S***** Seq: 0x286C4 Ack: 0x0 Win: 0x2000
TCP Options => MSS: 1460
4A 46 JF

05/23-15:47:40.070583 209.166.x.x:80 -> x.x.9.10:5032
TCP TTL:250 TOS:0x0 ID:30788 DF
**S***A* Seq: 0x8B918BE Ack: 0x286C5 Win: 0xFAF0
TCP Options => MSS: 1460
00 00 ..

05/23-15:47:40.070897 x.x.9.10:5032 -> 209.166.x.x:80
TCP TTL:128 TOS:0x10 ID:65196 DF
*****A* Seq: 0x286C5 Ack: 0x8B918BF Win: 0x2238
02 04 05 B4 4A 46 ....JF
```

- a) Null Session
- b) TCP Hijack
- c) Three-way handshake
- d) DoS

Answer is 'c'. This is a good example of a three-way handshake.

### Detect 3

*Time Stamp in all traces is GMT*

### Snort Alert:

```
[**] spp_portscan: PORTSCAN DETECTED from 160.12.x.x [**]
05/23-19:38:28.682971
[**] SCAN-SYN FIN [**]
05/23-19:38:28.682716 160.12.x.x:111 -> x.x.9.7:111
TCP TTL:19 TOS:0x0 ID:39426
f**SF**** Seq: 0x47F41B5B Ack: 0x2F9494D Win: 0x404

[**] SCAN-SYN FIN [**]
05/23-19:38:28.762276 160.12.x.x:111 -> x.x.9.10:111
TCP TTL:19 TOS:0x0 ID:39426
**SF**** Seq: 0x47F41B5B Ack: 0x2F9494D Win: 0x404
```

```
[**] spp_portscan: portscan status from 160.12.x.x: 2 connections across 2
hosts: TCP(2), UDP(0) STEALTH [**]
05/23-19:38:35.299447
[**] spp_portscan: End of portscan from 160.12.x.x [**]
05/23-19:38:45.298671
```

### Supporting Data from Snort PortScan Log:

```
May 23 19:38:28 160.12.x.x:111 -> x.x.9.7:111 SYNFIN **SF****
May 23 19:38:28 160.12.x.x:111 -> x.x.9.10:111 SYNFIN **SF****
```

### Supporting Data from Total log:

```
05/23-19:38:28.682716 160.12.x.x:111 -> x.x.9.7:111
TCP TTL:19 TOS:0x0 ID:39426
**SF**** Seq: 0x47F41B5B Ack: 0x2F9494D Win: 0x404
00 00 00 00 00 00 .....

05/23-19:38:28.682943 x.x.9.7:111 -> 160.12.x.x:111
TCP TTL:128 TOS:0x0 ID:31009
****R*A* Seq: 0x0 Ack: 0x47F41B5D Win: 0x0
00 00 00 00 00 00 .....

05/23-19:38:28.762276 160.12.x.x:111 -> x.x.9.10:111
TCP TTL:19 TOS:0x0 ID:39426
**SF**** Seq: 0x47F41B5B Ack: 0x2F9494D Win: 0x404
00 00 00 00 00 00 .....

05/23-19:38:28.762473 x.x.9.10:111 -> 160.12.x.x:111
TCP TTL:128 TOS:0x0 ID:11491
****R*A* Seq: 0x0 Ack: 0x47F41B5D Win: 0x0
00 00 20 45 4A 46 .. EJF
```

#### 1. Source of trace

Small subnet using a 3com ISDN Office Connect LAN Modem (OCLM) Router as a gateway to the Internet.

#### 2. Detect was generated by:

SNORT version 1.6 using 05172kany.rules

#### 3. Probability the source address was spoofed

This is classic information gathering. IP address was most likely not spoofed.

#### 4. Description of attack:

SCAN-SYN FIN – Triggers when a series of TCP packets with both the SYN and FIN flags set have been sent to the same destination port on a number of different hosts.

Illegal flag set, SIN-FIN, looking for port mapper, port 111. The sequence numbers of the packet sent to x.x.9.7 and x.x.9.10 are the same (Seq: 0x47F41B5B). This is evidence of use of a port scanning tool.

#### 5. Attack mechanism:

Illegal flag set, SIN-FIN, looking for port mapper.

```
sunrpc      111/tcp      SUN Remote Procedure Call
sunrpc      111/udp      SUN Remote Procedure Call
```

This is definitely intrusive information gathering using a port scanning tool. This intruder is looking for UNIX machines, using a SYN-FIN probe to port 111. First machine probed is 'x.x.9.7', this is a Win98 machine. It responds with a RST-ACK and no data. The second machine probed, x.x.9.10, is a WinNT Workstation running



Back Officer Friendly with a port 111 routine running. Host x.x.9.10 responds with a RST-ACK and data, emulating an active UNIX server.

Servers register themselves with the port mapper using RPC calls, and clients query the port mapper using RPC calls. If TCP is being used, (as in this case) the client does an active open to the server's TCP port number. (TCP/IP Illustrated Volume 1, pg. 466)

#### 6. Correlation's:

The CERT/CC Current Activity web page indicates that this type of probe is common and active in the wild.

sunrpc 111/tcp 111/udp [CA-99-16](#), Buffer Overflow in Sun Solstice AdminSuite Daemon sadmind

[CA-99-12](#), Buffer overflow in amd

[CA-99-08](#), Buffer overflow in rpc.cmsd

[CA-99-05](#), Vulnerability in statd exposes vulnerability in automountd

[CA-98-12](#), Remotely Exploitable Buffer Overflow Vulnerability in mountd

[CA-98-11](#), Vulnerability in ToolTalk RPC service

#### 7. Evidence of active targeting:

Definitely looking for UNIX hosts.

#### 8. Severity:

Severity = (criticality + lethality) – (system + net) = (3+5) – (5+2) = 0

Criticality = 3: Targets were workstations. However, I bumped it to a three due to the fact that if there were servers on this subnet they would have been targeted.

Lethality = 5: If RPC was running, this could have turned into a lethal attack.

System = 5 Both machines had all current Hot fixes applied and tightened security. Port Mapper was not actually running.

Net = 2 Router was wide open with limited logging capabilities. Router was secured with a strong password consisting of upper and lower case characters, numbers, and special characters.

#### 9. Defensive recommendation:

Both machines had all of the latest security patches applied. The NT machine had security tightened using the "Microsoft Internet Information Server 4.0 Security Checklist" as a guide. The Windows 98 machine had file and print sharing disabled. Both machines contained nothing of value and the subnet was isolated.

The OCLM router had NAT disabled, allowed NetBIOS traffic to be passed, and was in an "always on" state. This is not a desired configuration. After the testing of snort was completed, the detects were collected and the router settings were securely set. If this were a production environment TCP/UDP traffic to port 111 and the IP address of the intruder would have been blocked.

#### Multiple choice test question:

In the trace below, which type of Operating System is being targeted?

```
05/23-19:38:28.682716 160.12.x.x:111 -> x.x.9.7:111
TCP TTL:19 TOS:0x0 ID:39426
**SF**** Seq: 0x47F41B5B Ack: 0x2F9494D Win: 0x404
00 00 00 00 00 00 .....
```

- a) Windows NT
- b) UNIX
- c) OS/2
- d) Novell Netware

Answer is 'b'. UNIX machines can be easily identified by port mapper, port 111.

#### Detect 4

Time Stamp in all traces is GMT

Snort Alert:

```
[**] spp_portscan: PORTSCAN DETECTED from 208.58.x.x [**]
05/24-16:41:29.592282
[**] spp_portscan: portscan status from 208.58.x.x: 4 connections across 2
hosts: TCP(0), UDP(4) [**]
05/24-16:41:44.327083
[**] spp_portscan: End of portscan from 208.58.x.x [**]
05/24-16:41:54.327151
[**] spp_portscan: PORTSCAN DETECTED from 208.58.x.x [**]
05/24-16:42:20.187214
[**] spp_portscan: portscan status from 208.58.x.x: 4 connections across 2
hosts: TCP(0), UDP(4) [**]
05/24-16:42:34.325812
[**] spp_portscan: End of portscan from 208.58.x.x [**]
05/24-16:42:44.325460

[**] spp_portscan: PORTSCAN DETECTED from 208.58.x.x [**]
05/26-15:48:02.598048
[**] spp_portscan: portscan status from 208.58.x.x: 4 connections across 2
hosts: TCP(0), UDP(4) [**]
05/26-15:48:10.502808
[**] spp_portscan: End of portscan from 208.58.x.x [**]
05/26-15:48:23.637320
```

Supporting Data from Snort PortScan Log:

```
May 24 16:41:29 208.58.x.x:1646 -> x.x.9.7:5632 UDP
May 24 16:41:29 208.58.x.x:1646 -> x.x.9.7:22 UDP
May 24 16:41:29 208.58.x.x:1646 -> x.x.9.10:5632 UDP
May 24 16:41:29 208.58.x.x:1646 -> x.x.9.10:22 UDP
May 24 16:42:20 208.58.x.x:1648 -> x.x.9.7:5632 UDP
May 24 16:42:20 208.58.x.x:1648 -> x.x.9.7:22 UDP
May 24 16:42:20 208.58.x.x:1648 -> x.x.9.10:5632 UDP
May 24 16:42:20 208.58.x.x:1648 -> x.x.9.10:22 UDP

May 26 15:48:02 208.58.x.x:1277 -> x.x.9.7:5632 UDP
May 26 15:48:02 208.58.x.x:1277 -> x.x.9.7:22 UDP
May 26 15:48:02 208.58.x.x:1277 -> x.x.9.10:5632 UDP
May 26 15:48:02 208.58.x.x:1277 -> x.x.9.10:22 UDP
```

Supporting Data from Total log:

```
05/24-16:41:29.538639 208.58.x.x:1646 -> x.x.9.7:5632
UDP TTL:121 TOS:0x0 ID:40084
Len: 10
4E 51 00 00 00 00 00 00 00 00 00 00 00 00 00 00 NQ.....
00 00 ..

05/24-16:41:29.538804 x.x.9.7 -> 208.58.x.x
ICMP TTL:128 TOS:0x0 ID:32801
DESTINATION UNREACHABLE: PORT UNREACHABLE
00 00 00 00 45 00 00 1E 9C 94 00 00 79 11 F1 FA ....E.....y...
D0 3A 09 C4 D0 3A 09 07 06 6E 16 00 00 0A E1 DA .....:.....n.....
```

SANS GIAC – GIAC Practical

05/24-16:41:29.540246 208.58.x.x:1646 -> x.x.9.7:22  
UDP TTL:121 TOS:0x0 ID:40340  
Len: 10  
4E 51 00 00 00 00 00 00 00 00 00 00 00 00 00 00 NQ.....  
00 00 ..

05/24-16:41:29.540386 x.x.9.7 -> 208.58.x.x  
ICMP TTL:128 TOS:0x0 ID:33057  
DESTINATION UNREACHABLE: PORT UNREACHABLE  
00 00 00 00 45 00 00 1E 9D 94 00 00 79 11 F0 FA ....E.....y...  
D0 3A 09 C4 D0 3A 09 07 06 6E 00 16 00 0A F7 C4 .....n.....

05/24-16:41:29.590048 208.58.x.x:1646 -> x.x.9.10:5632  
UDP TTL:121 TOS:0x0 ID:41620  
Len: 10  
4E 51 00 00 00 00 00 00 00 00 00 00 00 00 00 00 NQ.....  
00 00 ..

05/24-16:41:29.590277 x.x.9.10 -> 208.58.x.x  
ICMP TTL:128 TOS:0x0 ID:28163  
DESTINATION UNREACHABLE: PORT UNREACHABLE  
00 00 00 00 45 00 00 1E A2 94 00 00 79 11 EB F7 ....E.....y...  
D0 3A 09 C4 D0 3A 09 0A 06 6E 16 00 00 0A E1 D7 .....n.....

05/24-16:41:29.592044 208.58.x.x:1646 -> x.x.9.10:22  
UDP TTL:121 TOS:0x0 ID:41876  
Len: 10  
4E 51 00 00 00 00 00 00 00 00 00 00 00 00 00 00 NQ.....  
00 00 ..

05/24-16:41:29.592196 x.x.9.10 -> 208.58.x.x  
ICMP TTL:128 TOS:0x0 ID:28419  
DESTINATION UNREACHABLE: PORT UNREACHABLE  
00 00 00 00 45 00 00 1E A3 94 00 00 79 11 EA F7 ....E.....y...  
D0 3A 09 C4 D0 3A 09 0A 06 6E 00 16 00 0A F7 C1 .....n.....

05/24-16:42:20.127894 208.58.x.x:1648 -> x.x.9.7:5632  
UDP TTL:121 TOS:0x0 ID:56728  
Len: 10  
4E 51 00 00 00 00 00 00 00 00 00 00 00 00 00 00 NQ.....  
00 00 ..

05/24-16:42:20.128087 x.x.9.7 -> 208.58.x.x  
ICMP TTL:128 TOS:0x0 ID:33313  
DESTINATION UNREACHABLE: PORT UNREACHABLE  
00 00 00 00 45 00 00 1E DD 98 00 00 79 11 B0 F6 ....E.....y...  
D0 3A 09 C4 D0 3A 09 07 06 70 16 00 00 0A E1 D8 .....p.....

05/24-16:42:20.129542 208.58.x.x:1648 -> x.x.9.7:22  
UDP TTL:121 TOS:0x0 ID:56984  
Len: 10  
4E 51 00 00 00 00 00 00 00 00 00 00 00 00 00 00 NQ.....  
00 00 ..

05/24-16:42:20.129696 x.x.9.7 -> 208.58.x.x  
ICMP TTL:128 TOS:0x0 ID:33569  
DESTINATION UNREACHABLE: PORT UNREACHABLE

```
00 00 00 00 45 00 00 1E DE 98 00 00 79 11 AF F6 ....E.....y...
D0 3A 09 C4 D0 3A 09 07 06 70 00 16 00 0A F7 C2 .....:..p.....
```

```
05/24-16:42:20.185044 208.58.x.x:1648 -> x.x.9.10:5632
UDP TTL:121 TOS:0x0 ID:58264
```

```
Len: 10
4E 51 00 00 00 00 00 00 00 00 00 00 00 00 00 00 NQ.....
00 00 ..
```

```
05/24-16:42:20.185266 x.x.9.10 -> 208.58.x.x
ICMP TTL:128 TOS:0x0 ID:28675
```

```
DESTINATION UNREACHABLE: PORT UNREACHABLE
00 00 00 00 45 00 00 1E E3 98 00 00 79 11 AA F3 ....E.....y...
D0 3A 09 C4 D0 3A 09 0A 06 70 16 00 00 0A E1 D5 .....:..p.....
```

```
05/24-16:42:20.187066 208.58.x.x:1648 -> x.x.9.10:22
UDP TTL:121 TOS:0x0 ID:58520
```

```
Len: 10
4E 51 00 00 00 00 00 00 00 00 00 00 00 00 00 00 NQ.....
00 00 ..
```

```
05/24-16:42:20.187270 x.x.9.10 -> 208.58.x.x
ICMP TTL:128 TOS:0x0 ID:28931
```

```
DESTINATION UNREACHABLE: PORT UNREACHABLE
00 00 00 00 45 00 00 1E E4 98 00 00 79 11 A9 F3 ....E.....y...
D0 3A 09 C4 D0 3A 09 0A 06 70 00 16 00 0A F7 BF .....:..p.....
```

*The detects from May 26 15:48:02 were same as above.*

**1. Source of trace**

Small subnet using a 3com ISDN Office Connect LAN Modem (OCLM) Router as a gateway to the Internet.

**2. Detect was generated by:**

SNORT version 1.6 using 05172kany.rules

**3. Probability the source address was spoofed**

Information gathering IP address was most likely not spoofed. The intruder is most likely looking for PCAnywhere (UDP 22) and (UDP 5632). It was interesting to read about this particular trace in the SANS GIAC 2.5 Intrusion Detection Workshop book pg. 260.

**4. Description of attack:**

spp\_portscan: PORTSCAN DETECTED - Probe for PCAnywhere, UDP 22 and UDP 5632.

**5. Attack mechanism:**

This was an attempt to discover the PCAnywhere remote logon program. If this program was running and detected by the intruder, he/she could have run a structured attack against the host and possibly gain Admin access. Looking at the times of the probes it appears as though the attacker was a novice. The first probe happened on May 24 16:41:29 GMT; the second almost a minute later, May 24 16:42:20 GMT; and the third happened two days later at approximately the same time of day, May 26 15:48:02 GMT. In all cases the intruder received an ICMP reply stating "PORT UNREACHABLE", or sorry, "no PCAnywhere here." A seasoned cracker, like a seasoned burglar, wouldn't try to turn a doorknob that he/she knew wasn't there.

**6. Correlation's:**

5632/udp pcANYWHEREstat probes were reported to CERT/CC during reporting period: 01/03/2000 04:00 -0500 GMT through 01/03/2000 08:00 -0500 GMT

This detect was also listed in the SANS GIAC 2.5 Intrusion Detection Workshop book pg. 260.

### 7. Evidence of active targeting:

Intruder was targeting remote logon program PCAnywhare.

### 8. Severity:

Severity = (criticality + lethality) – (system + net) = (2+3) – (5+2) = -2

Criticality = 2: Targets were workstations.

Lethality = 3: PCAnywhare was not installed. However, the intruder did gain some information about the subnet.

System = 5 Both systems had all current Hot fixes applied and tightened security.

Net = 2 Router was wide open with limited logging capabilities. Router was secured with a strong password consisting of upper and lower case characters, numbers, and special characters.

### 9. Defensive recommendation:

Both machines had all of the latest security patches applied. The NT machine had security tightened using the “Microsoft Internet Information Server 4.0 Security Checklist” as a guide. The Windows 98 machine had file and print sharing disabled. Both machines contained nothing of value and the subnet was isolated.

The OCLM router had NAT disabled, allowed NetBIOS traffic to be passed, and was in an “always on” state. This is not a desired configuration. After the testing of snort was completed, the detects were collected and the router settings were securely set. If this were a production environment TCP/UDP traffic to ports 22 and 5632 and the IP address of the intruder would have been blocked.

### Multiple choice test question:

What can be said about the following trace?

May 24 16:41:29 208.58.x.x:1646 -> x.x.9.7:5632 UDP

May 24 16:42:20 208.58.x.x:1648 -> x.x.9.10:22 UDP

May 26 15:48:02 208.58.x.x:1277 -> x.x.9.7:5632 UDP

- a) UDP port probe
- b) Probe for remote logon
- c) Intruder has a pattern
- d) All of the above

Answer is ‘d’. In the above trace, the intruder sends UDP packets to ports 22 and 5632 (PCAnywhare), probing for a remote login program. The time stamp of the detects indicates activity at a particular time of day. This pattern could be used to build a profile of the intruder.

### Detect 5

*Time Stamp in all traces is GMT*

### Snort Alert:

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:26:51.027846 10.65.x.x -> x.x.9.7  
ICMP TTL:63 TOS:0x0 ID:55479  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:26:51.076705 10.65.x.x -> x.x.9.7  
ICMP TTL:63 TOS:0x0 ID:55735  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:26:51.125829 10.65.x.x -> x.x.9.7  
ICMP TTL:63 TOS:0x0 ID:55991
```

TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:26:52.237265 209.122.x.x -> x.x.9.7  
ICMP TTL:253 TOS:0xC0 ID:50552  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:26:52.329117 209.122.x.x -> x.x.9.7  
ICMP TTL:253 TOS:0xC0 ID:50554  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:26:52.377244 209.122.x.x -> x.x.9.7  
ICMP TTL:253 TOS:0xC0 ID:50555  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:26:53.498176 207.172.x.x -> x.x.9.7  
ICMP TTL:252 TOS:0xC0 ID:65161  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:26:53.553608 207.172.x.x -> x.x.9.7  
ICMP TTL:252 TOS:0xC0 ID:65162  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:26:53.608536 207.172.x.x -> x.x.9.7  
ICMP TTL:252 TOS:0xC0 ID:65163  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:26:54.729965 207.172.x.x -> x.x.9.7  
ICMP TTL:251 TOS:0xC0 ID:64837  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:26:54.784991 207.172.x.x -> x.x.9.7  
ICMP TTL:251 TOS:0xC0 ID:64838  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:26:54.841451 207.172.x.x -> x.x.9.7  
ICMP TTL:251 TOS:0xC0 ID:64839  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:26:55.942655 166.48.x.x -> x.x.9.7  
ICMP TTL:250 TOS:0xC0 ID:53923  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:26:56.041491 166.48.x.x -> x.x.9.7  
ICMP TTL:250 TOS:0xC0 ID:53926  
TTL EXCEEDED

## SANS GIAC – GIAC Practical

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:26:56.095617 166.48.x.x -> x.x.9.7  
ICMP TTL:250 TOS:0xC0 ID:53928  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:26:57.202797 204.70.x.x -> x.x.9.7  
ICMP TTL:249 TOS:0xC0 ID:11127  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:26:57.257976 204.70.x.x -> x.x.9.7  
ICMP TTL:249 TOS:0xC0 ID:11128  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:26:57.315406 204.70.x.x -> x.x.9.7  
ICMP TTL:249 TOS:0xC0 ID:11129  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:26:58.502712 204.70.x.x -> x.x.9.7  
ICMP TTL:243 TOS:0xC0 ID:6837  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:26:58.635821 204.70.x.x -> x.x.9.7  
ICMP TTL:243 TOS:0xC0 ID:6838  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:26:58.770576 204.70.x.x -> x.x.9.7  
ICMP TTL:243 TOS:0xC0 ID:6839  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:27:01.259837 144.232.x.x -> x.x.9.7  
ICMP TTL:244 TOS:0x0 ID:0  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:27:02.745960 144.232.x.x -> x.x.9.7  
ICMP TTL:244 TOS:0x0 ID:0  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:27:04.211003 144.232.x.x -> x.x.9.7  
ICMP TTL:244 TOS:0x0 ID:0  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:27:04.380526 144.232.x.x -> x.x.9.7  
ICMP TTL:243 TOS:0x0 ID:0  
TTL EXCEEDED
```

```
[**] PING-ICMP Time Exceeded [**]  
05/24-17:27:04.443084 144.232.x.x -> x.x.9.7
```

## SANS GIAC – GIAC Practical

ICMP TTL:243 TOS:0x0 ID:0  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:27:04.506885 144.232.x.x -> x.x.9.7  
ICMP TTL:243 TOS:0x0 ID:0  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:27:05.656453 144.232.x.x -> x.x.9.7  
ICMP TTL:242 TOS:0x0 ID:0  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:27:05.720927 144.232.x.x -> x.x.9.7  
ICMP TTL:242 TOS:0x0 ID:0  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:27:05.782679 144.232.x.x -> x.x.9.7  
ICMP TTL:242 TOS:0x0 ID:0  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:27:06.898487 144.232.x.x -> x.x.9.7  
ICMP TTL:241 TOS:0xC0 ID:25462  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:27:07.001843 144.232.x.x -> x.x.9.7  
ICMP TTL:241 TOS:0xC0 ID:25463  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:27:07.064720 144.232.x.x -> x.x.9.7  
ICMP TTL:241 TOS:0xC0 ID:25464  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:27:08.185631 144.232.x.x -> x.x.9.7  
ICMP TTL:240 TOS:0xC0 ID:41198  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:27:08.287086 144.232.x.x -> x.x.9.7  
ICMP TTL:240 TOS:0xC0 ID:41199  
TTL EXCEEDED

[\*\*] PING-ICMP Time Exceeded [\*\*]  
05/24-17:27:08.361142 144.232.x.x -> x.x.9.7  
ICMP TTL:240 TOS:0xC0 ID:41200  
TTL EXCEEDED

### Supporting Data from Total log:

**\*\*\*Initial traceroute data is not shown in order to reduce the amount of displayed data.\*\*\***



SANS GIAC – GIAC Practical

05/24-17:26:50.974125 x.x.9.7 -> x.x.2.2  
ICMP **TTL:2** TOS:0x0 ID:46114  
ID:768 Seq:3328 ECHO  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

05/24-17:26:51.027846 10.65.x.x -> x.x.9.7  
ICMP TTL:63 TOS:0x0 ID:55479  
TTL EXCEEDED  
00 00 00 00 45 00 00 5C B4 22 00 00 00 01 5B 1F ....E..\.".....[.  
D0 3A 09 07 D0 1C 02 02 08 00 E7 FF 03 00 0D 00 .....

05/24-17:26:51.028520 x.x.9.7 -> x.x.2.2  
ICMP **TTL:2** TOS:0x0 ID:46370  
ID:768 Seq:3584 ECHO  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

05/24-17:26:51.076705 10.65.x.x -> x.x.9.7  
ICMP TTL:63 TOS:0x0 ID:55735  
TTL EXCEEDED  
00 00 00 00 45 00 00 5C B5 22 00 00 00 01 5A 1F ....E..\."....Z.  
D0 3A 09 07 D0 1C 02 02 08 00 E6 FF 03 00 0E 00 .....

05/24-17:26:51.077353 x.x.9.7 -> x.x.2.2  
ICMP **TTL:2** TOS:0x0 ID:46626  
ID:768 Seq:3840 ECHO  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

05/24-17:26:51.125829 10.65.x.x -> x.x.9.7  
ICMP TTL:63 TOS:0x0 ID:55991  
TTL EXCEEDED  
00 00 00 00 45 00 00 5C B6 22 00 00 00 01 59 1F ....E..\."....Y.  
D0 3A 09 07 D0 1C 02 02 08 00 E5 FF 03 00 0F 00 .....

05/24-17:26:52.184568 x.x.9.7 -> x.x.2.2  
ICMP **TTL:3** TOS:0x0 ID:47138  
ID:768 Seq:4096 ECHO  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

05/24-17:26:52.237265 209.122.x.x -> x.x.9.7  
ICMP TTL:253 TOS:0xC0 ID:50552  
TTL EXCEEDED  
00 00 00 00 45 00 00 5C B8 22 00 00 01 01 56 1F ....E..\."....V.  
D0 3A 09 07 D0 1C 02 02 08 00 E4 FF 03 00 10 00 .....

05/24-17:26:52.238011 x.x.9.7 -> x.x.2.2

```

ICMP TTL:3 TOS:0x0 ID:47394
ID:768 Seq:4352 ECHO
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

```

05/24-17:26:52.329117 209.122.x.x -> x.x.9.7
ICMP TTL:253 TOS:0xC0 ID:50554
TTL EXCEEDED
00 00 00 00 45 00 00 5C B9 22 00 00 01 01 55 1F ....E..\."....U.
D0 3A 09 07 D0 1C 02 02 08 00 E3 FF 03 00 11 00 .....

```

```

05/24-17:26:52.329839 x.x.9.7 -> x.x.2.2
ICMP TTL:3 TOS:0x0 ID:47650
ID:768 Seq:4608 ECHO
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

```

05/24-17:26:52.377244 209.122.x.x -> x.x.9.7
ICMP TTL:253 TOS:0xC0 ID:50555
TTL EXCEEDED
00 00 00 00 45 00 00 5C BA 22 00 00 01 01 54 1F ....E..\."....T.
D0 3A 09 07 D0 1C 02 02 08 00 E2 FF 03 00 12 00 .....

```

```

05/24-17:26:53.439015 x.x.9.7 -> x.x.2.2
ICMP TTL:4 TOS:0x0 ID:48162
ID:768 Seq:4864 ECHO
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

```

05/24-17:26:53.498176 207.172.x.x -> x.x.9.7
ICMP TTL:252 TOS:0xC0 ID:65161
TTL EXCEEDED
00 00 00 00 45 00 00 5C BC 22 00 00 01 01 52 1F ....E..\."....R.
D0 3A 09 07 D0 1C 02 02 08 00 E1 FF 03 00 13 00 .....

```

\*\*\*Final traceroute data is not shown in order to reduce the amount of displayed data.\*\*\*

**1. Source of trace**

Small subnet using a 3com ISDN Office Connect LAN Modem (OCLM) Router as a gateway to the Internet.

**2. Detect was generated by:**

SNORT version 1.6 using 05172kany.rules

**3. Probability the source address was spoofed**

IP addresses were not spoofed. These alerts were generated in response to a traceroute.

**4. Description of attack:**

PING-ICMP Time Exceeded – Triggers when a IP datagram is received with the “protocol” field of the IP header set to 1 (ICMP) and the “type” field in the ICMP header set to 11 (Time Exceeded for a Datagram).

**5. Attack mechanism:**

Traceroute! I included this detect because when I first saw it in the snort alert log I got very excited. However, after reviewing the Total logs my excitement changed to... oops! This is a classic example of traceroute. During the intrusion detection study I ran traceroute against another one of our machines on the Internet not knowing that snort would react by generating an alert. Host x.x.9.7 ran traceroute against x.x.2.2. The result of running the traceroute was “[\*\*] PING-ICMP Time Exceeded [\*\*]” alerts being generated by snort.

**6. Correlation’s:**

This was classified as a false positive due to the fact that the monitored subnet was the cause of the alert. No correlation was needed.

**7. Evidence of active targeting:**

No evidence of targeting. The monitored subnet was the cause of the alert.

**8. Severity:**

Severity = (criticality + lethality) – (system + net) = (2+2) – (5+2) = -3

Criticality = 2: Target was a workstation.

Lethality = 2: Traceroute

System = 5 WinNT with all current Hot fixes applied and tightened security.

Net = 2 Router was wide open with limited logging capabilities. Router was secured with a strong password consisting of upper and lower case characters, numbers, and special characters.

**9. Defensive recommendation:**

No defensive actions required.

**Multiple choice test question:**

The following trace is classic example of?

```
05/24-17:26:51.077353 x.x.9.7 -> x.x.2.2
ICMP TTL:2 TOS:0x0 ID:46626
ID:768 Seq:3840 ECHO
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

```
05/24-17:26:51.125829 10.65.x.x -> x.x.9.7
ICMP TTL:63 TOS:0x0 ID:55991
TTL EXCEEDED
00 00 00 00 45 00 00 5C B6 22 00 00 00 01 59 1F ....E..\."....Y.
D0 3A 09 07 D0 1C 02 02 08 00 E5 FF 03 00 0F 00 .:.....
```

```
05/24-17:26:52.184568 x.x.9.7 -> x.x.2.2
ICMP TTL:3 TOS:0x0 ID:47138
ID:768 Seq:4096 ECHO
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
```

- a) ICMP DoS
- b) Traceroute
- c) Spoofed IP
- d) Fragmented ICMP

Answer is ‘b’. The incrementing TTL value is always a dead giveaway for traceroute.

## Detect 6

*Time Stamp in all traces is GMT*

### Snort Alert:

```
[**] spp_portscan: PORTSCAN DETECTED from 208.147.x.x [**]
05/25-21:05:21.209165
[**] spp_portscan: portscan status from 208.147.x.x: 4 connections across 1
hosts: TCP(0), UDP(4) [**]
05/25-21:05:27.036327
[**] spp_portscan: portscan status from 208.147.x.x: 3 connections across 1
hosts: TCP(0), UDP(3) [**]
05/25-21:05:33.080405
[**] spp_portscan: portscan status from 208.147.x.x: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/25-21:05:39.118070
[**] spp_portscan: portscan status from 208.147.x.x: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/25-21:05:45.104116
[**] spp_portscan: portscan status from 208.147.x.x: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/25-21:05:51.100181
[**] spp_portscan: portscan status from 208.147.x.x: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/25-21:05:57.093133
[**] spp_portscan: portscan status from 208.147.x.x: 1 connections across 1
hosts: TCP(0), UDP(1) [**]
05/25-21:06:06.451939
[**] spp_portscan: End of portscan from 208.147.x.x [**]
05/25-21:06:15.091873
```

### Supporting Data from Snort PortScan Log:

```
May 25 21:05:16 208.147.x.x:17648 -> x.x.9.10:6970 UDP
May 25 21:05:27 208.147.x.x:28449 -> x.x.9.10:6976 UDP
May 25 21:05:22 208.147.x.x:9402 -> x.x.9.10:6974 UDP
May 25 21:05:24 208.147.x.x:8117 -> x.x.9.10:6972 UDP
May 25 21:05:29 208.147.x.x:28449 -> x.x.9.10:6976 UDP
May 25 21:05:33 208.147.x.x:28786 -> x.x.9.10:6980 UDP
May 25 21:05:30 208.147.x.x:11973 -> x.x.9.10:6978 UDP
May 25 21:05:39 208.147.x.x:28786 -> x.x.9.10:6980 UDP
May 25 21:05:45 208.147.x.x:28786 -> x.x.9.10:6980 UDP
May 25 21:05:51 208.147.x.x:28786 -> x.x.9.10:6980 UDP
May 25 21:05:57 208.147.x.x:28786 -> x.x.9.10:6980 UDP
May 25 21:06:02 208.147.x.x:28786 -> x.x.9.10:6980 UDP
```

### Supporting Data from Total log:

```
05/25-21:05:16.212112 208.147.89.119:17648 -> 208.58.9.10:6970
UDP TTL:49 TOS:0x0 ID:36120
Len: 19
82 FF 06 00 01 00 00 00 00 00 02 00 00 00 00 00 .....
00 00 ..
```

```
05/25-21:05:16.234409 208.147.x.x:554 -> x.x.9.10:1284
TCP TTL:49 TOS:0x0 ID:36215 DF
*****A* Seq: 0xAE1126D3 Ack: 0x27A19 Win: 0x7D78
```

```

00 00 00 00 00 00 .....
05/25-21:05:16.257680 208.147.x.x:554 -> x.x.9.10:1284
TCP TTL:49 TOS:0x0 ID:36292 DF
*****PA* Seq: 0xAE1126D3 Ack: 0x27A19 Win: 0x7D78
52 54 53 50 2F 31 2E 30 20 32 30 30 20 4F 4B 0D RTSP/1.0 200 OK.
0A 43 53 65 71 3A 20 35 0D 0A 44 61 74 65 3A 20 .CSeq: 5..Date:
54 68 75 2C 20 32 35 20 4D 61 79 20 32 30 30 30 Thu, 25 May 2000
20 32 31 3A 30 36 3A 34 32 20 47 4D 54 0D 0A 52 21:06:42 GMT..R
54 50 2D 49 6E 66 6F 3A 20 75 72 6C 3D 72 74 73 TP-Info: url=rts
70 3A 2F 2F 6E 65 77 61 72 6B 2E 72 65 61 6C 2E p://newark.real.
63 6F 6D 3A 35 35 34 2F 73 68 6F 77 63 61 73 65 com:554/showcase
2F 63 68 61 6E 6E 65 6C 73 2F 61 62 63 6E 65 77 /channels/abcnew
73 2F 73 74 61 72 74 2E 73 6D 69 2F 73 74 72 65 s/start.smi/stre
61 6D 69 64 3D 30 3B 73 65 71 3D 30 3B 72 74 70 amid=0;seq=0;rtp
74 69 6D 65 3D 30 0D 0A 0D 0A 52 54 53 50 2F 31 time=0...RTSP/1
2E 30 20 34 35 31 20 50 61 72 61 6D 65 74 65 72 .0 451 Parameter
20 4E 6F 74 20 55 6E 64 65 72 73 74 6F 6F 64 0D Not Understood.
0A 43 53 65 71 3A 20 36 0D 0A 44 61 74 65 3A 20 .CSeq: 6..Date:
54 68 75 2C 20 32 35 20 4D 61 79 20 32 30 30 30 Thu, 25 May 2000
20 32 31 3A 30 36 3A 34 32 20 47 4D 54 0D 0A 0D 21:06:42 GMT...
0A .

```

**1. Source of trace**

Small subnet using a 3com ISDN Office Connect LAN Modem (OCLM) Router as a gateway to the Internet.

**2. Detect was generated by:**

SNORT version 1.6 using 05172kany.rules

**3. Probability the source address was spoofed**

IP address was not spoofed.

**4. Description of attack:**

spp\_portscan - Snort alerted port scan activity from IP address 208.147.x.x after an Internet music tuner was enabled.

**5. Attack mechanism:**

When I first saw this alert I thought I had a real live port scan, possibly trolling for Trojans (port 6970 DeepThroat/GateCrasher). I saw a reference to RTSP in the data portion of a TCP packet sent by 208.147.x.x., while reviewing the Total log. The Real Time Streaming (media control) Protocol (RTSP) is a remote control for multimedia on the Internet.

What happened was this, while reviewing logs I decided to listen to some music via an Internet music tuner. The RSTP protocol is intended to control multiple data delivery sessions, provide a means for choosing delivery channels such as UDP, multicast UDP and TCP, and delivery mechanisms based upon RTP (RFC 1889).

The Real Time Streaming Protocol, or RTSP, is an application-level protocol for control over the delivery of data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both live data feeds and stored clips.

Further verifying this is the use of port 554 by 208.147.x.x. Port 554 is the IANA assigned Real Time Stream Control Protocol port.

**6. Correlations:**

From the WELL KNOWN PORT NUMBERS document located on the IANA web site:

```

rtsp      554/tcp  Real Time Stream Control Protocol
rtsp      554/udp  Real Time Stream Control Protocol

```

### 9. Evidence of active targeting:

No evidence of targeting in this detect. The client initiated the session.

### 10. Severity:

Severity = (criticality + lethality) – (system + net) = (2+2) – (5+2) = -3

Criticality = 2: Target was a workstation.

Lethality = 2: Legitimate session.

System = 5 WinNT with all current Hot fixes applied and tightened security.

Net = 2 Router was wide open with limited logging capabilities. Router was secured with a strong password consisting of upper and lower case characters, numbers, and special characters.

### 12. Defensive recommendation:

The attack is a false positive. No defensive measures required.

### Multiple choice test question:

Which statement about the following detect is true?

