*** Northcutt, this is a fine job., the student has done as good as you can do without codebits. Three is probably collateral effects of a DOS against IRC servers. Solid analysis process, pretty clear write up, I really enjoyed 2, that must have been fun. Some research into source addresses, not much into tools or techniques. 77 *

GIAC
Certified Intrusion Analyst (GCIA) Practical
10 Network Detects with Analysis
04-17-00
Roger W. Lutz

Network Detect Notes: Some of these detects are from my network via a Cisco PIX firewall log and a Cisco 2611 router ACL log. The remainder of the detects are from the SANS web site as noted. Addresses in detects from my network have been sanitized to protect the innocent by x-ing out the first three octets. Detects from the SANS site are left as is. Detects captured on firewall were captured while ACL was temporarily relaxed on the edge router. All threat level assessments are based on the premise that the detects were captured outside of the firewall / edge router with ACL or at least in the firewall itself. Obviously if any of these types of activities are in the internal network there is a much higher threat assessment to be made.

Notes on our network: Our network has multiple public addresses for our systems and we NAT (Network Address Translate) most users through one global address. We have not yet set up an IDS like SNORT or SHADOW so currently we manually review via SYSLOG with Open System Solutions Private I.

Network Detect 1 Netbios Name Service Scan

```
Apr  4 05:09:32.832: %SEC-6-IPACCESSLOGP: list 101 denied udp xxx.xxx.xxx.132(137) -> xxx.xxx.xxx.84(137), 1 packet
Apr  4 05:09:41.855: %SEC-6-IPACCESSLOGP: list 101 denied udp xxx.xxx.xxx.132(137) -> xxx.xxx.xxx.85(137), 1 packet
Apr  4 05:09:50.882: %SEC-6-IPACCESSLOGP: list 101 denied udp xxx.xxx.xxx.132(137) -> xxx.xxx.xxx.86(137), 1 packet
Apr  4 05:09:59.981: %SEC-6-IPACCESSLOGP: list 101 denied udp xxx.xxx.xxx.132(137) -> xxx.xxx.xxx.87(137), 1 packet
Apr  4 05:15:13.358: %SEC-6-IPACCESSLOGP: list 101 denied udp xxx.xxx.xxx.132(137) -> xxx.xxx.xxx.88(137), 1 packet
```

Active Targeting: Yes, this was targeted at devices on my network so Northcutt's rule applies. However I feel pretty confident that since the scan so uniformly moves across my network, it is probably scanning many others and not just mine.

History: NETBIOS is a constant source of detects and we deny them early in our edge router. This is the first time I've seen the source scanning through our addresses though. We have only started monitoring recently but typically we see only one source and one destination. Without proper filtering / firewalling, this activity can be dangerous because by default MS machines are configured connectivity not security.

Technique / Intent: Source is scanning target addresses incrementing by one with each packet. Relatively slow speed indicates that this could be a manual scan rather than an automated application. No attempt has been made to avoid detection (i.e. low and slow). Their intent is to connect to our systems on port 137 and gather information.

Analysis: Since the source port is 137 it is probably NBTSTAT. This tool can be used to gather sensitive information about NT and Windows systems. I don't get to worked up over NETBIOS since we block it at the firewall and it's so prevalent that I'd get nothing else done if I spent too much time on it. I probably would not e-mail the network administration of the source about this because I'm not absolutely convinced that this is malicious.

Severity: (Critical + Lethal) – Countermeasures = -3

Criticality =	**2** This crosses some critical addresses
Lethality =	**3** NETBIOS can be a great tool for recon because it can gather sensitive information on NT/Windows machines.
System Countermeasures =	**3** Modern patched OS but I can't give it a 4 because it's MS (*I've been to John Greens class*)

Network Countermeasures = **5** Router ACL denies packets before they even reach firewall