```
05/25-21:05:16.234409 208.147.x.x:554 -> x.x.9.10:1284
TCP TTL:49 TOS:0x0 ID:36215 DF
*****A* Seq: 0xAE1126D3 Ack: 0x27A19 Win: 0x7D78
00 00 00 00 00 00 .....
```

- a) Packet has an illegal window size
- b) It's a crafted packet
- c) Sender is running with superuser privileges
- d) 1284 is a known Trojan port

Answer is 'c'. Port 554 is a reserved port assigned by IANA to the Real Time Stream Control Protocol. Reserved ports range from 1 to 1023. Only a process with superuser privileges can assign itself a reserved port. (TCP/IP Illustrated Volume 1, pg. 13)

### Detect 7

*Time Stamp in all traces is GMT*

### Snort Alert:

```
[**] OVERFLOW-NOOP-X86 [**]
05/26-15:45:53.496136 206.132.x.x:80 -> x.x.9.10:2591
TCP TTL:50 TOS:0x0 ID:12387 DF
*****PA* Seq: 0xEFFCCE4B Ack: 0x297A2 Win: 0x4470
```

```
[**] OVERFLOW-NOOP-X86 [**]
05/26-15:46:38.288032 206.132.x.x:80 -> x.x.9.10:2592
TCP TTL:50 TOS:0x0 ID:32446 DF
*****PA* Seq: 0x5DAADA8B Ack: 0x2986E Win: 0x4470
```

```
[**] OVERFLOW-NOOP-X86 [**]
05/26-15:46:38.367921 206.132.x.x:80 -> x.x.9.10:2592
TCP TTL:50 TOS:0x0 ID:32448 DF
*****PA* Seq: 0x5DAAE28B Ack: 0x2986E Win: 0x4470
```

```
[**] OVERFLOW-NOOP-X86 [**]
05/26-15:46:38.825934 206.132.x.x:80 -> x.x.9.10:2592
TCP TTL:50 TOS:0x0 ID:32453 DF
*****PA* Seq: 0x5DAAEA8B Ack: 0x2986E Win: 0x4470
```

```
[**] OVERFLOW-NOOP-X86 [**]
05/26-15:46:39.282247 206.132.x.x:80 -> x.x.9.10:2592
```



SANS GIAC – GIAC Practical

```
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
1D 5E 5F 89 4D EB 14 ED 60 17 72 5E 51 02 12 78 .^_.M...`r^Q..x
EB F4 01 60 2B 7F 46 1F 5E 12 78 18 DE 5E 2B 7F ...`+.F.^x...^+.
91 1E E9 36 E9 26 E9 2E AE B3 E9 4A 25 A4 60 5E ...6.&.....J%..`^
5F BB 21 88 8C A1 AD AB 5E B3 CA D2 C7 CB BF D2 _.!.....^.....
C3 7E A3 D4 C7 CA 7E C0 D7 7E A3 D4 C7 CA 7E AD ..~.....~.....~..
CC C3 5E 42 9E E4 22 42 9E 83 65 65 9A 62 D2 51 ..^B.."B...ee.b.Q
DE 5A 61 D2 6A DE 5A 64 D2 65 DE 5A 65 D2 60 49 .Za.j.Zd.e.Ze.`I
40 98 3E D2 3C AE DE 5A 61 D3 63 DE 52 62 49 61 @.>.<..Za.c.RbIa
DE 52 60 E8 52 B6 DE 04 25 5F 56 DE 04 28 5F 56 .R`.R...%_V..(_V
DE 04 2A 5F 56 DE 04 2E 5F 56 DE 04 30 5F 56 DE ..*_V..._V..0_V.
04 31 5F 56 DE 04 29 5F 56 66 04 25 5F 66 04 28 .1_V..) _Vf.%_f.(
5F 66 E4 2A 5F 66 14 2E 5F 66 04 30 5F 66 E4 31 _f.*_f..f.0_f.1
5F 66 04 29 5F 21 19 5E 8F B1 B1 1C 01 5F DE 65 _f.)_!..^....._e
00 A1 AC D3 57 21 A3 D4 C7 CA 12 AC 91 27 EB F4 ....W!.....!..
92 5F 2B 7F D0 65 46 63 5E 12 AD 49 53 21 16 5F ..+...eFc^..IS!.._
A1 EB F4 1F 60 91 27 2B 7F 16 60 9B 2B 7F D0 4B .....`'+...`+..K
F1 12 9D 17 72 5E EB F4 ED 60 2B 7F E9 50 E9 62 ....r^...`+..P.b
60 22 9A 05 D2 C9 DF DA 6E A3 D4 D2 C2 E9 E4 1B `"...n.....
60 63 5E 5F 9B 5E 4E D5 B6 9B 5E 7E D0 B1 E7 E4 `c^..^N...^~....
26 5F 0E 5E 46 AD 5E 42 9E 68 1E D2 58 E6 E4 2F &..^F.^B.h..X../
5F E6 E4 BF 60 46 69 5D 12 9E EB F4 25 5F 17 72 _...`Fi]....%_r
5E 2B 7F EB 1C 2B 60 EB 14 5E 5F 17 01 5F 0A 8A ^+...+`...^..._..
00 08 40 58 0E 60 46 7B 5E EB F4 2B 60 17 01 5F ..@X.`F{^...+`..._
12 9E 2B 7F 16 5F B5 E9 EC 17 60 E9 F4 19 60 2B ..+..._...`...+`
7F 12 9C 2B 7F 21 12 A0 91 27 F7 2B 7F 21 EE EE ...+!...!..+!...
EE EE EE EE EE EE EE EE EE EE EE EE EE EE EE EE .....
EE EE ..
```

05/26-15:46:38.288032 206.132.x.x:80 -> x.x.9.10:2592

TCP TTL:50 TOS:0x0 ID:32446 DF

\*\*\*\*\*PA\* Seq: 0x5DAADA8B Ack: 0x2986E Win: 0x4470

```
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
```



SANS GIAC – GIAC Practical

```
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
```

05/26-15:46:38.367921 206.132.x.x.80 -> x.x.9.10:2592

TCP TTL:50 TOS:0x0 ID:32448 DF

\*\*\*\*\*PA\* Seq: 0x5DAAE28B Ack: 0x2986E Win: 0x4470

```
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....  
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 .....
```





No information about NOOP attack found at <http://www.securityfocus.com/>  
No information about NOOP attack found at <http://www.cert.org/>

### 7. Evidence of active targeting:

The host that that requested data from 206.132.x.x:80 was targeted for this attack.

### 8. Severity:

Severity = (criticality + lethality) – (system + net) = (2+4) – (5+2) = -1

Criticality = 2: Target was a workstation.

Lethality = 4: Buffer Overflow Attempt.

System = 5 WinNT with all current Hot fixes applied and tightened security.

Net = 2 Router was wide open with limited logging capabilities. Router was secured with a strong password consisting of upper and lower case characters, numbers, and special characters.

### 9. Defensive recommendation:

Defenses are fine. All systems had updated patches. The attack did not appear to work.

### Multiple choice test question:

In a TCP Session, which is true about the PUSH flag?

- a) must be set for a packet to contain data.
- b) Is used only in FTP sessions
- c) Tells the receiver to pass all data to the receiving process.
- d) All of the above.

Answer is 'c'. It's a notification from the sender to the receiver for the receiver to pass all the data that it has to the receiving process. (TCP/IP Illustrated, Volume 1, pg. 284)

## Detect 8

*Time Stamp in all traces is GMT*

### Snort Alert:

```
[**] MISC-WinGate-1080-Attempt [**]  
05/26-20:51:38.882947 216.234.161.197:1502 -> 208.58.9.10:1080  
TCP TTL:50 TOS:0x0 ID:45706 DF  
**S***** Seq: 0x6692F9C7 Ack: 0x0 Win: 0x4000  
TCP Options => MSS: 1460
```

### Supporting Data from Total log:

```
05/26-20:51:10.644817 x.x.9.10:3242 -> 216.234.x.x:6667  
TCP TTL:128 TOS:0x10 ID:13024 DF  
**S***** Seq: 0x2A82F Ack: 0x0 Win: 0x2000  
TCP Options => MSS: 1460  
74 03 t.
```

```
05/26-20:51:10.757642 216.234.x.x:6667 -> x.x.9.10:3242  
TCP TTL:50 TOS:0x0 ID:42127 DF  
**S***A* Seq: 0x65CA3DF4 Ack: 0x2A830 Win: 0x2238  
TCP Options => MSS: 1460  
00 00 ..
```

```
05/26-20:51:10.759818 x.x.9.10:3242 -> 216.234.x.x:6667
```

SANS GIAC – GIAC Practical

TCP TTL:128 TOS:0x10 ID:13792 DF  
\*\*\*\*\*A\* Seq: 0x2A830 Ack: 0x65CA3DF5 Win: 0x2238  
02 04 05 B4 74 03 . . . . t.

\*\*\*\*\*Data after 3 way hand shake not shown in order to save space\*\*\*\*\*

05/26-20:51:38.877035 216.234.x.x:6667 -> x.x.9.10:3242  
TCP TTL:50 TOS:0x0 ID:45672 DF  
\*\*\*\*\*PA\* Seq: 0x65CA3DF5 Ack: 0x2A880 Win: 0x21E8  
3A 69 72 63 2E 6D 69 72 63 78 2E 63 6F 6D 20 30 :irc.mircx.com 0  
30 31 20 47 75 65 73 74 37 35 31 33 33 20 3A 57 01 Guest75133 :W  
65 6C 63 6F 6D 65 20 74 6F 20 74 68 65 20 6D 49 elcome to the mI  
52 43 2D 58 20 49 52 43 20 4E 65 74 77 6F 72 6B RC-X IRC Network  
20 47 75 65 73 74 37 35 31 33 33 21 7E 6A 61 76 Guest75133!~jav  
61 40 32 30 38 2E 35 38 2E 39 2E 31 30 0D 0A 3A a@208.58.9.10.:  
69 72 63 2E 6D 69 72 63 78 2E 63 6F 6D 20 30 30 irc.mircx.com 00  
32 20 47 75 65 73 74 37 35 31 33 33 20 3A 59 6F 2 Guest75133 :Yo  
75 72 20 68 6F 73 74 20 69 73 20 69 72 63 2E 6D ur host is irc.m  
69 72 63 78 2E 63 6F 6D 2C 20 72 75 6E 6E 69 6E ircx.com, runnin  
67 20 76 65 72 73 69 6F 6E 20 64 61 6C 34 2E 36 g version dal4.6  
2E 37 62 2E 44 72 65 61 6D 46 6F 72 67 65 0D 0A .7b.DreamForge..  
3A 69 72 63 2E 6D 69 72 63 78 2E 63 6F 6D 20 30 :irc.mircx.com 0  
30 33 20 47 75 65 73 74 37 35 31 33 33 20 3A 54 03 Guest75133 :T  
68 69 73 20 73 65 72 76 65 72 20 77 61 73 20 63 his server was c  
72 65 61 74 65 64 20 54 75 65 20 41 70 72 20 31 reated Tue Apr 1  
38 20 32 30 30 30 20 61 74 20 31 33 3A 33 35 3A 8 2000 at 13:35:  
33 36 20 4D 44 54 0D 0A 3A 69 72 63 2E 6D 69 72 36 MDT.:irc.mir  
63 78 2E 63 6F 6D 20 30 34 20 47 75 65 73 74 cx.com 004 Guest  
37 35 31 33 33 20 69 72 63 2E 6D 69 72 63 78 2E 75133 irc.mircx.  
63 6F 6D 20 64 61 6C 34 2E 36 2E 37 62 2E 44 72 com dal4.6.7b.Dr  
65 61 6D 46 6F 72 67 65 20 6F 69 77 73 67 68 4F eamForge oiwsgH0  
6B 63 66 72 52 61 41 62 20 62 69 6B 6C 6D 6E 6F kcfrRaAb biklmno  
70 73 74 76 52 0D 0A 3A 69 72 63 2E 6D 69 72 63 pstvR.:irc.mirc  
78 2E 63 6F 6D 20 30 30 35 20 47 75 65 73 74 37 x.com 005 Guest7  
35 31 33 33 20 4E 4F 51 55 49 54 20 54 4F 4B 45 5133 NOQUIT TOKE  
4E 20 57 41 54 43 48 3D 31 32 38 20 53 41 46 45 N WATCH=128 SAFE  
4C 49 53 54 20 3A 61 72 65 20 61 76 61 69 6C 61 LIST :are availa  
62 6C 65 20 6F 6E 20 74 68 69 73 20 73 65 72 76 ble on this serv  
65 72 0D 0A 3A 69 72 63 2E 6D 69 72 63 78 2E 63 er.:irc.mircx.c  
6F 6D 20 32 35 31 20 47 75 65 73 74 37 35 31 33 om 251 Guest7513  
33 20 3A 54 68 65 72 65 20 61 72 65 20 32 31 20 3 :There are 21  
75 73 65 72 73 20 61 6E 64 20 32 38 37 20 69 6E users and 287 in  
76 69 73 69 62 6C 65 20 6F 6E 20 38 20 73 65 72 visible on 8 ser  
76 65 72 73 0D 0A 3A 69 72 63 2E 6D 69 72 63 78 vers.:irc.mircx  
2E 63 6F 6D 20 32 35 32 20 47 75 65 73 74 37 35 .com 252 Guest75  
31 33 33 20 31 32 20 3A 6F 70 65 72 61 74 6F 72 133 12 :operator  
28 73 29 20 6F 6E 6C 69 6E 65 0D 0A 3A 69 72 63 (s) online.:irc  
2E 6D 69 72 63 78 2E 63 6F 6D 20 32 35 33 20 47 .mircx.com 253 G  
75 65 73 74 37 35 31 33 33 20 39 20 3A 75 6E 6B uest75133 9 :unk  
6E 6F 77 6E 20 63 6F 6E 6E 65 63 74 69 6F 6E 28 nown connection(  
73 29 0D 0A 3A 69 72 63 2E 6D 69 72 63 78 2E 63 s):irc.mircx.c  
6F 6D 20 32 35 34 20 47 75 65 73 74 37 35 31 33 om 254 Guest7513  
33 20 31 30 36 20 3A 63 68 61 6E 6E 65 6C 73 20 3 106 :channels  
66 6F 72 6D 65 64 0D 0A 3A 69 72 63 2E 6D 69 72 formed.:irc.mir  
63 78 2E 63 6F 6D 20 32 35 35 20 47 75 65 73 74 cx.com 255 Guest  
37 35 31 33 33 20 3A 49 20 68 61 76 65 20 32 32 75133 :I have 22  
37 20 63 6C 69 65 6E 74 73 20 61 6E 64 20 37 20 7 clients and 7

SANS GIAC – GIAC Practical