Network Detect 2 Outlook Backdoor

```
*Mar 25 15:51:27.203: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1091) -> xxx.xxx.xxx.84(4459), 1 packet
*Mar 25 15:53:07.002: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1048) -> xxx.xxx.xxx.84(4452), 1 packet
*Mar 25 15:53:32.364: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1096) -> xxx.xxx.xxx.84(135), 1 packet
*Mar 25 15:56:15.091: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1057) -> xxx.xxx.xxx.84(4459), 5 packets
*Mar 25 15:57:15.145: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1091) -> xxx.xxx.xxx.84(4459), 3 packets
*Mar 25 15:58:15.203: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1048) -> xxx.xxx.xxx.84(4452), 5 packets
*Mar 25 15:59:15.262: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1096) -> xxx.xxx.xxx.84(135), 3 packets
*Mar 25 16:03:49.592: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1069) -> xxx.xxx.xxx.84(4459), 1 packet
*Mar 25 16:04:40.215: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1118) -> xxx.xxx.xxx.84(4459), 1 packet
*Mar 25 16:04:49.510: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1119) -> xxx.xxx.xxx.84(135), 1 packet
*Mar 25 16:05:25.625: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1062) -> xxx.xxx.xxx.84(4452), 1 packet
*Mar 25 16:09:15.802: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1069) -> xxx.xxx.xxx.84(4459), 5 packets
*Mar 25 16:10:15.857: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1119) -> xxx.xxx.xxx.84(135), 3 packets
*Mar 25 16:11:15.915: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.12(1062) -> xxx.xxx.xxx.84(4452), 5 packets

*Mar 25 17:05:33.099: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.9(1045) -> xxx.xxx.xxx.84(135), 1 packet
*Mar 25 17:07:34.930: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.9(1052) -> xxx.xxx.xxx.84(135), 1 packet
*Mar 25 17:11:19.182: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.9(1045) -> xxx.xxx.xxx.84(135), 3 packets
*Mar 25 17:11:55.466: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.9(1059) -> xxx.xxx.xxx.84(135), 1 packet
```

Active Targeting: Yes, this was targeted at a specific device on my network.

History: This was the first time I had seen anything like this come up in our log files. I am unaware of any Trojans using these ports.

Technique / Intent: Source address scans the target address (our mail server) on three specific ports. Packets arrive constantly for approximately 20 minutes and then stop. Approximately 55 minutes later a different address on the same subnet hits us again on the same ports to the same server. Source port numbers have a consistency to them also. Intent was to connect to our Exchange server. My initial assumption was that this was malicious.

Analysis: Initially this capture appeared to display the anomalous features of a port scan. The source port repeats itself and had a definite target of my mail server, after performing an nslookup, it turned out to be a dial up account to our very own ISP. Then the IP address changed which made me believe that the person was attempting to elude detection. These connections were denied due to a TCP established only statement with a log on deny all at the end of the list. As it turned out, our ISP told us that the person was an employee of our company. I tried her extension and in fact she was at work. She was trying to use a dial up at work while connected to the LAN so that she could use an online schooling program. She then tried to connect to MS Outlook, which traveled via dialup our Internet router and was logged. Therefore I was able to catch a backdoor to our network via a dialup account that was supposed to have gone away years ago when we got our T1.

Severity: (Critical + Lethal) – Countermeasures = -1

Criticality =	**4** Target was e-mail server
Lethality =	**3** While the detect at the router ACL was benign, an open back door to my network could have led to bad things
System Countermeasures =	**3** Modern patched OS but I can't give it a 4 because it's MS
Network Countermeasures =	**5** Router ACL denies packets before they even reach firewall

Network Detect 3 IRC Denials from Multiple Sources

```
Apr  4 07:31:35.611: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.23(6667) -> xxx.xxx.xxx.110(1117), 1 packet
Apr  4 07:33:14.698: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.10(6667) -> xxx.xxx.xxx.110(1156), 1 packet
Apr  4 07:35:36.037: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.20(6667) -> xxx.xxx.xxx.110(1209), 1 packet
Apr  4 07:36:27.125: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.130(6667) -> xxx.xxx.xxx.110(1220), 1 packet
Apr  4 07:37:09.174: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.130(6667) -> xxx.xxx.xxx.110(1241), 1 packet
Apr  4 07:37:20.521: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.23(6667) -> xxx.xxx.xxx.110(1117), 3 packets
Apr  4 07:38:20.575: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.10(6667) -> xxx.xxx.xxx.110(1156), 1 packet
Apr  4 07:38:35.920: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.23(6667) -> xxx.xxx.xxx.110(1271), 1 packet
Apr  4 07:40:08.929: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.10(6667) -> xxx.xxx.xxx.110(1301), 1 packet
Apr  4 07:41:20.730: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.20(6667) -> xxx.xxx.xxx.110(1209), 5 packets
Apr  4 07:42:20.785: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.130(6667) -> xxx.xxx.xxx.110(1220), 5 packets
Apr  4 07:42:36.158: %SEC-6-IPACCESSLOGP: list 101 denied tcp xxx.xxx.xxx.20(6667) -> xxx.xxx.xxx.110(1351), 1 packet
```