```
73 65 72 76 65 72 73 0D 0A 3A 69 72 63 2E 6D 69 servers...:irc.mi
72 63 78 2E 63 6F 6D 20 32 36 35 20 47 75 65 73 rcx.com 265 Gues
74 37 35 31 33 33 20 3A 43 75 72 72 65 6E 74 20 t75133 :Current
6C 6F 63 61 6C 20 75 73 65 72 73 3A 20 32 32 37 local users: 227
20 20 4D 61 78 3A 20 33 36 32 0D 0A 3A 69 72 63 Max: 362...:irc
2E 6D 69 72 63 78 2E 63 6F 6D 20 32 36 36 20 47 .mircx.com 266 G
75 65 73 74 37 35 31 33 33 20 3A 43 75 72 72 65 uest75133 :Curre
6E 74 20 67 6C 6F 62 61 6C 20 75 73 65 72 73 3A nt global users:
20 33 30 38 20 20 4D 61 78 3A 20 36 39 31 0D 0A 308 Max: 691..
3A 69 72 63 2E 6D 69 72 63 78 2E 63 6F 6D 20 33 :irc.mircx.com 3
37 35 20 47 75 65 73 74 37 35 31 33 33 20 3A 2D 75 Guest75133 :-
20 69 72 63 2E 6D 69 72 63 78 2E 63 6F 6D 20 4D irc.mircx.com M
65 73 73 61 67 65 20 6F 66 20 74 68 65 20 44 61 essage of the Da
79 20 2D 20 0D 0A 3A 69 72 63 2E 6D 69 72 63 78 y - ...:irc.mircx
2E 63 6F 6D 20 33 37 32 20 47 75 65 73 74 37 35 .com 372 Guest75
31 33 33 20 3A 2D 20 33 31 2F 31 32 2F 31 39 36 133 :- 31/12/196
39 20 31 37 3A 30 30 0D 0A 9 17:00..
```

```
05/26-20:51:38.878333 x.x.9.10:3242 -> 216.234.x.x:6667
TCP TTL:128 TOS:0x10 ID:48352 DF
****R*** Seq: 0x2A880 Ack: 0x101E51E7 Win: 0x0
02 04 05 B4 0C 6E .....
```

```
05/26-20:51:38.882947 216.234.x.x:1502 -> x.x.9.10:1080
TCP TTL:50 TOS:0x0 ID:45706 DF
**S***** Seq: 0x6692F9C7 Ack: 0x0 Win: 0x4000
TCP Options => MSS: 1460
00 00 ..
```

```
05/26-20:51:38.883195 x.x.9.10:1080 -> 216.234.x.x:1502
TCP TTL:128 TOS:0x0 ID:48608
****R*A* Seq: 0x0 Ack: 0x6692F9C8 Win: 0x0
02 04 05 B4 0C 6E .....
```

```
05/26-20:51:38.906874 216.234.x.x:6667 -> x.x.9.10:3242
TCP TTL:50 TOS:0x0 ID:45762 DF
***F*PA* Seq: 0x65CA41FE Ack: 0x2A880 Win: 0x21E8
3A 69 72 63 2E 6D 69 72 63 78 2E 63 6F 6D 20 33 :irc.mircx.com 3
37 36 20 47 75 65 73 74 37 35 31 33 33 20 3A 45 76 Guest75133 :E
6E 64 20 6F 66 20 2F 4D 4F 54 44 20 63 6F 6D 6D nd of /MOTD comm
61 6E 64 2E 0D 0A 3A 47 75 65 73 74 37 35 31 33 and...:Guest7513
33 20 4D 4F 44 45 20 47 75 65 73 74 37 35 31 33 3 MODE Guest7513
33 20 3A 2B 69 0D 0A 45 52 52 4F 52 20 3A 43 6C 3 :+i..ERROR :Cl
6F 73 69 6E 67 20 4C 69 6E 6B 3A 20 47 75 65 73 osing Link: Gues
74 37 35 31 33 33 5B 32 30 38 2E 35 38 2E 39 2E t75133[208.58.9.
31 30 5D 20 28 43 6C 69 65 6E 74 20 65 78 69 74 10] (Client exit
65 64 29 0D 0A ed) ..
```

```
05/26-20:51:38.907048 x.x.9.10:3242 -> 216.234.x.x:6667
TCP TTL:128 TOS:0x0 ID:48864
****R*** Seq: 0x2A880 Ack: 0x2A880 Win: 0x0
02 04 05 B4 0C 6E .....
```

```
05/26-20:51:41.844661 206.132.x.x:80 -> x.x.9.10:3259
TCP TTL:50 TOS:0x0 ID:47565 DF
*****A* Seq: 0x71434551 Ack: 0x2A9B3 Win: 0x4470
48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D HTTP/1.1 200 OK.
```

```
0A 44 61 74 65 3A 20 46 72 69 2C 20 32 36 20 4D .Date: Fri, 26 M
61 79 20 32 30 30 30 20 31 39 3A 34 36 3A 33 35 ay 2000 19:46:35
20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 41 70 GMT..Server: Ap
61 63 68 65 2F 31 2E 32 2E 35 0D 0A 4C 61 73 74 ache/1.2.5..Last
2D 4D 6F 64 69 66 69 65 64 3A 20 57 65 64 2C 20 -Modified: Wed,
31 30 20 4D 61 79 20 32 30 30 30 20 30 39 3A 32 10 May 2000 09:2
36 3A 30 34 20 47 4D 54 0D 0A 45 54 61 67 3A 20 6:04 GMT..ETag:
22 31 37 33 32 38 33 63 2D 39 30 65 2D 33 39 31 "173283c-90e-391
39 32 62 32 63 22 0D 0A 43 6F 6E 74 65 6E 74 2D 92b2c"..Content-
4C 65 6E 67 74 68 3A 20 32 33 31 38 0D 0A 41 63 Length: 2318..Ac
63 65 70 74 2D 52 61 6E 67 65 73 3A 20 62 79 74 cept-Ranges: byt
65 73 0D 0A 4B 65 65 70 2D 41 6C 69 76 65 3A 20 es..Keep-Alive:
74 69 6D 65 6F 75 74 3D 31 35 0D 0A 43 6F 6E 6E timeout=15..Conn
65 63 74 69 6F 6E 3A 20 4B 65 65 70 2D 41 6C 69 ection: Keep-Ali
76 65 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79 70 65 ve..Content-Type
3A 20 69 6D 61 67 65 2F 67 69 66 0D 0A 0D 0A 47 : image/gif....G
49 46 38 37 61 FA 00 32 00 D5 00 00 04 02 04 84 IF87a..2.....
82 84 44 42 44 C4 C2 C4 24 22 24 A4 A2 A4 64 62 ..DBD...$"$.db
64 E4 E2 E4 14 12 14 94 92 94 54 52 54 D4 D2 D4 d.....TRT...
34 32 34 B4 B2 B4 74 72 74 F4 F2 F4 0C 0A 0C 8C 424...trt.....
8A 8C 4C 4A 4C CC CA CC 2C 2A 2C AC AA AC 6C 6A ..LJL...,*,...lj
6C EC EA EC 1C 1A 1C 9C 9A 9C 5C 5A 5C DC DA DC l.....\Z\...
3C 3A 3C BC BA BC 7C 7A 7C FC FA FC 04 06 04 84 <:<...|z|.....
86 84 44 46 44 C4 C6 C4 24 26 24 A4 A6 A4 64 66 ..DFD...$$$.df
64 E4 E6 E4 14 16 14 94 96 94 54 56 54 D4 D6 D4 d.....TVT...
34 36 34 B4 B6 B4 74 76 74 F4 F6 F4 0C 0E 0C 8C 464...tvt.....
8E 8C 4C 4E 4C CC CE CC 2C 2E 2C AC AE AC 6C 6E ..LNL...,...ln
6C EC EE EC 1C 1E 1C 9C 9E 9C 5C 5E DC DE DC l.....\^\.
3C 3E 3C BC BC BC 7C 7E 7C FC FE FC 2C 00 00 00 <><...|~|...
00 FA 00 32 00 00 06 FF 40 80 70 48 2C 1A 8F C8 ...2....@.pH,...
A4 72 C9 6C 3A 9F D0 A8 74 4A AD 5A AF D8 AC 76 .r.l:...tJ.Z...v
CB ED 7A BF E0 B0 78 4C 2E 9B CF E8 B4 7A CD 6E ..z...xL.....zn
BB DF F0 B8 7C 4E AF DB B1 A0 BB 7E CF CF E6 F3 ....|N.....~....
7D 81 82 6E 80 00 85 83 88 89 77 7F 4B 30 28 08 }..n.....w.K0(.
8F 08 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F .....
A0 A1 A2 A3 A4 A2 30 08 10 4E 0E 3F AC AD AE AF .....0..N.?....
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF .....
AD 05 86 4C 3C 29 09 09 29 C6 CA C8 CC CB CE CD ...L<)..).....
D0 CF D2 D1 D4 D3 D6 D5 D8 D7 DA D9 DC DB DE DD .....
E0 DF E2 E1 D0 31 29 2A 00 A9 8A EB EC 72 20 87 .....1)*.....r .
48 EF F2 F3 F4 F5 F6 F7 F5 30 10 20 FA 20 FB F8 H.....0. . . .
FE F6 A5 4B 07 B0 A0 C1 83 08 13 16 DC 07 83 5F ...K.....-
C2 54 EF 1A FE B3 97 4E 9D 3F 23 F2 26 2A 2C D8 .T.....N.?#.&*,.
07 DE 12 10 28 50 C0 68 57 C5 23 13 0C 28 04 2A ....(P.hW.#..(*
C1 11 12 05 06 94 30 1F A1 60 44 92 0A 0E 0E 32 .....0..`D....2
FE 31 D9 F0 23 45 4D FF 29 79 48 70 60 E1 04 05 .1..#EM.)yHp`...
2B 13 C3 90 18 FD F0 80 E9 87 0F AC 9E 42 E5 91 +.....B..
F4 27 13 40 79 3C B0 C2 30 C4 E4 3B 00 50 85 99 .'.@y<..0...;P..
B4 8A 51 48 85 1F 27 8A B2 72 40 50 69 AD 17 1F ..QH...'..r@Pi...
38 54 25 7B 15 80 8E A7 08 9A 80 D0 61 82 2A 5D 8T%{.....a.*]
BD 00 62 7C D8 E1 04 86 09 13 34 96 40 38 6C C1 ..b|.....4.@8l.
44 63 0B 7C 2B 3C FD 41 F4 E2 DF 8F 00 0C B0 1A Dc.|+<.A.....
79 D9 4B 8A 1F 1B CE 8C F8 71 C3 06 CD 36 01 65 y.K.....q...6.e
72 7E 02 41 D2 23 75 49 40 20 70 F1 E3 03 09 97 r~.A.#uI@ p.....
B0 95 62 C8 EB A4 B5 24 7E 92 9A 38 1A E9 0F 12 ..b....$~..8....
6F 21 8E 10 AC BE 0A E1 94 A4 DC 49 4E 65 F8 71 o!.....INe.q
40 E6 D8 21 2F 97 C7 C3 2A 31 01 AB 05 04 9A 84 @...!/...*1.....
```

```

14 C2 6F FC 10 DF 30 B4 03 0E B0 60 41 ED 1F 0B ..o...0....`A...
7C 1C 5F A2 60 C0 03 56 17 66 68 50 62 F4 FD 87 |. _.`..V.fhPb...
17 3F 20 A5 C4 05 3D CD 45 1E 79 79 28 30 C2 7D .? ...=.E.yy(0.}
2F 54 C0 D3 05 4D 0C F0 C3 00 1C 34 30 00 5C 37 /T...M.....40.\7
68 80 02 05 15 9C 50 FF DB 06 19 CC 74 1D 00 30 h.....P.....t..0
F4 30 00 81 4C F5 80 CE 12 1C FC F0 02 5C 4C 51 .0..L.....\LQ
C6 DF 51 06 12 71 5A 43 0E 40 B5 42 62 58 21 21 ..Q..qZC.@.BbX!!
02 2B 0A 04 D0 C1 01 61 DD 66 C1 04 FF 5D 30 81 .+.....a.f...]0.
04 35 C6 06 C0 0C AF 40 55 02 0A 4C A8 00 20 2C .5.....@U..L...
54 22 81 43 54 50 AD B5 C4 0E 3F 24 A0 18 00 2A T".CTP....?$.**
78 08 4B 5A 4A E4 D1 03 75 66 72 39 00 4F AD 40 x.KZJ...ufr9.O.@
55 C1 7C 47 F4 17 A5 09 23 0A 50 DB 7F AD 08 C0 U.|G....#.P....
DF 53 02 3A 91 07 03 37 BC 70 82 5F 23 02 C0 C3 .S:....7.p._#...
0F 0F AC E0 0A 80 1B 74 D0 65 54 37 88 00 45 82 .....t.eT7..E.
87 99 60 40 01 27 DC F0 80 5F 65 91 48 E0 0E 3D ..`@.'..._e.H.=
D8 60 40 08 13 FC 90 25 46 28 98 50 C0 53 0E 1C .`@....%F(.P.S..
46 C2 12 07 84 D9 E4 10 08 10 78 C1 08 26 B8 70 F.....x...&p
22 5A 98 49 F8 DF 08 2E 18 E0 E8 7D 3F AC 90 82 "Z.I.....}?...
01 25 D4 76 03 55 D7 81 60 80 01 9A 9A 70 E2 07 .%.v.U..`.....p..
0B 8C 88 83 09 92 5E C0 18 95 63 21 00 55 A0 BD .....^....c!.U..
35 C0 0A 93 89 0A D1 FF E2 7D 3B 14 60 42 02 7C 5.....};.`B.|
BA 38 80 09 36 10 F9 43 0D B7 3A 91 00 5C 3A 2C .8..6..C...:;,
81 01 54 0E AC 86 03 03 5E 0D A4 42 6D 4F D4 2A ..T.....^...BmO.*
66 3C 42 D0 F6 41 00 59 A2 50 00 75 98 AD F9 80 f<B..A.Y.P.u....
03 59 92 30 03 53 35 84 67 48 0E 50 19 70 E0 11 .Y.0.S5.gH.P.p..
F0 60 B0 E6 05 AB 92 0C 80 77 07 D8 38 16 0A E3 .`.....w...8...
46 31 DD 0F 0D E4 5B 44 8B 3F 74 C0 02 20 3E 3C F1....[D.?t... ><
75 82 06 BC F1 B0 01 B6 51 00 E2 1A 02 34 C4 5C u.....Q.....\
F0 52 13 10 A0 12 66 AB A8 EA 4F A2 0A DF 9A C7 .R....f....O.....
00 1F 5C 20 57 73 2B FF 40 58 9A 00 48 D8 03 79 ..\ Ws+.@X..H..y
A9 98 50 1B 0E 07 92 70 C3 0F 22 47 6B C8 29 38 ..P....p.."Gk.)8
CC B6 D5 75 54 4E B7 83 3C F9 8A 1B 20 60 10 04 ...uTN..<.... `..
00 55 0F 29 3D B1 E8 0F 96 26 E5 61 06 42 28 07 .U.)=....&.a.B(.
C0 67 2D A7 .g-.
```

```

05/26-20:51:42.030326 x.x.9.10:3259 -> 206.132.x.x:80
TCP TTL:128 TOS:0x10 ID:49120 DF
*****A* Seq: 0x2A9B3 Ack: 0x71434B05 Win: 0x2238
02 04 05 B4 0C 6E .....n
```

```

05/26-20:51:42.305892 206.132.x.x:80 -> x.x.9.10:3259
TCP TTL:50 TOS:0x0 ID:49218 DF
****PA* Seq: 0x71434B05 Ack: 0x2A9B3 Win: 0x4470
8B 20 02 02 94 30 C0 82 93 89 CC 30 04 E6 26 5B . ...0.....0..&[
43 00 0C 24 16 5B 1E 66 BF C0 55 BA 55 3B E9 28 C..$. [.f..U.U;.(
E0 A9 8C C4 72 13 6B 76 30 0C E8 B5 F1 96 87 DE ....r.kv0.....
48 8D 88 FF 81 01 1D 74 30 F4 7F 1F 50 09 5D 57 H.....t0...P.]W
8B 7B 3D 72 12 30 EF ED 84 06 50 6D 40 81 CD 45 .{=r.0....Pm@...E
0C BE F5 10 60 FA C4 48 04 C0 42 01 43 B3 AF 00 ....`...H..B.C...
58 FB E5 15 5C C9 CA 01 3E 78 6E E3 81 9A F5 EE X...\...>xn.....
44 D5 1E 7D 05 80 87 31 9C 17 FC D7 4B 88 5D 95 D..}...1....K.].
05 B5 A5 5C 7C DB 49 70 00 A7 2B 50 A5 5C D6 67 ...\|.Ip..+P.\.g
F0 1B 5E 11 68 E7 04 93 41 C5 06 24 0A D5 11 70 ..^..h...A..$...p
26 17 22 D4 AA 7D 8C B0 5B 14 34 03 97 0E B4 C0 &."...}[.4.....
07 36 20 10 B9 C6 37 10 10 48 40 03 2D 98 00 B2 .6 ...7..H@.-...
2A E0 24 B3 7D 80 2B CC 2B 9D CA 40 C0 13 31 11 *.$..}..+..+..@..1.
A7 6B 2D 03 5B 07 26 54 95 F2 CD A7 78 16 C8 57 .k-.[.&T....x..W
```



SANS GIAC – GIAC Practical