Active Targeting: Yes, this was targeted at a specific device on my network.

History: This was the first time I had seen this source port come up in our denied log files.

Technique / Intent: Multiple sources addresses attempt to connect to our router global address from port 6667 beginning at 7:30 AM and continuing throughout the day. This pattern continued throughout the week. I was unsure of the intent but the abruptness of its beginning and the unceasing flow of packets concerned me enough to investigate further.

Analysis: This worried me at first, mainly because I had not seen this port 6667 on my network before and I had many hundreds of entries on this day. It seemed to me in the back of my mind that some Trojan used a port like 666 (*I was thinking of Satanz Backdoor 666*). As it turned out of course it is IRC (Internet Relay Chat). This was denied because of my permit TCP established statement on my packet filtering router. That means that these TCP sessions were instigated (SYN'd) from a remote location and when looking at the source addresses it was coming at me from several different locations. My threat level flag was high, until I researched further. I trace routed these addresses and found that several sites with IRC in the name. In retrospect the time frame tells me that someone who starts work around 7:30 was probably recently "turned on" to chat by a friend last night. While I realize that this probably not an attack further research indicates that IRC is dangerous to run and should be blocked. I will also block it outbound in the future.

Severity: (Critical + Lethal) – Countermeasures = -2

Criticality =	**3** Target was global address
Lethality =	**3** IRC can be a dangerous thing to run on a network (insecure)
System Countermeasures =	**3** Modern patched OS (probably WIN98) as usual I can’t give it a 4 because it’s MS
Network Countermeasures =	**5** Router ACL denies packets before they even reach firewall

Network Detect 4 Port Scan

```
Apr 14 16:03:23 [xxx.xxx.xxx.1] Apr 14 2000 16:02:33: %PIX-2-106006: Deny inbound UDP from 207.71.92.193/137 to xxx.xxx.xxx.140/137
Apr 14 16:03:25 [xxx.xxx.xxx.1] Apr 14 2000 16:02:34: %PIX-2-106006: Deny inbound UDP from 207.71.92.193/137 to xxx.xxx.xxx.140/137
Apr 14 16:03:26 [xxx.xxx.xxx.1] Apr 14 2000 16:02:36: %PIX-2-106006: Deny inbound UDP from 207.71.92.193/137 to xxx.xxx.xxx.140/137

Apr 14 16:03:32 [xxx.xxx.xxx.1] Apr 14 2000 16:02:41: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2356 to xxx.xxx.xxx.140/21 flags SYN
Apr 14 16:03:32 [xxx.xxx.xxx.1] Apr 14 2000 16:02:42: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2356 to xxx.xxx.xxx.140/21 flags SYN
Apr 14 16:03:33 [xxx.xxx.xxx.1] Apr 14 2000 16:02:43: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2356 to xxx.xxx.xxx.140/21 flags SYN
Apr 14 16:03:34 [xxx.xxx.xxx.1] Apr 14 2000 16:02:43: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2356 to xxx.xxx.xxx.140/21 flags SYN

Apr 14 16:03:34 [xxx.xxx.xxx.1] Apr 14 2000 16:02:43: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2364 to xxx.xxx.xxx.140/23 flags SYN
Apr 14 16:03:34 [xxx.xxx.xxx.1] Apr 14 2000 16:02:44: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2364 to xxx.xxx.xxx.140/23 flags SYN
Apr 14 16:03:35 [xxx.xxx.xxx.1] Apr 14 2000 16:02:44: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2364 to xxx.xxx.xxx.140/23 flags SYN
Apr 14 16:03:35 [xxx.xxx.xxx.1] Apr 14 2000 16:02:45: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2364 to xxx.xxx.xxx.140/23 flags SYN

Apr 14 16:03:36 [xxx.xxx.xxx.1] Apr 14 2000 16:02:45: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2373 to xxx.xxx.xxx.140/25 flags SYN
Apr 14 16:03:36 [xxx.xxx.xxx.1] Apr 14 2000 16:02:46: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2373 to xxx.xxx.xxx.140/25 flags SYN
Apr 14 16:03:37 [xxx.xxx.xxx.1] Apr 14 2000 16:02:46: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2373 to xxx.xxx.xxx.140/25 flags SYN
Apr 14 16:03:37 [xxx.xxx.xxx.1] Apr 14 2000 16:02:47: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2373 to xxx.xxx.xxx.140/25 flags SYN

Apr 14 16:03:38 [xxx.xxx.xxx.1] Apr 14 2000 16:02:47: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2383 to xxx.xxx.xxx.140/79 flags SYN
Apr 14 16:03:38 [xxx.xxx.xxx.1] Apr 14 2000 16:02:48: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2383 to xxx.xxx.xxx.140/79 flags SYN
Apr 14 16:03:39 [xxx.xxx.xxx.1] Apr 14 2000 16:02:48: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2383 to xxx.xxx.xxx.140/79 flags SYN

Apr 14 16:04:03 [xxx.xxx.xxx.1] Apr 14 2000 16:03:13: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2506 to xxx.xxx.xxx.140/80 flags SYN
Apr 14 16:04:04 [xxx.xxx.xxx.1] Apr 14 2000 16:03:13: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2506 to xxx.xxx.xxx.140/80 flags SYN
Apr 14 16:04:05 [xxx.xxx.xxx.1] Apr 14 2000 16:03:14: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2506 to xxx.xxx.xxx.140/80 flags SYN
Apr 14 16:04:05 [xxx.xxx.xxx.1] Apr 14 2000 16:03:15: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2506 to xxx.xxx.xxx.140/80 flags SYN

Apr 14 16:04:05 [xxx.xxx.xxx.1] Apr 14 2000 16:03:15: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2514 to xxx.xxx.xxx.140/110 flags SYN
Apr 14 16:04:06 [xxx.xxx.xxx.1] Apr 14 2000 16:03:15: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2514 to xxx.xxx.xxx.140/110 flags SYN
Apr 14 16:04:06 [xxx.xxx.xxx.1] Apr 14 2000 16:03:16: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2514 to xxx.xxx.xxx.140/110 flags SYN
Apr 14 16:04:07 [xxx.xxx.xxx.1] Apr 14 2000 16:03:17: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2514 to xxx.xxx.xxx.140/110 flags SYN

Apr 14 16:04:07 [xxx.xxx.xxx.1] Apr 14 2000 16:03:17: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2527 to xxx.xxx.xxx.140/113 flags SYN
Apr 14 16:04:08 [xxx.xxx.xxx.1] Apr 14 2000 16:03:17: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2527 to xxx.xxx.xxx.140/113 flags SYN
Apr 14 16:04:08 [xxx.xxx.xxx.1] Apr 14 2000 16:03:18: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2527 to xxx.xxx.xxx.140/113 flags SYN
Apr 14 16:04:09 [xxx.xxx.xxx.1] Apr 14 2000 16:03:18: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2527 to xxx.xxx.xxx.140/113 flags SYN
Apr 14 16:04:09 [xxx.xxx.xxx.1] Apr 14 2000 16:03:19: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2540 to xxx.xxx.xxx.140/139 flags SYN
Apr 14 16:04:10 [xxx.xxx.xxx.1] Apr 14 2000 16:03:19: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2540 to xxx.xxx.xxx.140/139 flags SYN
Apr 14 16:04:10 [xxx.xxx.xxx.1] Apr 14 2000 16:03:20: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2540 to xxx.xxx.xxx.140/139 flags SYN
Apr 14 16:04:11 [xxx.xxx.xxx.1] Apr 14 2000 16:03:20: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2540 to xxx.xxx.xxx.140/139 flags SYN

Apr 14 16:04:11 [xxx.xxx.xxx.1] Apr 14 2000 16:03:21: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2548 to xxx.xxx.xxx.140/143 flags SYN
Apr 14 16:04:12 [xxx.xxx.xxx.1] Apr 14 2000 16:03:21: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2548 to xxx.xxx.xxx.140/143 flags SYN
Apr 14 16:04:12 [xxx.xxx.xxx.1] Apr 14 2000 16:03:22: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2548 to xxx.xxx.xxx.140/143 flags SYN
Apr 14 16:04:13 [xxx.xxx.xxx.1] Apr 14 2000 16:03:22: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2548 to xxx.xxx.xxx.140/143 flags SYN

Apr 14 16:04:13 [xxx.xxx.xxx.1] Apr 14 2000 16:03:22: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2554 to xxx.xxx.xxx.140/443 flags SYN
Apr 14 16:04:13 [xxx.xxx.xxx.1] Apr 14 2000 16:03:23: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2554 to xxx.xxx.xxx.140/443 flags SYN
Apr 14 16:04:14 [xxx.xxx.xxx.1] Apr 14 2000 16:03:24: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2554 to xxx.xxx.xxx.140/443 flags SYN
Apr 14 16:04:15 [xxx.xxx.xxx.1] Apr 14 2000 16:03:24: %PIX-2-106001: Inbound TCP connection denied from 207.71.92.221/2554 to xxx.xxx.xxx.140/443 flags SYN
```