```

0E 58 71 82 1E 34 C0 04 3B FC 01 9D 8A 90 8A D5 .Xq..4...;.....
09 D0 08 C5 DB 20 12 AE C7 8A 10 55 25 5D 2C 60 .....U%],`
45 03 87 80 BE 61 18 F1 09 21 60 8A 07 10 80 36 E....a...!`....6
B0 EC 2D 7D 08 1A 02 0A 70 20 82 0D BC 20 34 4A ..-}....p ... 4J
30 DB 0D 88 43 35 5B 81 51 08 33 EC 01 02 0A E1 0...C5[.Q.3.....
9D 00 AA FF 4C 7E 85 A0 9F 10 89 D0 9F ED 21 61 ....L~.....!a
05 1F 58 81 04 1E 01 00 05 D4 EF 56 AA A3 18 DE ..X.....V....
92 40 40 27 C1 60 66 03 00 D7 40 80 97 3F 56 10 .@@'.`f...@...?V.
C5 81 6E 1C 06 E3 9E 90 07 EF 9C A0 10 49 FB 41 ..n.....I.A
0E B7 03 80 DD 1C 62 3A 31 54 99 09 3D 20 84 A7 .....b:1T..=..
1D 41 85 A1 6A 81 8B 14 40 1E 1A 40 29 95 0C 5B .A..j...@...@)...[
D3 00 AA 62 C2 1B 2A 2D 09 3C C9 C1 10 60 E0 82 ...b...*-...<...`..
A7 F8 AF 08 79 F8 CC 03 98 24 90 97 C5 CC 49 84 ....y....$....I.
9A 90 DB 92 10 C5 1F 4C 51 08 55 4C 26 C5 04 15 .....LQ.UL&...
BC 1B 34 80 03 3C 98 CE 7D FC 78 04 0C DC 20 03 ..4...<...}.x...
25 50 00 0F 14 30 A4 7B E5 2B 74 0B 60 41 04 94 %P...0.{.+t.`A..
58 84 2A 26 41 07 70 D9 81 0C 04 50 00 F7 54 CF X.*&A.p....P..T.
49 6B 1A 5B 21 CC B6 47 31 FE 12 09 04 7A 80 07 Ik.[!..G1.....z..
38 20 80 54 3D E0 05 C7 1C 5F 0C 6A F3 CD 1C 34 8 .T=....._j...4
D0 24 8D 8C CE 0E E0 F2 4D 01 80 93 03 0C E5 01 .$.....M.....
0F 5E 88 04 9C 5D 92 8A 99 D4 A6 1D 99 C0 01 47 .^...]......G
4D EA 02 FF 27 F8 00 3D 0D 91 07 0C C8 A2 52 4B M...'....=.....RK
E0 C1 43 BB 94 C3 EB EC E0 03 ED FB 5C D8 DE 43 ..C.....\...C
A9 7F 32 4C 52 AE 1B E8 7F 6E 08 17 72 1A C1 04 ..2LR.....n..r...
70 71 45 4C 4F 08 36 0D 00 A8 4B 7E 7A 59 53 3F pqELO.6...K~zYS?
07 B3 07 20 4B 16 7E BA 15 0B 9E 72 52 21 FC 14 ... K.~.....rR!..
82 C3 10 CC 4A 99 40 01 33 BD 40 45 AC 18 E5 76 .....J.@.3.@E...v
3C B0 81 5A 7D 67 06 B4 1C CB 1F 3C B0 B6 0F DC <...Z}g.....<...
40 06 4B 00 90 30 BD 02 08 19 F8 13 2D 16 F8 0C @.K..0.....-<...
9A 9C 04 A5 19 54 25 6A 29 B3 E9 0F 10 38 22 82 .....T%j).....8".
A2 25 04 01 60 45 17 63 83 80 0C AC ED 05 3B 10 .%...`E.c.....;..
5F 9D 58 41 D9 24 48 76 16 53 C9 29 2B 40 25 04 _..XA.$Hv.S.)+@%.
C1 BE 0E 00 13 7B C0 14 84 C2 03 1A 50 09 9C 11 .....{.....P...
35 02 09 48 C0 02 1E B0 00 07 28 BC 0A 6F 39 40 5..H.....(..o9@
81 44 B1 80 03 1E 63 02 0E 8E CB 01 12 E0 00 A9 .D....c.....
CC A3 01 07 44 0B 00 14 F0 80 03 87 00 01 6E 9B .....D.....n.
40 03 91 16 17 01 43 71 1C 0C 28 C0 02 16 90 60 @.....Cq..(.....`
44 DA E5 FF 41 6E BB 02 52 DF 5E B7 BD BD 05 A7 D...An..R.^.....
7A 8A 00 83 EB 6A E7 B8 C9 15 02 01 7C CB BC CE z....j.....|...
E4 F4 70 FE 0D F0 79 58 C0 80 F2 0E 65 28 C7 3D ..p...yX....e.(=
6E 81 17 4C E0 F2 32 98 03 0C 38 70 79 1B 3C 61 n..L..2...8py.<a
07 57 B8 C0 12 6E 30 83 37 7C E0 0D 6B 78 C2 11 .W...n0.7|.kx..
06 27 0F 24 90 01 40 DE 20 01 0D F6 AD 87 09 DC .'.$.@. ....
61 10 B7 F8 C1 0A FE 30 83 1B 9C E1 04 CB F8 C6 a.....0.....
11 4E 30 84 29 BC E2 17 E3 58 C7 40 96 F0 8E 41 .N0.)....X.@...A
7C 63 04 73 38 C6 0B EE 30 0E 7A E4 23 60 38 F9 |c.s8...0.z.#`8.
C9 BA 98 14 94 A7 4C E5 2A 5B B9 16 F8 4A 14 0A .....L.*[...J..
74 A0 03 03 70 D9 CB 5D FE B2 98 C3 4C 66 30 9B t...p...]....Lf0.
79 CC 67 2E 33 9A D7 AC E6 36 A7 D9 CB 2E 70 54 y.g.3....6....pT
2B 2E 10 83 69 99 20 CC 6F 66 73 9E DD AC E7 3E +...i. .ofs....>
F3 F9 CF 7B 0E B4 9F 05 0D E8 41 EB B9 32 FD F5 ...{.....A..2..
EF C0 04 E0 DB F3 0A F8 D1 07 62 84 A4 5F 3B E9 .....b..._;.
4A 53 FA D2 96 CE 34 A6 37 7D 1A 8C D0 83 D3 A0 JS....4.7}.....
D6 B4 A8 33 43 4D EA 51 9B BA D4 A8 3E B5 A4 D5 ...3CM.Q....>...
07 E9 56 BB FA D5 B0 8E B5 AC 67 4D EB 5A DB FA ..V.....gM.Z..
D6 B8 CE B5 AE 77 CD EB 5E FB FA D7 C0 0E B6 B0 .....w...^.....
87 4D EC 62 C3 21 08 00 3B .M.b.!...;

```

```
05/26-20:51:42.446705 x.x.9.10:3259 -> 206.132.x.x:80
TCP TTL:128 TOS:0x10 ID:49632 DF
*****A* Seq: 0x2A9B3 Ack: 0x71434F6E Win: 0x1DCF
02 04 05 B4 0C 6E . . . . .n
```

**1. Source of trace**

Small subnet using a 3com ISDN Office Connect LAN Modem (OCLM) Router as a gateway to the Internet.

**2. Detect was generated by:**

SNORT version 1.6 using 05172kany.rules

**3. Probability the source address was spoofed**

IP address was not spoofed. Client on monitored subnet initiated the session.

**4. Description of attack:**

MISC-WinGate-1080-Attempt – Probe for WinGate on 1080.

**5. Attack mechanism:**

During an attempted IRC session a probe for port 1080 (WinGate) was sent to a host on the monitored subnet. Some IRC Chat applications will use the SOCKS proxy of WinGate for firewall access, by default WinGate installs a SOCKS proxy on port 1080. (WinGate allows networked computers to *simultaneously* share an Internet connection. Serves as a firewall, prohibiting intruders from accessing your network)

The session went something like this:

```
05/26-20:51:38.877035 – OK lets chat
05/26-20:51:38.878333 – I'm not set up to chat
05/26-20:51:38.882947 – Do you have a WinGate Proxy?
05/26-20:51:38.883195 – No!
05/26-20:51:38.906874 – Fine... be that way.
05/26-20:51:38.907048 – End already.
05/26-20:51:41.844661 – Want an HTTP session instead?
05/26-20:51:42.030326 – OK!
05/26-20:51:42.305892 – Here is some data.
05/26-20:51:42.446705 – Got it... Thank!
```

**6. Correlations:**

From WinGate Knowledge base (<http://kb.deerfield.com/index.cfm?a=1014&view=default&k=1>):  
Article Number: 1014

WinGate does not include an IRC proxy. For a general Internet Chat application to work through WinGate, WinGate requires a mapped link to your IRC server. The standard port for IRC connections is 6667, and this link will be installed when you install WinGate. Some IRC Chat applications will use the SOCKS proxy of WinGate for firewall access, by default WinGate installs a SOCKS proxy on port 1080.

**7. Evidence of active targeting:**

Sever was looking for WinGate.

**8. Severity:**

Severity = (criticality + lethality) – (system + net) = (2+2) – (5+2) = -3  
Criticality = 2: Target was a workstation.  
Lethality = 2: Port Probe during legitimate session.  
System = 5 WinNT with all current Hot fixes applied and tightened security.

Net = 2 Router was wide open with limited logging capabilities. Router was secured with a strong password consisting of upper and lower case characters, numbers, and special characters.

**9. Defensive recommendation:**

Block IRC connections (Port 6667) to and from local subnet.

**Multiple choice test question:**

The following detect is an example of which type of request?

```
05/26-20:51:38.882947 216.234.x.x:1502 -> x.x.9.10:1080
TCP TTL:50 TOS:0x0 ID:45706 DF
**S***** Seq: 0x6692F9C7 Ack: 0x0 Win: 0x4000
TCP Options => MSS: 1460
00 00 ..
```

- a) Telnet session
- b) WinGate Proxy
- c) TFTP request
- d) NetSpy Trojan probe

Answer is 'b'. By default WinGate installs a SOCKS proxy on port 1080.

**Detect 9**

*Time Stamp in all traces is GMT*

**Snort Alert:**

```
[**] spp_portscan: PORTSCAN DETECTED from 134.76.x.x [**]
05/27-06:21:19.259014
[**] SCAN-SYN FIN [**]
05/27-06:21:19.258763 134.76.x.x:53 -> x.x.9.7:53
TCP TTL:24 TOS:0x0 ID:39426
**SF***** Seq: 0x66776829 Ack: 0x3CEC5009 Win: 0x404
[**] SCAN-SYN FIN [**]
05/27-06:21:19.315656 134.76.x.x:53 -> x.x.9.10:53
TCP TTL:24 TOS:0x0 ID:39426
**SF***** Seq: 0x66776829 Ack: 0x3CEC5009 Win: 0x404

[**] spp_portscan: portscan status from 134.76.x.x: 2 connections across 2
hosts: TCP (2), UDP (0) STEALTH [**]
05/27-06:21:34.812862
[**] spp_portscan: End of portscan from 134.76.x.x [**]
05/27-06:21:44.812113

[**] spp_portscan: PORTSCAN DETECTED from 134.76.x.x [**]
05/27-06:32:49.341621
[**] SCAN-SYN FIN [**]
05/27-06:32:49.341358 134.76.x.x:111 -> x.x.9.7:111
TCP TTL:24 TOS:0x0 ID:39426
**SF***** Seq: 0x9DAEBC7 Ack: 0x7F20582C Win: 0x404
[**] SCAN-SYN FIN [**]
05/27-06:32:49.400815 134.76.x.x:111 -> x.x.9.10:111
TCP TTL:24 TOS:0x0 ID:39426
**SF***** Seq: 0x9DAEBC7 Ack: 0x7F20582C Win: 0x404

[**] spp_portscan: portscan status from 134.76.x.x: 2 connections across 2
hosts: TCP (2), UDP (0) STEALTH [**]
05/27-06:33:04.795436
```

```
[**] spp_portscan: End of portscan from 134.76.x.x [**]
05/27-06:33:14.794713

[**] spp_portscan: PORTSCAN DETECTED from 134.76.x.x [**]
05/27-06:45:48.412829
[**] SCAN-SYN FIN [**]
05/27-06:45:48.412579 134.76.x.x:53 -> x.x.9.7:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x20B44442 Ack: 0x63CC4D25 Win: 0x404

[**] SCAN-SYN FIN [**]
05/27-06:45:48.472450 134.76.x.x:53 -> x.x.9.10:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x20B44442 Ack: 0x63CC4D25 Win: 0x404

[**] spp_portscan: portscan status from 134.76.x.x: 2 connections across 2
hosts: TCP(2), UDP(0) STEALTH [**]
05/27-06:45:54.775640
[**] spp_portscan: End of portscan from 134.76.x.x [**]
05/27-06:46:04.775698
```

### Supporting Data from Snort PortScan Log:

```
May 27 06:21:19 134.76.x.x:53 -> x.x.9.7:53 SYNFIN **SF****
May 27 06:21:19 134.76.x.x:53 -> x.x.9.10:53 SYNFIN **SF****
May 27 06:32:49 134.76.x.x:111 -> x.x.9.7:111 SYNFIN **SF****
May 27 06:32:49 134.76.x.x:111 -> x.x.9.10:111 SYNFIN **SF****
May 27 06:45:48 134.76.x.x:53 -> x.x.9.7:53 SYNFIN **SF****
May 27 06:45:48 134.76.x.x:53 -> x.x.9.10:53 SYNFIN **SF****
```

### Supporting Data from Total log:

```
05/27-06:21:19.258763 134.76.x.x:53 -> x.x.9.7:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x66776829 Ack: 0x3CEC5009 Win: 0x404
00 00 00 00 00 00 .....

05/27-06:21:19.258907 x.x.9.7:53 -> 134.76.x.x:53
TCP TTL:128 TOS:0x0 ID:15152
****R*A* Seq: 0x0 Ack: 0x6677682B Win: 0x0
00 00 00 00 00 00 .....

05/27-06:21:19.315656 134.76.x.x:53 -> x.x.9.10:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x66776829 Ack: 0x3CEC5009 Win: 0x404
00 00 00 00 00 00 .....

05/27-06:21:19.315902 x.x.9.10:53 -> 134.76.x.x:53
TCP TTL:128 TOS:0x0 ID:19952
****R*A* Seq: 0x0 Ack: 0x6677682B Win: 0x0
47 45 54 20 2F 74 GET /t
```

\*\*\*\*\*

```
05/27-06:32:49.341358 134.76.x.x:111 -> x.x.9.7:111
TCP TTL:24 TOS:0x0 ID:39426
```

## SANS GIAC – GIAC Practical

```
**SF**** Seq: 0x9DAEBC7   Ack: 0x7F20582C   Win: 0x404
00 00 00 00 00 00      .....
```

```
05/27-06:32:49.343053 x.x.9.7:111 -> 134.76.x.x:111
TCP TTL:128 TOS:0x0 ID:15408
****R*A* Seq: 0x0   Ack: 0x9DAEBC9   Win: 0x0
00 00 00 00 00 00      .....
```

```
05/27-06:32:49.400815 134.76.x.x:111 -> x.x.9.10:111
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x9DAEBC7   Ack: 0x7F20582C   Win: 0x404
00 00 00 00 00 00      .....
```

```
05/27-06:32:49.401063 x.x.9.10:111 -> 134.76.x.x:111
TCP TTL:128 TOS:0x0 ID:37616
****R*A* Seq: 0x0   Ack: 0x9DAEBC9   Win: 0x0
47 45 54 20 2F 74      GET /t
```

.....

```
05/27-06:45:48.412579 134.76.x.x:53 -> x.x.9.7:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x20B44442   Ack: 0x63CC4D25   Win: 0x404
00 00 00 00 00 00      .....
```

```
05/27-06:45:48.414141 x.x.9.7:53 -> 134.76.x.x:53
TCP TTL:128 TOS:0x0 ID:15664
****R*A* Seq: 0x0   Ack: 0x20B44444   Win: 0x0
00 00 00 00 00 00      .....
```

```
05/27-06:45:48.472450 134.76.x.x:53 -> x.x.9.10:53
TCP TTL:24 TOS:0x0 ID:39426
**SF**** Seq: 0x20B44442   Ack: 0x63CC4D25   Win: 0x404
00 00 00 00 00 00      .....
```

```
05/27-06:45:48.472739 x.x.9.10:53 -> 134.76.x.x:53
TCP TTL:128 TOS:0x0 ID:55280
****R*A* Seq: 0x0   Ack: 0x20B44444   Win: 0x0
47 45 54 20 2F 74      GET /t
```

.....

### 1. Source of trace

Small subnet using a 3com ISDN Office Connect LAN Modem (OCLM) Router as a gateway to the Internet.

### 2. Detect was generated by:

SNORT version 1.6 using 05172kany.rules

### 3. Probability the source address was spoofed

This is classic information gathering. IP address was most likely not spoofed.

### 10. Description of attack:

SCAN-SYN FIN – Triggers when a series of TCP packets with both the SYN and FIN flags set have been sent to the same destination port on a number of different hosts.

Illegal flag set, SYN-FIN, looking for DNS (port 53) and port mapper (port 111). The sequence numbers of the packets sent to x.x.9.7 and x.x.9.10 are the same during a scan. Scan 1 Seq: 0x66776829, Scan2 Seq: 0x9DAEBC7, scan 3 Seq: 0x20B44442. This is evidence of use of a port scanning tool.

#### 11. Attack mechanism:

Illegal flag set, SYN-FIN, looking for DNS and port mapper.

This is definitely intrusive information gathering using a port scanning tool. This intruder is looking for DNS and UNIX machines, using a SYN-FIN probe to ports 53 and 111.

#### 12. Correlation's:

The CERT/CC Current Activity web page indicates that this type of probe is common and active in the wild.

domain 53/tcp 53/udp [IN-2000-04](#), Denial of Service Attacks using Nameservers

[CA-2000-03](#), Continuing Compromises of Nameservers

[CA-99-14](#), Multiple Vulnerabilities in BIND

[CA-98.05](#), Multiple Vulnerabilities in BIND

sunrpc 111/tcp 111/udp [CA-99-16](#), Buffer Overflow in Sun Solstice AdminSuite Daemon sadmind

[CA-99-12](#), Buffer overflow in amd

[CA-99-08](#), Buffer overflow in rpc.cmsd

[CA-99-05](#), Vulnerability in statd exposes vulnerability in automountd

[CA-98.12](#), Remotely Exploitable Buffer Overflow Vulnerability in mountd

[CA-98.11](#), Vulnerability in ToolTalk RPC service

#### 13. Evidence of active targeting:

Definitely looking for DNS and UNIX hosts.

#### 14. Severity:

Severity = (criticality + lethality) – (system + net) = (3+5) – (5+2) = 0

Criticality = 3: Targets were workstations. However, I bumped it to a three due to the fact that if there were servers on this subnet they would have been targeted.

Lethality = 5: If DNS or RPC were running, this could have turned into a lethal attack.

System = 5 Both machines had all current Hot fixes applied and tightened security. Port Mapper was not actually running.

Net = 2 Router was wide open with limited logging capabilities. Router was secured with a strong password consisting of upper and lower case characters, numbers, and special characters.

#### 15. Defensive recommendation:

Both machines had all of the latest security patches applied. The NT machine had security tightened using the “Microsoft Internet Information Server 4.0 Security Checklist” as a guide. The Windows 98 machine had file and print sharing disabled. Both machines contained nothing of value and the subnet was isolated.

The OCLM router had NAT disabled, allowed NetBIOS traffic to be passed, and was in an “always on” state. This is not a desired configuration. After the testing of snort was completed, the detects were collected and the router settings were securely set. If this were a production environment TCP/UDP traffic to ports 53 and 111 and the IP address of the intruder would have been blocked.

#### Multiple choice test question:

Distributed database that is used by TCP/IP applications.

- a) FTP
- b) SSH
- c) SMS
- d) DNS

Answer is ‘d’. The Domain Name System, or DNS, is a distributed database that is used by TCP/IP applications to map between hostnames and IP addresses, and to provide electronic mail routing information. (TCP/IP Illustrated, Volume 1, pg. 187)

## Detect 10

*Time Stamp in all traces is GMT*

### Snort Alert:

```
[**] spp_portscan: PORTSCAN DETECTED from 203.149.232.20 [**]
05/28-22:58:05.647859
[**] SCAN-SYN FIN [**]
05/28-22:58:05.647609 203.149.232.20:53 -> 208.58.9.7:53
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x43E3A07E Ack: 0x2BBFB2C3 Win: 0x404

[**] SCAN-SYN FIN [**]
05/28-22:58:05.740730 203.149.232.20:53 -> 208.58.9.10:53
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x43E3A07E Ack: 0x2BBFB2C3 Win: 0x404

[**] spp_portscan: portscan status from 203.149.232.20: 2 connections across
2 hosts: TCP(2), UDP(0) STEALTH [**]
05/28-22:58:11.124397
[**] spp_portscan: End of portscan from 203.149.232.20 [**]
05/28-22:58:21.123439

[**] spp_portscan: PORTSCAN DETECTED from 203.149.232.20 [**]
05/30-14:45:31.677045
[**] SCAN-SYN FIN [**]
05/30-14:45:31.676432 203.149.232.20:53 -> 208.58.9.7:53
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x7E91E99B Ack: 0x6448AB41 Win: 0x404

[**] SCAN-SYN FIN [**]
05/30-14:45:31.734656 203.149.232.20:53 -> 208.58.9.10:53
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x7E91E99B Ack: 0x6448AB41 Win: 0x404

[**] spp_portscan: portscan status from 203.149.232.20: 2 connections across
2 hosts: TCP(2), UDP(0) STEALTH [**]
05/30-14:45:42.964341
[**] spp_portscan: End of portscan from 203.149.232.20 [**]
05/30-14:45:52.964368
```

### Supporting Data from Snort PortScan Log:

```
May 28 22:58:05 203.149.232.20:53 -> 208.58.9.7:53 SYNFIN **SF****
May 28 22:58:05 203.149.232.20:53 -> 208.58.9.10:53 SYNFIN **SF****

May 30 14:45:31 203.149.232.20:53 -> 208.58.9.7:53 SYNFIN **SF****
May 30 14:45:31 203.149.232.20:53 -> 208.58.9.10:53 SYNFIN **SF****
```

### Supporting Data from Total log:

```
05/28-22:58:05.647609 203.149.232.20:53 -> 208.58.9.7:53
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x43E3A07E Ack: 0x2BBFB2C3 Win: 0x404
00 00 00 00 00 00 .....
```

```
05/28-22:58:05.647777 208.58.9.7:53 -> 203.149.232.20:53
TCP TTL:128 TOS:0x0 ID:15920
****R*A* Seq: 0x0 Ack: 0x43E3A080 Win: 0x0
00 00 00 00 00 00 .....

05/28-22:58:05.740730 203.149.232.20:53 -> 208.58.9.10:53
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x43E3A07E Ack: 0x2BBFB2C3 Win: 0x404
00 00 00 00 00 00 .....

05/28-22:58:05.741017 208.58.9.10:53 -> 203.149.232.20:53
TCP TTL:128 TOS:0x0 ID:52261
****R*A* Seq: 0x0 Ack: 0x43E3A080 Win: 0x0
47 45 54 20 2F 74 GET /t
```



```
05/30-14:45:31.676432 203.149.232.20:53 -> 208.58.9.7:53
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x7E91E99B Ack: 0x6448AB41 Win: 0x404
00 00 00 00 00 00 .....

05/30-14:45:31.676840 208.58.9.7:53 -> 203.149.232.20:53
TCP TTL:128 TOS:0x0 ID:16688
****R*A* Seq: 0x0 Ack: 0x7E91E99D Win: 0x0
00 00 00 00 00 00 .....

05/30-14:45:31.734656 203.149.232.20:53 -> 208.58.9.10:53
TCP TTL:27 TOS:0x0 ID:39426
**SF**** Seq: 0x7E91E99B Ack: 0x6448AB41 Win: 0x404
00 00 00 00 00 00 .....

05/30-14:45:31.734955 208.58.9.10:53 -> 203.149.232.20:53
TCP TTL:128 TOS:0x0 ID:33116
****R*A* Seq: 0x0 Ack: 0x7E91E99D Win: 0x0
47 45 54 20 2F 74 GET /t
```

**1. Source of trace**

Small subnet using a 3com ISDN Office Connect LAN Modem (OCLM) Router as a gateway to the Internet.

**2. Detect was generated by:**

SNORT version 1.6 using 05172kany.rules

**3. Probability the source address was spoofed**

This is classic information gathering. IP address was most likely not spoofed.

**16. Description of attack:**

SCAN-SYN FIN – Triggers when a series of TCP packets with both the SYN and FIN flags set have been sent to the same destination port on a number of different hosts. Illegal flag set, SIN-FIN, looking for DNS (port 53). The sequence numbers of the packets sent to x.x.9.7 and x.x.9.10 are the same during a scan. Scan 1 Seq: 0x43E3A07E, Scan2 Seq: 0x7E91E99B. Port scanning tool used in this scan.

**17. Attack mechanism:**

Illegal flag set, SIN-FIN, looking for DNS. This is definitely intrusive information gathering using a port scanning tool. This intruder is looking for DNS servers, using a SYN-FIN probe to port 53.



### 18. Correlation's:

The CERT/CC Current Activity web page indicates that this type of probe is common and active in the wild.

Domain 53/tcp 53/udp [IN-2000-04](#), Denial of Service Attacks using Nameservers

[CA-2000-03](#), Continuing Compromises of Nameservers

[CA-99-14](#), Multiple Vulnerabilities in BIND

[CA-98-05](#), Multiple Vulnerabilities in BIND

### 19. Evidence of active targeting:

Definitely looking for DNS servers.

### 20. Severity:

Severity = (criticality + lethality) – (system + net) = (3+5) – (5+2) = 0

Criticality = 3: Targets were workstations. However, I bumped it to a three due to the fact that if there were servers on this subnet they would have been targeted.

Lethality = 5: If DNS was running, this could have turned into a lethal attack.

System = 5 Both machines had all current Hot fixes applied and tightened security. Port Mapper was not actually running.

Net = 2 Router was wide open with limited logging capabilities. Router was secured with a strong password consisting of upper and lower case characters, numbers, and special characters.

### 21. Defensive recommendation:

Both machines had all of the latest security patches applied. The NT machine had security tightened using the “Microsoft Internet Information Server 4.0 Security Checklist” as a guide. The Windows 98 machine had file and print sharing disabled. Both machines contained nothing of value and the subnet was isolated.

The OCLM router had NAT disabled, allowed NetBIOS traffic to be passed, and was in an “always on” state. This is not a desired configuration. After the testing of snort was completed, the detects were collected and the router settings were securely set. If this were a production environment TCP/UDP traffic to ports 53 and the IP address of the intruder would have been blocked.

### Multiple choice test question:

Which statement about sequence numbers is true?

- a) Requires the ‘S’ flag set.
- b) Indicate fragmentation.
- c) Remains the same throughout the session.
- d) Identifies the byte in the stream of data.

Answer is ‘d’. The sequence number identifies the byte in the stream of data from the sending TCP to the receiving TCP. (TCP/IP Illustrated, Volume 1, pg. 226)

## APPENDIX

© SANS Institute 2000 - 2002, Author retains full rights.

## OCLM SECRETS - Bubba's Guide To The Telnet Interface

Last revised Jan. 30, 2000  
Bill Garfield [wdg@hal-pc.org](mailto:wdg@hal-pc.org) (aka "bubba")

### **Disclaimer:** (please read)

The information contained herein is an unofficial summary of undocumented and unsupported commands available in the Telnet Interface of the 3Com Office Connect LAN Modem (OCLM). It's important for the user to understand that the Telnet Interface is an 'Engineering Interface' that was not designed for the regular user. Information is presumed to be correct to the best of my knowledge but may contain inaccuracies.

The Telnet Interface is rich with numerous commands, some of which offer the user increased functionality or control not otherwise available through the web GUI interface. Telnet Interface commands should be considered case sensitive and command switches are position dependent. Beware that these commands provide no checking for proper syntax and are unforgiving. Command entries take immediate effect when entered, with no opportunity to confirm or cancel. No copyright to this work is claimed and this document is hereby released into the public domain, as-is. You are encouraged to link to this document rather than copy it, as there may be changes, updates, corrections or special announcements that you might otherwise miss.

**Attention:** Please note that the information which follows and the OCLM telnet interface in particular, is **UNSUPPORTED, UNDOCUMENTED and UNTESTED**. You therefore are both literally and figuratively "on your own" with this. I cannot make it any clearer than that. Please do not pester 3Com Tech Support for assistance with the telnet interface or these commands. They will likely only be able to tell you that the telnet interface is not supported and you will have wasted your time and theirs. Please do not e-mail me with your questions either. However, if you have questions (and I know some of you will) please be so kind as to post those questions in the ISDN discussion group [comp.dcom.isdn](http://comp.dcom.isdn) where everyone can see your question and everyone can also see the answer. In that way perhaps we can all learn more about this incredible product and each have a chance to contribute a little tidbit from time to time. This guide is not a 3Com document, nor was it produced with their help. Proceed entirely at your own risk. Note also that the telnet interface commands can and have changed with new firmware releases. Translation: What works today may not work tomorrow, or may work differently, or not at all.

**Password Protecting The Telnet Interface:** It should be noted before we begin that use of a \*reasonably secure\* configuration password is **strongly encouraged**, especially when using static IP addresses and permanent or always-on connections. Users should be aware that the telnet interface of the OCLM is ahead of any firewall software you might have installed at your end and therefore can be remotely accessed (hacked) quite easily over the WAN connection, even when dynamically assigned IP addresses are in use.

**Thanks to:** Everyone who has helped in putting this document together, especially to Steve Ferguson ([stf@altavista.net](mailto:stf@altavista.net)) for his excellent work with the RT command.

current as of firmware release 5.3.1

? Show list of commands <>

Summary: Displays a list of available telnet commands

Usage: Use this command to dump a list of the available commands to the screen

Syntax:

? | Dump the list of available commands for the telnet interface

**bacp**: Configure BACP Table <SP# enable/disable|?>

Summary: Allows the user to enable/disable the Bandwidth Allocation Control Protocol (BACP).

Usage: Use this command to enable or disable BACP/BAP for a given service provider. BAP/BACP is used to provide the dialup number for multilink. Note that it is possible for a multilink (2B) call to be negotiated without BACP. You may see this in the Current Call screen in the GUI interface.

Syntax:

**bacp 0 enable** | Enable BAP/BACP for Service provider 0  
**bacp 0 disable** | Disable BAP/BACP for Service provider 0

(Note: Defined service providers begin with provider 0)

**cv**: Call View <c|-|?>

Summary: Provides a user-intuitive log of call initiation/termination events.

Usage: Once enabled, CallView will track calls being brought up and torn down. It provides the date/time of call initiation, the reason for the call coming up, the reason for the call going down, and the referenced service provider. In order to start collection of CallView events, you must initially enable the feature by executing the command once. Note that 'cv' is also available as a component module within the event log (evlog) command below.

Syntax:

**cv** | The first time you type this command enables call logging  
| Subsequent use of this command dumps the log contents  
**cv c** | Clears the callview log  
**cv -** | Disables logging of Callview events

Note that executing the 'evlog' command will disable CallView logging and will additionally erase the contents of the cv buffer. These commands use the same memory space. However, if desired the 'cv' command can be enabled as a component of the event log. (See example in the evlog command below)

**evlog**: Event Log <c|+<Modules>|-<Modules>|0|s|?>

Summary: Provides logging capabilities for debugging of modules.

Usage: This command allows logging of events. The user can enable or disable specific event logging modules, allowing the user to capture events specific to a particular subsystem of the OCLM. The events are stored in a FIFO (circular) buffer, so that when the available memory space is exhausted the oldest events will be deleted in order to make room for new events. Multiple modules can be added/removed at the same time by using the '+' or '-' followed by the specific module names you wish to include or exclude. Log data will survive a hardware crash although the selected modules will be reset. This can be useful for

post-mortem debugging.

Syntax:

`evlog` | Dump all captured events in the event log - can result in  
| a copious amount of output  
`evlog c` | Clears the event log  
`evlog s` | Display all modules that can be captured  
`evlog 0` | Disable all modules to be captured except for EXCP (exception)  
`evlog +ppp` | Enable PPP logging. This will add PPP to the modules currently  
| being captured. From this point on, all ppp events will be  
| placed in the event log.  
`evlog -ppp` | Disable PPP logging. This will remove PPP from the list of  
| modules currently being captured. Note that events already  
| captured will remain in the log and new entries will be  
| appended.  
`evlog +cv` | Enables the CallView function from within the event log  
`evlog -cv` | Disables the CallView function  
`evlog 1` | Enables all modules (that's a 1 not an L)

Example of concatenated commands:

`evlog +ppp q931` | Enabled PPP and q931 logging  
`evlog -ppp q931` | Disables logging of PPP and q931 events

Modules "Exception" and "Call Trace" are enabled by default. An evlog dump will also include configuration information. This is useful if 3Com tech support asks for a certain evlog trace capture to help in diagnosing a problem. Beware that certain evlog trace commands, particularly 'ppp' will also capture your ISP user id and password in both hex and clear text (except when CHAP is used).

`exit`: Terminate Telnet Session

Summary: Exits command mode and closes the telnet session.

Usage: This command closes (disconnects) the management & telnet session. See also the quit command.

Syntax:

`exit` | Closes the telnet session.

`iconn`: Help page to manage ISDN connection

Summary: Shows a help page on how to prevent spurious calls

Usage: This command provides some tips on how to prevent unwanted calls from coming up. The two principle causes tend to be NetBIOS traffic from Windows workstations and DNS requests for the Windows workgroup name.

Syntax:

`iconn` | Shows the help page

Prepared by Martin J Seery

This is an informational command only.

**ldns**: Static DNS Entries <Index |paddr DNS-Name |c Index(clearing)|?>

Summary: Allows for the configuration of DNS entries into the built in DNS server.

Usage: This command allows the user to configure DNS entries into the local (internal) DNS server. These DNS translations will then be referenced by all workstations making requests to the built in DNS server. This can speed up surfing to the defined site(s) and also prevent unwanted calls (when the defined sites are on your local LAN or you wish to spoof them as being there to block certain sites)

Syntax:

**ldns** | Dumps the list of entries in the local DNS table

**ldns c 1** | Deletes entry 1 in the list of DNS entries

**ldns 0 192.168.1.1 www.myhost.com**

| Resolves the name www.myhost.com to 192.168.1.1 when  
| requests are made to the embedded DNS server.

**netbios**: Call Filter of NetBios Packet <a|c|d|?>

Summary: Configures the blocking of Netbios packets

Usage: As of firmware release 5.3.0, netbios filtering now has three valid states: Always, Call Initiation, and Disabled. If set to 'always', all Netbios traffic will be filtered (blocked). Calls will not be brought up as a result of Netbios traffic, nor will Netbios traffic be allowed to pass across the WAN after the call has been brought up. If set to 'call initiation', calls will not be brought up as a result of Netbios traffic, but Netbios traffic will be routed across the WAN if the call is already up. This is equivalent to the Netbios filtering provided in firmware release 5.2.0.

Syntax:

**netbios** | Display the current state of Netbios filtering

**netbios a** | Sets Netbios filtering to 'Always' This is the current  
| default setting in firmware vers. 5.3.1

**netbios c** | Sets Netbios filtering to 'Call Initiation' This was  
| the default setting in firmware vers. 5.2.0

**netbios d** | Disables Netbios filtering

See also the "Filter" command below

**reuseip**: Attempt reuse previous IP address in IPCP <on|off|?>

Summary: Enable attempts to reuse an IP address in IPCP negotiation

Usage: During IPCP negotiation in the initiation of the PPP session, the client has the opportunity to "suggest" an IP address to use. If this feature is enabled, the OCLM will automatically suggest the IP address it had previously used in the last call. Note well that this feature will only work if the host being dialed into allows the client to specify an IP address in IPCP negotiation and that IP address is currently available. Use of this feature may cause interoperability problems with certain routers and is therefore not recommended.

Syntax:

[reuseip on](#) | Enable "Attempt to Reuse IP" feature  
[reuseip off](#) | Disable "Attempt to Reuse IP" feature

[static](#): Static Table <c PC# Index u(dp)/t(cp) DstnPort | z(ero)|e|?>

Summary: Allows the creation of static NAT translation table mappings based on port number

Usage: This feature allows the user to setup servers on the private network. By using the static command, the user can configure the OCLM to route all packets destined to a specific port number to a specific workstation on the local LAN.

Syntax:

[static](#) | Displays the table of currently configured entries  
[static c 2 1 t 80](#) | Configures all traffic destined for TCP port 80 to  
| workstation 2  
[static c 3 1 t 23](#) | Configures all traffic destined for TCP port 23 to  
| workstation 3  
[static c 0 1 u 53](#) | Configures all traffic destined for UDP port 53 to  
| workstation 0  
[static c 5 1 0 0](#) | Delete the first entry under workstation 5  
[static z](#) | Delete all entries in the table  
[static e](#) | Shows several examples of how to use the command

If the user wants to configure multiple entries for a given workstation, the third parameter "Index" should be incremented. A maximum of ten port mappings can be configured for each workstation. Note that the 25 individual workstations are numbered 0 through 24 for configuration purposes.

[Xconfig](#): X Window Configuration <?(help)>

Although still shown in the command list, this command is no longer valid. Use the command 'static e' to see an example of how to use the static command to configure a workstation for X-Windows.

[auto](#): Service Provider Auto-Call <SP# on/off>

Summary: Configures 'Automatic Call Initiation' for service providers

Usage: This command controls whether or not automatic calls are brought up to a given service provider. This is equivalent to the "Allow Automatic Call Initiation?" field in the Service Provider page in the GUI. If disabled, traffic will not cause calls to be brought up. Calls to service providers will only be brought up via "Manual Call Control".

Syntax:

[auto 0 on](#) | Enable Automatic Call Initiation for the 1st service provider  
[auto 0 off](#) | Disable Automatic Call Initiation for the 1st service provider

Note that service providers count starting with provider 0

**bacpstate**: Determine the state of BACP for <Remoteld>

Summary: Determines whether Bandwidth Allocation Control Protocol (BACP) was negotiated on a call to a given service provider.

Usage: Bandwidth Allocation Control Protocol is used to provide the dial number for multilink calls. bacpstate will display whether or not BACP was negotiated for a current call to a given service provider.

Syntax:

**bacpstate 0** | Displays status of BACP negotiation for current call to  
| service provider 0

Note that "service provider 0" refers to your first service provider.

This is an informational/diagnostic command only.

**bapconn**: Send a BAP Setup msg to Sr <Remoteld>

Summary: Brings up the second channel on an already established multilink call.

Usage: This command allows the user to bring up the second channel on a multilink call. The command will only work if the service provider is configured for multilink and the first channel is already up. May also require that the service provider support this feature across multiple chassis (aka stacking).

Syntax:

**bapconn 0** | Brings up the second channel on call to service provider 0

**bapdisc**: Send a BAP Disc msg to Sr <Remoteld> <BChannel>

Summary: Disconnects the second channel on an multilink call to a service provider

Usage: This command allows the user to disconnect the second channel on a multilink call. The command will only work if the service provider is configured for multilink, and both channels are already up.

Syntax:

**bapdisc 0** | Disconnects second channel on call to service provider 0

**bw**: Display BW info

Summary: Displays bandwidth information for an established data call

Prepared by Martin J Seery



Usage: This command will show usage statistics regarding an established data call. This includes idle timers and overflow/underflow rates. This is a diagnostic/informational function only.

Syntax:

**bw**

**ccpstate**: Determine the state of CCP for <Remoteld>

Summary: CCP = Compression Control Protocol. The ccpstate command can be used to indicate whether Data Compression was negotiated for a call that is currently up to <Remoteld>

Usage: This command will inform the user whether or not compression was negotiated during the PPP session initiation. This information can also be seen in the "Current Call Statistics" page in the GUI under the "Data Call Options" field. This is a diagnostic/informational function only.

Syntax:

**ccpstate 0** | Display whether or not compression is in use on call to  
| service provider 0

**config**: Configure Profile <|w|c|d<Dstn #>|p<PC #>|s|?>

Summary: Allows for configuration of internal parameters directly into Flash ROM

Usage: This command provides the equivalent level of configuration as can be found in the GUI interface. Parameters correspond closely to the 'disp' command.

In my opinion, This is probably one of the more dangerous commands in the telnet interface. The user should exercise caution when using these commands, as there appears to be no checking for invalid parameters. Incorrect configuration of the LAN parameters in particular has been reported to render the unit inoperable. (Translation: It's probably not such a swell idea to use this method to configure your OCLM)

Parameters:

config l | LAN Parameters (IP address, subnet mask, etc)  
config w | WAN Parameters (ISDN numbers, SPIDs, etc)  
config c | Control Parameters (Data Call Parameters)  
config d | Service Providers (ISPs and Private Networks)  
config p | Workstations  
config s | Supplementary Services  
config g | Dial-In Global Parameters  
config u | Dial-In Users

Syntax example:

```
config l LANModem mypassword 0 192.168.2.1 225.225.225.224 1 1
```

The above example sets the OCLM password to 'mypassword' sets the domainname to '0' sets the OCLM IP address to 192.168.2.1, sets the OCLM subnet mask to 225.225.225.224, enables the DHCP Server and enables the DNS Server.

The format of the config command is as follows:

"config a b c d e f g h i j k l m n o p q r s t u v w" where:

a = profile to configure (d0 = 1st, d1=2nd, d2=3rd, d3=4th)

b = Name of ISP

c = Phone number #1

d = Phone number #2

e = PPP username

f = PPP password

g = DNS #1

h = DNS #2

i = Private Network Domain ('0' if none)

j = Use NAT? ('2' for Yes, '3' for No. )

k = WAN Link IP address

l = WAN Link Subnet Mask

m = Private ('1') or ISP ('0')

n = Use Private Network For Internet Access ('0' = No)

o = Private Network IP Address

p = Private Network Subnet mask

q = B-Channel Rate ('0' for 56K)

r = Multilink? ('0' for 56K LAN Modem)

s = Compression ('0' = off)

t = DN#2 is Alternate? ('0' = Yes)

u = No Autocall? ('0' Autodial enabled)

v = Intelligent NAT ('0' = enabled)

w = Default Workstation

**conn:** Connect ISDN call to <RemId>

Summary: Brings up call to a specific service provider

Usage: Use this command to initiate a call to a specific service provider. This is the functional equivalent of the Manual Call screen in the GUI interface.

Syntax:

**conn 0** | Brings up a call to first service provider

**defpc:** Default Workstation <SP# PC#>

Summary: Sets "Default Workstation for Incoming Packets" to a specific PC

Usage: This command correlates to the "Default Workstation for Incoming Packets" on the Service Provider Page in the GUI interface.

Syntax:

Prepared by Martin J Seery

**defpc 0 2** | Sets the Default Workstation for the 1st service provider  
| to the third workstation

Note, service providers and workstations begin with 0 as the 1st iteration. Your first service provider is '0'  
and your first workstation is '0'

**disc**: Disconnect ISDN call to <RemId BChan (optional)>

Summary: Disconnects a call to a specific service provider

Usage: Use this command to disconnect a call to a specific service provider. A call must be up to this  
service provider in order for the command to be valid.

Syntax:

**disc 0** | Disconnects a call to first service provider

**disp**: Display information <b|d|i|p|s|v|?>

Summary: Displays various information regarding the OCLM

Usage: This command will display various types of information regarding the OCLM. The most useful  
arguments are 'p' and 'v'. The 'p' argument will show the OCLM configuration. This includes the display of  
the LAN profile, the WAN profile, the service providers, and the workstations. The 'v' parameter shows  
version information such as firmware versions, boot code versions, serial number, etc. Most of the  
equivalent information can be found in the "Statistics->System" page in the GUI interface.

Syntax:

**disp b** | Show buffer utilization  
**disp d** | Show Routing table  
**disp i** | Show network interfaces  
**disp p** | Show configuration profile  
**disp s** | Show source routing table  
**disp v** | Show version information  
**disp ?** | Show a list of disp command switches

This is an informational/diagnostics command only

**dm**: Display Memory <Address (hex)> <# of Bytes (256 max)>

Summary: Displays raw memory

Usage: This command will dump the contents of RAM to the screen. It appears to have no useful  
purpose.

Syntax:

**dm 0 10** | Display 10 bytes, starting at address 0x00

Prepared by Martin J Seery

This is an informational/diagnostics command only

**dring**: Distinctive Ringing for DN1 <on|off>

Summary: Enables Distinctive Ringing service for first phone number

Usage: This command allows the Distinctive Ringing supplementary service to be enabled for the first phone number.

Syntax:

**dring on** | Enables distinctive ringing for first phone number

**dring off** | Disables distinctive ringing

It appears there is no means by which to enable distinctive ringing on the second phone number. However, you can enter your SPIDS and Directory Numbers in reverse order and accomplish the same thing. The phone co. doesn't care what order the SPIDS are in. Just be sure to match the SPIDS with the DIR Nos. In other words, if you reverse the SPID order, you must also reverse the directory numbers.

**eed**: Dumps eeprom to the screen

Summary: Dumps the contents of the internal EEPROM

Usage: This command will show the contents of the EEPROM in the OCLM. It would appear that the EEPROM is used to store the Product ID, Ethernet Mac address of the OCLM and the serial number. The equivalent information can be found in the "Statistics->System" page in the GUI. There appears to be no command to modify the contents of the EEPROM.

Syntax:

**eed** | Display EEPROM contents

This is an informational/diagnostics command only

**eth**: Display Ethernet driver information <0|1|2|3|4|5|6|?>

Summary: Displays information regarding the Ethernet driver

Usage: This command displays information regarding the Ethernet driver. Shows information such as TX errors, collisions, interrupts, and state of internal registers of the Ethernet controller.

Syntax:

**eth 0** | Displays all Ethernet driver information

**eth 1** | Displays all Ethernet interrupt events

**eth 2** | Displays all Ethernet hardware registers

**eth 3** | Displays Ethernet control block

**eth 4** | Displays Ethernet MIB table

Prepared by Martin J Seery

[eth 5](#) | Resets Ethernet interrupt event counters  
[eth 6](#) | Resets Ethernet MIB table  
[eth ?](#) | Displays the above command list

This is primarily an informational/diagnostic command. MIB stands for Management Information Base, not "Men In Black".

[ftp](#): Start FTP server

Summary: Places the OCLM in firmware download mode (Alert LED will flash).

Usage: This command will start the internal FTP server in the OCLM. It is equivalent to the "Enter Firmware Download Mode" command on the Maintenance page in the GUI interface. The "Upgrade Wizard" also does this automatically.

Syntax:

[ftp](#) | Starts the FTP server in the OCLM

Note: If no ftp file transfer activity occurs the ftp server will automatically time out after a period of 5 minutes. Also the Alert LED (!) will flash while the ftp server is running. Note too that the user profile 'prof.bin' may be uploaded and downloaded through the ftp server. Just remember that your prof.bin file must have been created with the same vers. of firmware that the OCLM is now running.

Additional note: If you have firewall software running on your PC, the firewall may possibly block or prevent the ftp file transfer or firmware upgrade if it intercepts or blocks tcp/udp packets on ports 20 and 21.

[if](#): Set Network Interface Parameters

Summary: Unknown

Usage: Unknown

Syntax: Unknown

[loop](#): Loop back B channel <o(n)|(of) f 1|2>

Summary: Places the ISDN bearer channel in digital loopback for testing

Usage: This command is used to initiate a digital loopback on either of the ISDN B (bearer) channels, usually for bit error rate (BERT) testing. Note that the phone company can send a diagnostic code to your NT1 to loop up your bearer channels and does not need (or use) this command. They can similarly cause the bearer channels to go in and out of loopback mode on US domestic models having a built-in NT-1 (U-interface).

Syntax:

[loop o 1](#) | Puts bearer channel 1 in digital loopback

[loop f 1](#) | Cancels digital loopback of bearer channel 1

**nat:** Display NAT table <0|1>

Summary: Displays the internal table containing NAT state information

Usage: This command shows internal state information regarding all current sessions running in NAT.

Syntax:

**nat 0**  
**nat 1**

This is an informational/diagnostics command only.

**pm:** Patch Memory <Address (hex)> <Bytes (hex, 16 max)>

Summary: Unknown

Usage: Unknown

Syntax: Unknown

**quit:** Quit Command mode

Summary: Exits Command Mode

Usage: Exits command mode but leaves the telnet session open. See also the exit command.

Syntax:

**quit**

**reset:** Reset <f|n|?>

Summary: Reboot the OCLM

Usage: If the 'f' parameter is used, the OCLM will be rebooted and the internal configuration will be reset to factory defaults. This includes clearing of all service providers, workstation parameters, and the resetting of the OCLM IP address to factory default. If the 'n' parameter is used, only a warm boot is performed and all configuration data will be saved.

Syntax:

**reset f** | Perform a factory reset (loads default values)  
**reset n** | Perform a non-destructive warm boot

**route:** Service Provider Auto-Route <SP# on/off>

Prepared by Martin J Seery

Summary: Enables the "Intelligent NAT" feature for the service provider

Usage: This command is the functional equivalent of the Service Provider->Miscellaneous->Enable Intelligent NAT function in the GUI interface

Syntax:

`route 0 on` | Enables Intelligent NAT for the first Service provider  
`route 0 off` | Disables Intelligent NAT for the first Service provider

Note: Service provider '0' refers to your first provider

`rt`: Set pNA+ Routing Table

Summary: Adds or deletes static routes to the routing table

Usage: to add static routes to the routing table (for when you have multiple internal networks behind the OCLM and are running a router internally).

Syntax:

(to add a static route)

`rt add {network-hex} {gateway-hex} {netmask-hex} {interface} {type} {proto}`

(to delete a static route)

`rt delete {network-hex} {gateway-hex} {netmask-hex} {interface} {type} {proto}`

Examples:

Add a static route for the 192.168.1.0 network using 192.168.2.21 as a gateway. The '1' is interface 1 (the LAN interface on the OCLM). The '3' is the type of route, which is direct connected. The 'disp d' command shows Type as 4 after I add it, but 3 seems to work fine. The '2' is type of protocol, which appears to always be 2. I'm not sure how they differentiate between types 2 and 4, but I haven't had any need to do so.

```
rt add c0a80100 c0a80215 fffff00 1 3 2
```

Delete the just-added static route:

```
rt delete c0a80100 c0a80215 fffff00 1 3 2
```

**How to convert dotted-decimal address to hex:**

To find the correct hex numbers, each octet should be converted to hex. The simplest way to do this is to use a scientific calculator (the Windows calculator will do), but it can be done by hand. Divide each octet by 16 and keep the remainder. Then convert the result and remainder to a single-character value as shown:

0-9: 0-9

10: A

11: B

12: C

13: D

14: E

15: F

Example:

192/16 = 12, remainder 0  
192 decimal = C0 hex

168/16 = 10, remainder 8  
168 decimal = A8 hex

1/16 = 0, remainder 1  
1 decimal = 01 hex

Thus, something like 192.168.1.0 in decimal is C0A80100 in hex. The most common netmask will be a Class C address, which is 255.255.255.0. This translates to FFFFFFF0 in hex.

**sf**: Send a UDP frame

Summary: Unknown

Usage: Unknown

Syntax: Unknown

**stats**: Display Diagnostics/Statistics information <b|c|d|i||p|s|w|?>

Summary: Displays various statistics about the OCLM

Usage: This command will show statistics on various aspects of the OCLM. This includes information on Current Calls, Service Providers, and Last Call Statistics. The information presented is equivalent to that presented in the Statistics pages in the GUI interface.

Syntax:

**stats b** | Display BRI (ISDN Line) statistics  
**stats c** | Display Current Call Statistics  
**stats d** | Display statistics on service providers  
**stats i** | Not implemented  
**stats l** | Display Last Call Statistics  
**stats p** | Not implemented  
**stats s** | Display CPU Utilization  
**stats w** | Not implemented  
**stats ?** | Displays command switch functions b|c|d||s only

This is an informational/diagnostics command only

**sw**: Run SPID Wizard <DirNum1> <DirNum2>

Summary: Runs the SPID Wizard using the two phone numbers as input

Usage: This command runs the SPID Wizard. It is functionally equivalent to the SPID Wizard in the GUI

Prepared by Martin J Seery



interface.

Syntax:

`sw 2815551212 2815551213` | Runs the spid wizard using the phone  
| numbers 281-555-1212 and 281-555-1213  
| as input

`tget`: Display system current date and time

Summary: Displays the date and time in the OCLM

Usage: This command will display the current date and time of the real-time clock in the OCLM.

Syntax:

`tget` | Displays the current date and time

`tset`: Set current date and time <year month day hour minute second>

Summary: Sets the date and time of the real-time clock in the OCLM

Usage: This command sets the internal real-time-clock to the specified time.

Syntax:

`tset 2000 01 30 12 50 02` | Sets time to 01/30/00 12:50:02 pm

`filter`: Configure to filter packets on specific ports

Summary: Provides packet call filtering capability (finally!)

Usage: This command allows the user to create filters, to filter by protocol, port number, whether or not to filter inbound outbound or both, and whether to filter call initiation, or always filter (block) these packets.

*Errata 30-Jan-00: It has come to my attention that the direction or bound arguments are presently non-functional and have no effect. Filter statements therefore effect packets in both directions. netBIOS filtering may also behave unpredictably when entered through the telnet interface.*

Arguments:

command: a - add entry to table  
l - list the table  
d - delete an entry from the table  
? - display help  
port: nn - port number to block  
protocol: t - block TCP packets  
u - block UDP packets  
c - block ICMP packets  
g - block IGMP packets

Prepared by Martin J Seery

e - block GRE packets (PPTP)  
b - block both TCP and UDP packets  
direction: i - inbound  
o - outbound  
b - both inbound and outbound  
blocktype: a - Always filter (block)  
c - Only block call initiation

Syntax:

`filter ?` | Displays onscreen help (poorly formatted display)  
`filter l` | Show the current active filter list  
`filter a 7 c i a` | Adds a filter to always block incoming ICMP packets  
`filter a 80 t o a` | Adds a filter entry to always block outbound TCP  
| HTTP packets  
`filter a 80 t i a` | Adds a filter entry to always block inbound TCP  
| HTTP packets  
`filter a 80 u o a` | Adds a filter entry to always block outbound UDP  
| packets on port 80 (HTTP)  
`filter a 80 t o c` | Adds a filter entry to block outbound TCP  
| HTTP packets. This blocks call initiation only.  
| Once the call is up, these packets will be routed  
`filter d 0` | Deletes the first entry from the filter list

`lastpacket`: Show info regarding last packet in and out of the OCLM

Summary: Display information regarding the last inbound/outbound packets

Usage: This command will show the protocol and port number of the last packet to pass through the OCLM. This can be helpful in determining what traffic is keeping a call from going down.

Syntax:

`lastpacket`

This is an informational/diagnostic command only

`erase`: Erase/Clear all the filter list in flash

Summary: Deletes all entries from the filter list

Usage: **WARNING** - This command will immediately delete **all** entries from the filter list. The entire list will be cleared, restoring it to its factory-default state, without warning. Netbios filtering will also be disabled.

Syntax:

`erase`

`addfilter`: `addfilter <s|e|d> <b|m>`

Summary: Use to show|enable|disable Broadcast|Multicast packet routing

Usage: Use this command to display and optionally control (enable/disable) routing of broadcast and/or multicast data packets.

Syntax:

`addrfilter e m` | Multicast pkts from WAN will be dropped (filter enabled)  
`addrfilter d m` | Multicast pkts from WAN will be forwarded across WAN link  
| (filtering disabled)

This command appears to have been added in firmware version 5.3.1

Click on this link to go to 3Com's OCLM home page:

<http://www.3com.com/support/docs/lanmodem/welcome.html>

END

### **Ports used by trojans (2000-05-20)**

The table shows examples of existing trojans and ports being used. The lower ports are often used by trojans that steals password and either mail the passwords to attackers or hide them in FTP-directories. The higher ports are often used by Remote Access trojans that can be reached over the network. If you find probes directed against ports normally not used, it may be someone trying to connect to a trojan inside your network. I hope this list will be of some help for you.

During 1999 at last 290 new Windows trojans was released on the Internet. During the first four months of this year we have found almost 60 new trojans written during the first four months year 2000. All new Unix trojans (mostly Linux, Sun and BSD) are now also included on this list.

This list was updated 2000-05-20 and includes some 80 new entries compared with the last list. All new entries are in bold types.

Please observe that all ports are TCP ports unless so labled. The table have been compiled by Joakim von Braun (von Braun Consultants), who also answers any questions.

Default ports used by some known trojan horses:

port 2 - Death

port 21 - Back Construction, Blade Runner, Doly Trojan, Fore, FTP trojan, Invisible FTP, Larva, MBT, Motiv, Net Administrator, Senna Spy FTP Server, WebEx, WinCrash

port 23 - Tiny Telnet Server, Truva Atl

port 25 - Aji, Antigen, Email Password Sender, Happy 99, I Love You, Kuang 2, Magic Horse, Moscow Email Trojan, Naebi, NewApt, ProMail trojan, Shtrilitz, Stealth, Tapiras, Terminator, WinPC, WinSpy

port 31 - Agent 31, Hackers Paradise, Masters Paradise

port 41 - DeepThroat

port 48 - DRAT

port 50 - DRAT  
port 59 - DMSetup  
port 79 - Firehotcker  
port 80 - Back End, Executor, Hooker, RingZero  
port 99 - Hidden Port  
port 110 - ProMail trojan  
port 113 - Invisible Identd Deamon, Kazimas  
port 119 - Happy 99  
port 121 - JammerKillah  
port 123 - Net Controller  
port 146 - Infector  
port 146 (UDP) - Infector  
port 170 - A-trojan  
port 421 - TCP Wrappers  
port 456 - Hackers Paradise  
port 531 - Rasmin  
port 555 - Ini-Killer, NeTAdministrator, Phase Zero, Stealth Spy  
port 606 - Secret Service  
port 666 - Attack FTP, Back Construction, NokNok, Cain & Abel,  
Satanz Backdoor, ServeU, Shadow Phyre  
port 777 - Aim Spy  
port 808 - WinHole  
port 911 - Dark Shadow  
port 999 - DeepThroat, WinSatan  
port 1000 - Der Spacher 3  
port 1001 - Der Spacher 3, Silencer, WebEx  
port 1010 - Doly Trojan  
port 1011 - Doly Trojan  
port 1012 - Doly Trojan  
port 1015 - Doly Trojan  
port 1016 - Doly Trojan  
port 1020 - Vampire  
port 1024 - NetSpy  
port 1042 - Bla  
port 1045 - Rasmin  
port 1050 - MiniCommand  
port 1080 - WinHole  
port 1081 - WinHole  
port 1082 - WinHole  
port 1083 - WinHole  
port 1090 - Xtreme  
port 1095 - RAT  
port 1097 - RAT  
port 1098 - RAT  
port 1099 - BFEvolution, RAT  
port 1170 - Psyber Stream Server, Streaming Audio trojan, Voice  
port 1200 (UDP) - NoBackO  
port 1201 (UDP) - NoBackO  
port 1207 - SoftWAR  
port 1212 - Kaos  
port 1225 - Scarab  
port 1234 - Ultors Trojan  
port 1243 - BackDoor-G, SubSeven, SubSeven Apocalypse, Tiles  
port 1245 - VooDoo Doll  
port 1255 - Scarab  
port 1256 - Project nEXT

port 1269 - Mavericks Matrix  
port 1313 - NETrojan  
port 1349 (UDP) - BO DLL  
port 1492 - FTP99CMP  
port 1509 - Psyber Streaming Server  
port 1524 - Trinoo  
port 1600 - Shivka-Burka  
port 1777 - Scarab  
port 1807 - SpySender  
port 1969 - OpC BO  
port 1981 - Shockrave  
port 1999 - BackDoor, TransScout  
port 2000 - Der Spaeher 3, Insane Network, TransScout  
port 2001 - Der Spaeher 3, TransScout, Trojan Cow  
port 2002 - TransScout  
port 2003 - TransScout  
port 2004 - TransScout  
port 2005 - TransScout  
port 2023 - Ripper  
port 2080 - WinHole  
port 2115 - Bugs  
port 2140 - Deep Throat, The Invasor  
port 2155 - Illusion Mailer  
port 2283 - HVL Rat5  
port 2300 - Xplorer  
port 2565 - Striker  
port 2583 - WinCrash  
port 2600 - Digital RootBeer  
port 2716 - The Prayer  
port 2773 - SubSeven  
port 2801 - Phineas Phucker  
port 3000 - Remote Shutdown  
port 3024 - WinCrash  
port 3128 - RingZero  
port 3129 - Masters Paradise  
port 3150 - Deep Throat, The Invasor  
port 3456 - Teror Trojan  
port 3459 - Eclipse 2000, Sanctuary  
port 3700 - Portal of Doom  
port 3791 - Eclypse  
port 3801 (UDP) - Eclypse  
port 4000 - Skydance  
port 4092 - WinCrash  
port 4242 - Virtual hacking Machine  
port 4321 - BoBo  
port 4444 - Prosiak  
port 4567 - File Nail  
port 4590 - ICQTrojan  
port 5000 - Bubbel, Back Door Setup, Sockets de Troie  
port 5001 - Back Door Setup, Sockets de Troie  
port 5011 - One of the Last Trojans (OOTLT)  
port 5031 - NetMetropolitan  
port 5031 - NetMetropolitan  
port 5321 - Firehotcker  
port 5343 - wCrat  
port 5400 - Blade Runner, Back Construction

port 5401 - Blade Runner, Back Construction  
port 5402 - Blade Runner, Back Construction  
port 5550 - Xtcp  
port 5512 - Illusion Mailer  
port 5555 - ServeMe  
port 5556 - BO Facil  
port 5557 - BO Facil  
port 5569 - Robo-Hack  
port 5637 - PC Crasher  
port 5638 - PC Crasher  
port 5742 - WinCrash  
port 5882 (UDP) - Y3K RAT  
port 5888 - Y3K RAT  
port 6000 - The Thing  
port 6272 - Secret Service  
port 6400 - The Thing  
port 6667 - Schedule Agent  
port 6669 - Host Control, Vampyre  
port 6670 - DeepThroat, BackWeb Server, WinNuke eXtream  
port 6711 - SubSeven  
port 6712 - Funny Trojan, SubSeven  
port 6713 - SubSeven  
port 6723 - Mstream  
port 6771 - DeepThroat  
port 6776 - 2000 Cracks, BackDoor-G, SubSeven  
port 6838 (UDP) - Mstream  
port 6912 - Shit Heep (not port 69123!)  
port 6939 - Indoctrination  
port 6969 - GateCrasher, Priority, IRC 3  
port 6970 - GateCrasher  
port 7000 - Remote Grab, Kazimas, SubSeven  
port 7001 - Freak88  
port 7215 - SubSeven  
port 7300 - NetMonitor  
port 7301 - NetMonitor  
port 7306 - NetMonitor  
port 7307 - NetMonitor  
port 7308 - NetMonitor  
port 7789 - Back Door Setup, ICKiller  
port 7983 - Mstream  
port 8080 - RingZero  
port 8787 - Back Orifice 2000  
port 8897 - HackOffice  
port 8988 - BacHack  
port 8989 - Recon  
port 9000 - Netadministrator  
port 9325 (UDP) - Mstream  
port 9400 - InCommand  
port 9872 - Portal of Doom  
port 9873 - Portal of Doom  
port 9874 - Portal of Doom  
port 9875 - Portal of Doom  
port 9876 - Cyber Attacker, RUX  
port 9878 - TransScout  
port 9989 - iNi-Killer  
port 9999 - The Prayer

port 10067 (UDP) - Portal of Doom  
port 10086 - Syphillis  
port 10101 - BrainSpy  
port 10167 (UDP) - Portal of Doom  
port 10520 - Acid Shivers  
port 10607 - Coma  
port 10666 (UDP) - Ambush  
port 11000 - Senna Spy  
port 11050 - Host Control  
port 11223 - Progenic trojan, Secret Agent  
port 12076 - Gjamer  
port 12223 - Hack '99 KeyLogger  
port 12345 - GabanBus, My Pics, NetBus, Pie Bill Gates, Whack Job,  
X-bill  
port 12346 - GabanBus, NetBus, X-bill  
port 12349 - BioNet  
port 12361 - Whack-a-mole  
port 12362 - Whack-a-mole  
port 12623 (UDP) - DUN Control  
port 12624 - Buttman  
port 12631 - WhackJob  
port 12754 - Mstream  
port 13000 - Senna Spy  
port 15104 - Mstream  
port 16660 - Stacheldracht  
port 16484 - Mosucker  
port 16772 - ICQ Revenge  
port 16969 - Priority  
port 17166 - Mosaic  
port 17300 - Kuang2 The Virus  
port 17777 - Nephron  
port 18753 (UDP) - Shaft  
port 19864 - ICQ Revenge  
port 20001 - Millennium  
port 20002 - AcidkoR  
port 20034 - NetBus 2 Pro, Whack Job  
port 20203 - Chupacabra  
port 20331 - Bla  
port 20432 - Shaft  
port 20432 (UDP) - Shaft  
port 21544 - Girlfriend  
port 21554 - WinSp00fer  
port 22222 - Prosiak  
port 23023 - Logged  
port 23432 - Asylum  
port 23456 - Evil FTP, Ugly FTP, Whack Job  
port 23476 - Donald Dick  
port 23476 (UDP) - Donald Dick  
port 23477 - Donald Dick  
port 26274 (UDP) - Delta Source  
port 27374 - SubSeven  
port 27444 (UDP) - Trinoo  
port 27573 - SubSeven  
port 27665 - Trinoo  
port 29891 (UDP) - The Unexplained  
port 30001 - TerrOr32

port 30029 - AOL Trojan  
port 30100 - NetSphere  
port 30101 - NetSphere  
port 30102 - NetSphere  
port 30103 - NetSphere  
port 30103 (UDP) - NetSphere  
port 30303 - Sockets de Troie  
port 30947 - Intruse  
port 30999 - Kuang2  
port 31335 (UDP) - Trinoo  
port 31336 - Bo Whack, ButtFunnel  
port 31337 - Baron Night, BO client, BO2, Bo Facil  
port 31337 (UDP) - BackFire, Back Orifice, DeepBO  
port 31338 - NetSpy DK, ButtFunnel  
port 31338 (UDP) - Back Orifice, DeepBO  
port 31339 - NetSpy DK  
port 31666 - BOWhack  
port 31785 - Hack´a´Tack  
port 31787 - Hack´a´Tack  
port 31788 - Hack´a´Tack  
port 31789 (UDP) - Hack´a´Tack  
port 31791 (UDP) - Hack´a´Tack  
port 31792 - Hack´a´Tack  
port 32100 - Peanut Brittle, Project nEXT  
port 32418 - Acid Battery  
port 33333 - Blakharaz, Prosiak  
port 33577 - PsychWard  
port 33777 - PsychWard  
port 33911 - Spirit 2001a  
port 34324 - BigGluck, TN  
port 34555 (UDP) - Trinoo (Windows)  
port 35555 (UDP) - Trinoo (Windows)  
port 37651 - YAT  
port 40412 - The Spy  
port 40421 - Agent 40421, Masters Paradise  
port 40422 - Masters Paradise  
port 40423 - Masters Paradise  
port 40426 - Masters Paradise  
port 41666 - Remote Boot  
port 41666 (UDP) - Remote Boot  
port 44444 - Prosiak  
port 47262 (UDP) - Delta Source  
port 50505 - Sockets de Troie  
port 50766 - Fore, Schwindler  
port 51996 - Cafeini  
port 52317 - Acid Battery 2000  
port 53001 - Remote Windows Shutdown  
port 54283 - SubSeven  
port 54320 - Back Orifice 2000  
port 54321 - School Bus  
port 54321 (UDP) - Back Orifice 2000  
port 57341 - NetRaider  
port 58339 - ButtFunnel  
port 60000 - Deep Throat  
port 60068 - Xzip 6000068  
port 61348 - Bunker-Hill



port 61466 - Telecommando  
port 61603 - Bunker-Hill  
port 63485 - Bunker-Hill  
port 65000 - Devil, Stacheldracht  
port 65432 - The Traitor  
port 65432 (UDP) - The Traitor  
port 65535 - RC

---

Please observe that the ports 34555 and 35555 concerns the Windows version of Trinoo, not the Sun version.

This page was last uploaded Thursday, 25-May-00 20:00:16 MET DST.

If you have any questions or information about actual trojan attacks or ports used by trojans not listed above, please contact Joakim von Braun at

<joakim.von.braun@risab.se>.

Copyright (c) von Braun Consultants and Simovits Consulting. The above text may be cited provided that the source of the information is acknowledged. Nyheter

1999-02

"Trojanlistan"

Simovits Consulting, Wenner-Gren Center Sveavägen 166, 113 46  
Stockholm Sweden

Telephone +46 - (0) 8 - 728 33 69 - Fax +46 - (0) 8 - 728 3352

Internet: <www.simovits.com>

# Upcoming Training

Click Here to  
**{Get CERTIFIED!}**



Security Awareness Summit & Training 2017	Nashville, TN	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Memphis SEC504	Memphis, TN	Aug 21, 2017 - Aug 26, 2017	Community SANS
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
Mentor Session AW - SEC504	Milwaukee, WI	Aug 23, 2017 - Sep 29, 2017	Mentor
Mentor Session AW - SEC504	New York, NY	Aug 24, 2017 - Sep 08, 2017	Mentor
Mentor Session - SEC504	Denver, CO	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS vLive - SEC504: Hacker Tools, Techniques, Exploits and Incident Handling	SEC504 - 201709,	Sep 05, 2017 - Oct 12, 2017	vLive
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS Dublin 2017	Dublin, Ireland	Sep 11, 2017 - Sep 16, 2017	Live Event
Mentor AW - SEC504	Santa Clara, CA	Sep 11, 2017 - Sep 22, 2017	Mentor
Mentor Session - SEC504	Arlington, VA	Sep 20, 2017 - Nov 01, 2017	Mentor
Community SANS Columbia SEC504	Columbia, MD	Sep 25, 2017 - Sep 30, 2017	Community SANS
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS SEC504 at Cyber Security Week 2017	The Hague, Netherlands	Sep 25, 2017 - Sep 30, 2017	Live Event
Mentor Session - SEC504	Boston, MA	Sep 26, 2017 - Nov 07, 2017	Mentor
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Mentor Session AW - SEC504	Houston, TX	Oct 02, 2017 - Dec 11, 2017	Mentor
Mentor Session - SEC504	Columbia, SC	Oct 03, 2017 - Nov 14, 2017	Mentor
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event
Community SANS Chicago SEC504	Chicago, IL	Oct 09, 2017 - Oct 14, 2017	Community SANS