Active Targeting: Yes, this was targeted at a specific device on my network.

History: In the short time that we have been monitoring our network we have seen this one twice. (*Less interesting the second time*)

Technique / Intent: Single source address attempts to connect to our firewall global address on several "well known" ports in rapid succession with no attempt to elude detection. This is an automated scan looking for an open port. Intent would appear to be an all out attempt to exploit poor network security. However, this turned out to be a requested port scan by a user on my network so there is no malicious intent. See Analysis for details.

Analysis: As usual with the few detects from my network, this too turned out to be fairly benign. While this is truly a port scan for vulnerability, an nslookup reveals the attacker to be www.grc.com. This is a resource site aimed at educating its visitors on securing their machines. As a service it offers the ability to have your ports scanned. When you first connect to the site it attempts the NETBIOS connect with 3 packets without your knowledge. If you request further scanning it runs through FTP, TELNET, SMTP etc. I imagine that if I've seen this twice already, it will become a popular detect in the future. While I'm not crazy about inviting people to scan our network, I think the threat from this site is relatively low.

Severity: (Critical + Lethal) – Countermeasures = <u>-1</u>

Criticality =	**3** Target was firewall global address
Lethality =	**4** This could be an intelligence gathering prelude to an attack of a more serious sort
System Countermeasures =	**3** Modern patched OS (probably WIN98) as usual I can't give it a 4 because it's MS
Network Countermeasures =	**5** Router ACL denies packets before they even reach firewall

Network Detect 5 (From the SANS site 4-15-00) Syn / Fin Scan to POP-2 Port

(Detects from my network were somewhat vanilla, let's see something interesting from the GIAC page)

```
01:00:43.335934 xxx.yyy.103.1.109 > aaa.bbb.122.172.109:
SF 214832337:214832337(0) win 1028
01:00:43.355946 xxx.yyy.103.1.109 > aaa.bbb.122.173.109:
SF 214832337:214832337(0) win 1028
01:00:43.376123 xxx.yyy.103.1.109 > aaa.bbb.122.174.109:
SF 214832337:214832337(0) win 1028
```

Active Targeting: Yes, this was targeted at a specific network, incrementing the node number in a scanning fashion.

History: This is a known scan type that used to use fixed ports 0 or 65535. This scan is probably a newer variant of the tool.

Technique / Intent: Crafted packets with a non-logical combination of flags set may be permitted through some filtering devices because the FIN flag indicates that a session already exists. Several exploits utilize this signature including Jackal and NMAP. Note that the sequence numbers are not incrementing, also indicating a crafted packet. While I don't know the intent of hitting the POP-2 port, I can be reasonably certain that the crafted packet nature of this scan indicates a malicious action.

Analysis: A SYN/FIN scan utilizing POP-2 scanning the network rapidly. There is 0 data in the payload so I would probably think this is an attempt at recon and not an attempt to push something through the door. I would watch the source address and probably try to contact the administrators since this is probably coming from a compromised system.

Severity: (Critical + Lethal) – Countermeasures = 1

Criticality = **3** We can assume that the target is at least a 3 if we gave it a valid (non-NAT) Internet address
Lethality = **4** I will rate this at a 4 because I'm not aware and therefore cautious of a POP2 exploit
System Countermeasures = **3** Assuming a Modern patched OS (probably WIN98 or NT) as usual I can't give it a 4 because it could be MS
Network Countermeasures = **3** Although a statefull firewall should halt this, a packet filtering router might not due to its established rule

Network Detect 6 (From the SANS site 4-14-00) Ring 0 Scan

```
Apr 11 14:28:20.461205 158.135.8.129,3545 ->
10.0.3.133,80 PR tcp len 20 44 -S
Apr 11 14:28:26.398249 158.135.8.129,3545 ->
10.0.3.133,80 PR tcp len 20 44 -S
Apr 11 14:28:38.421751 158.135.8.129,3545 ->
10.0.3.133,80 PR tcp len 20 44 -S
Apr 11 14:29:02.468325 158.135.8.129,3595 ->
10.0.3.133,8080 PR tcp len 20 44 -S
Apr 11 14:29:05.482538 158.135.8.129,3595 ->
10.0.3.133,8080 PR tcp len 20 44 -S
Apr 11 14:29:11.400713 158.135.8.129,3595 ->
10.0.3.133,8080 PR tcp len 20 44 -S
Apr 11 14:29:23.436332 158.135.8.129,3595 ->
10.0.3.133,8080 PR tcp len 20 44 -S
Apr 11 14:29:47.746981 158.135.8.129,3639 ->
10.0.3.133,3128 PR tcp len 20 44 -S
Apr 11 14:29:50.752318 158.135.8.129,3639 ->
10.0.3.133,3128 PR tcp len 20 44 -S
Apr 11 14:29:56.817173 158.135.8.129,3639 ->
10.0.3.133,3128 PR tcp len 20 44 -S
Apr 11 14:30:08.801161 158.135.8.129,3639 ->
10.0.3.133,3128 PR tcp len 20 44 -S
```

Active Targeting: Yes, This scan is looking for specific ports on a specific address.

History: This is a known scan type that seeks an exploitable Trojan known as Ring0. This Trojan has been around since September 1999. and affects Windows 95/98 machines.

Technique / Intent: Ring0is a modern complex Trojan which is thought to infect machines through e-mail attachments. Once infected the host machine will scan random IP addresses for ports 80 8080 and 3128. If a connect occurs, it will send its own IP address and proxy port off. The intent is for the compromised host to deliver its IP address and proxy port number to the recipient.

Analysis: This is a classic example of the Ring0 Trojan scanning the ports 80 8080 and 3128 looking for a connect. Probably not cause for alarm if it's coming to your network but I would not want to see it leaving my network. Nor would I like to see a response from my network to the request.

Severity: (Critical + Lethal) – Countermeasures = 2

Criticality =	**3** We can assume that the target is at least a 3 if we gave it a valid Internet address
Lethality =	**3** I would say 5 if it was coming from my network and 3 if it was coming to my network. I'll compromise on 3
System Countermeasures =	**2** If this is coming from my network then my system countermeasures have failed.
Network Countermeasures =	**2** Incoming this should be stopped but not outgoing

Network Detect 7 (From the SANS site 4-13-00) Zone Transfer Scan

```
Apr 11 05:32:59 24.27.209.180:1482 -> a.b.c.19:53 SYN **S*****
Apr 11 05:32:59 24.27.209.180:1489 -> a.b.c.26:53 SYN **S*****
Apr 11 05:32:59 24.27.209.180:1496 -> a.b.c.33:53 SYN **S*****
Apr 11 05:32:59 24.27.209.180:1514 -> a.b.c.51:53 SYN **S*****
Apr 11 05:32:59 24.27.209.180:1525 -> a.b.c.62:53 SYN **S*****
Apr 11 05:33:02 24.27.209.180:1546 -> a.b.c.83:53 SYN **S*****
Apr 11 05:32:59 24.27.209.180:1684 -> a.b.c.221:53 SYN **S*****
Apr 11 05:32:59 24.27.209.180:1695 -> a.b.c.232:53 SYN **S*****
Apr 11 05:32:59 24.27.209.180:1698 -> a.b.c.235:53 SYN **S*****
```

Active Targeting: Yes, this was targeted at devices on "my" network so Northcutt's rule applies.

History: This is a scan for Zone Transfer. It can be used to download a host table from a poorly configured UNIX server.

Technique / Intent: Attacker attempts to connect to TCP 53 with the intent of gathering intelligence.

Analysis: Most well know DNS servers are configured so as not to be vulnerable to this type of scan. However UNIX systems running named are at risk for revealing host tables to a TCP 53 connection. I would keep an eye on the source of these packets. Somebody wants to get information on my network, which should raise awareness even though this particular scan will not be successful. Possibly an e-mail is in order to the remote network administrator.

Severity: (Critical + Lethal) – Countermeasures = 0

Criticality =	**5** DNS servers are significant and should not be allowed to be compromised
Lethality =	**3** If countermeasures aren't in place host tables could be obtained
System Countermeasures =	**3** I would be cautious in assuming that all UNIX systems are patched and up to date
Network Countermeasures =	**5** (TCP), This isn't getting past a border router ACL let alone a statefull firewall

Network Detect 8 (From the SANS site 4-11-00) Trojan Scan (Netbus / Subseven)

```
Apr 8 01:37:56 cc1014244-a kernel: securityalert: tcp if=ef0 from
216.77.245.249:2606 to 24.3.21.199 on unserved port 1243
Apr 8 01:37:56 cc1014244-a kernel: securityalert: tcp if=ef0 from
216.77.245.249:2607 to 24.3.21.199 on unserved port 12345
Apr 8 01:37:56 cc1014244-a kernel: securityalert: tcp if=ef0 from
216.77.245.249:2608 to 24.3.21.199 on unserved port 20034
Apr 8 01:37:56 cc1014244-a kernel: securityalert: tcp if=ef0 from
216.77.245.249:2609 to 24.3.21.199 on unserved port 27374
```

Active Targeting: Yes, this was targeted at devices on “my” network so Northcutt’s rule applies.

History: This is a scan for various Trojans.

Technique / Intent: Attacker is running automated tool (note speed of scan and the even sequencing of source port) to probe with the intent to connect to known Trojan ports. If the attacker is successful in connecting to a Trojan port, they will then exploit the particular features of the Trojan they find.

Analysis: Because of the speed of the scan I would say that there is no attempt to avoid detection. Because of the even sequencing of the source port numbers I think that we are the only network being scanned by this host at this time. The attacker is searching for a connection on NETBUS or SUBSEVEN 2.1. Because he would be denied at an edge router ACL with the established command or by a firewall, I would simply make note of the source IP and keep an eye out for any further activity from them. A nslookup reveals that it is an ADSL user on the Bell South network. Probably a script kiddie, but we’ll keep an eye out. An e-mail to the Bell South network administrator is also in order.

Severity: (Critical + Lethal) – Countermeasures = 2

Criticality =	**5** If a connect were achieved to one of these ports it would indicate that Trojans are already on the network.
Lethality =	**5** Attack could succeed and grant privileges if no network countermeasures were in place and the Trojans were installed
System Countermeasures =	**3** Assuming a Modern patched OS (probably WIN98 or NT) as usual I can’t give it a 4 because it could be MS
Network Countermeasures =	**5** This isn’t getting past a border router ACL let alone a statefull firewall

Network Detect 9 (From the SANS site 2-29-00) Back Orifice Scan

```
[**] IDS188/probe-back-orifice [**]
02/13-03:52:42.176130 24.130.49.191:7430 -> 172.16.1.239:31337
UDP TTL:119 TOS:0x0 ID:16631
Len: 26

[**] IDS188/probe-back-orifice [**]
02/13-03:52:42.190341 24.130.49.191:7430 -> 172.16.1.240:31337
UDP TTL:119 TOS:0x0 ID:16887
Len: 26

[**] IDS188/probe-back-orifice [**]
02/13-03:52:42.215992 24.130.49.191:7430 -> 172.16.1.242:31337
UDP TTL:119 TOS:0x0 ID:17399
Len: 26
```

Active Targeting: Yes, this was targeted at devices on “my” network so Northcutt’s rule applies.

History: This is a scan for Back Orifice. This is one of the most common Trojans and was created by the Cult of the Dead Cow hacker group. Once installed on to a machine running Windows 95/98, a remote machine can connect and assume high level functions on the target machine.

Technique / Intent: Attacker is running automated tool (note speed of scan) to probe in an attempt to connect to port 31337. The intent is to find, connect to and exploit the features of a back orifice Trojan.

Analysis: Because of the speed of the scan I would say that there is no attempt to avoid detection. The attacker is searching for a connection on port 31337, which is the default back orifice port. Because they would be denied at the firewall, I would simply make note of the source IP and keep an eye out for any further activity from them. A nslookup reveals that it is coming from Media One.net, most likely a dial-up user. Probably a script kiddie, but again we’ll keep an eye out. An e-mail to the Media One network administrator is also in order.

Severity: (Critical + Lethal) – Countermeasures = 3

Criticality =	**5** If a connect were achieved to one of these ports it would indicate that Trojans are already on the network.
Lethality =	**5** Attack could succeed and grant monitoring and control if no network countermeasures were in place and the Trojan were installed
System Countermeasures =	**3** Assuming a Modern patched OS (probably WIN98 or NT) as usual I can’t give it a 4 because it could be MS
Network Countermeasures =	**4** This isn’t getting past a statefull firewall but might get past a poorly set up router ACL

Network Detect 10 (From the SANS site 2-24-00) SNMP Scan

```
Feb 24 10:59:42 host2 snort:
SNMP access, public: 212.60.50.151:3696 -> x.x.x.80:161
Feb 24 10:59:45 host2 snort:
SNMP access, public: 212.60.50.151:3696 -> x.x.x.83:161
Feb 24 11:00:06 host2 snort:
SNMP access, public: 212.60.50.151:3696 -> x.x.x.101:161
Snort:
[**] SNMP access, public [**]
02/24-10:57:56.119759 212.60.50.151:3696 -> x.x.x.14:161
UDP TTL:102 TOS:0x0 ID:59829
Len: 62
30 34 02 01 00 04 06 70 75 62 6C 69 63 A1 27 02 04.....public.'.
03 00 A6 D1 02 01 00 02 01 00 30 1A 30 0B 06 07 ..........0.0...
2B 06 01 02 01 01 01 05 00 30 0B 06 07 2B 06 01 +........0...+..
02 01 01 02 05 00 ......
```

Active Targeting: Yes, this was targeted at devices on “my” network so Northcutt’s rule applies.

History: SNMP or Simple Network Management Protocol can be used for network mapping and also for remote management of network equipment such as switches hub and routers.

Technique / Intent: Attacker is attempting to connect to port 161, (SNMP) with the default community string of public. The intent is to gather information on “our” network.

Analysis: This would raise an eyebrow for me. The source is looking to connect to “my” SNMP enabled devices which could be switches or routers etc. If by default I was to have SNMP enabled on equipment the community string of public would probably be in place. This is especially true on new equipment that might have been placed for growth but not configured for security yet. SNMP is not as glamorous as Back Orifice so I wouldn’t think this is a novice hacker. I was unable to obtain an nslookup on the address so I think I would watch carefully for activity from this address as well as to look for other hosts as well. The attacker may be advanced enough to have several proxy locations from which to work.

Severity: (Critical + Lethal) – Countermeasures = 3

Criticality =	**4** If a connect were achieved valuable network information could be obtained.
Lethality =	**5** Attack could succeed and grant monitoring and control if no network countermeasures were in place and SNMP defaults were in place
System Countermeasures =	**2** Defaults are often left enabled on layer 2 devices
Network Countermeasures =	**4** This isn’t getting past a statefull firewall or a properly configured router AC. I give it a 4 though because this source concerns me.