Global Information Assurance Certification Paper

GCIA Intrusion Detection in Depth
Scott Higgins
SAN Course On-Line
Practical Assignment Version 3.2
(revised May 2002)

Assignment 1- Describe the State of Intrusion Detection

The Challenge of Intrusion Detection on a Secure/Closed Network

Overview

Intrusion detection on an open network environment, such as the Internet, is a complex and ever changing discipline, which requires diligence and constant watch. Always having to be vigilant and expecting the unexpected is something you come to live with during a career as an intrusion detection analyst. The tools an intrusion detection analyst uses to perform their duties on an open network are many. Several of these tools are services provided by other web sites or organizations, in addition to shareware type applications the analyst may be partial to. For the most part, there are typically a steadfast few that are readily available for quick diagnosis via hyperlink. What about those networks that are not connected to the Internet, the so-called secure or closed networks which some companies and the government employ? For the purposes of this paper I will focus on conducting IA on the government's secure/closed network, which is not connected to the Internet. For example, the government employs a secure Wide Area Network (WAN) made up of many different organizations and networks spread all over the world. Some of these networks include deployed sites in the middle of foreign places managed by inexperienced Information Assurance security personnel. Although the networks conform to a standard

security baseline, there are many that lack the basic configuration niceties, such as current or existent DNS resolution information. Do these networks deserve less scrutiny or vigilance because the users are supposedly more trusted than the average bit jockey riding the Internet? Do they deserve less scrutiny because the network is secure or closed? No, they deserve just as much scrutiny, if not more attention, because it is a secure network.

Know Thy Self – The true path to intrusion detection enlightenment

There are several great truths to intrusion detection. Truth number one: in order to effectively audit a network for security vulnerabilities; one must be familiar with how security can be compromised. In other words, know thy enemy. Another of the great truths of intrusion detection that is drilled into our heads: become familiar with the network you are watching over; or know thy self. Become intimately familiar with the type of traffic you will be seeing on your network, the frequency, type of connections, and protocols. It is imperative to have a solid understanding of these things in order to determine an information security traffic baseline. Establishing a traffic pattern baseline for a secure/closed network is very challenging even for the most accomplished intrusion detection analyst. Knowing these different but critical pieces of information, in addition to using our information analyst's toolbox, makes detecting anomalous traffic more of a challenge than a headache. Most government secure/closed networks suffer from a lack o,f what I call, configuration common sense. Because of security, wide range of locations of organizations, and assumed responsibility of the trusted individuals and environments that often times surround secure/closed networks; the sharing of configuration information is never achieved. In addition, many things that are typically restricted for security reasons on corporate networks, such as IRC, are done on a daily basis on the secure/closed network. Network and system information is so tightly held, that at times, DNS resolutions on a secure/closed network return no data because the site did not want to provide any data for security reasons. This lack of basic information makes an analyst's job more difficult by taking one of their basic tools away and not being able to discover who to contact about a specific network. To resolve this problem, a configuration control board (CCB) should develop, support, and enforce a mutually acceptable amount of DNS data points that each organization should have to provide in order to facilitate the IA process. Currently, many sections and organizations must be queried on a continual basis in order to stay abreast of current changes. Sun Tzu said many years ago that if you know your enemy, you have won half the battle; if you also know yourself, then you have won the battle. This holds true in the intrusion detection realm today and should become a guiding principal for each intrusion analyst.

Internet Tools – Like the comfort of a best friend

When a forensic scientist performs their job, they must posses the correct set of tools in order to complete the investigation as correctly and efficiently as possible. Without these tools, they may be able to get the job done, but it will take longer and not be done to the standards set forth in the law. The same holds true for an intrusion detection analyst. When an intrusion detection analyst is working on an open network, such as the Internet, they have a variety of tools at their disposal, via the web, in addition to their experience. Some examples of the tools at their disposal: ARIN.net, geektools.com, Snort.org, SANS.org, and dogpile.com. These tools provide the analyst with the ability to quickly and efficiently retrieve and correlate data that is used for triage during the initial analysis of network traffic. What happens when the analyst cannot execute a basic ARIN.net lookup and get an informative response? The IA analyst would be unable to determine a POC for the network they are concerned with and thus unable to provide any assistance to the site. This could effectively prevent a timely IA response to an incident or intrusion, lose evidence, or hamper the organization IA mission. This could be easily corrected by implementing and enforcing an ARIN.net type function on the secure network. What if the intrusion detection analyst's tools were not available to them? What happens when an analyst can not conduct a thorough analysis on network traffic? An analyst's ability to conduct in depth analysis would be severely hampered and would only allow a high level analysis of network traffic; for example, number of connections or port connections, which is not as effective as having the appropriate tools available. Most of the time the analysts that watch over a secure/closed network have access to a set of tools that are comparable in intention, but greatly reduced in capability and functionality. An example of this would be ARIN.net on the Internet. Arin.net returns valuable information on points of contact and other pertinent information on specific IP addresses and IP subnets. This information typically consists of a name, email and/or physical address, and phone number of the responsible party that oversees the IP or subnet in question (figure 1).

Yahoo! (NETBLK-A-YAHOO-U23)
701 First Avenue
Sunnyvale, California 94089
US

Netname: A-YAHOO-U23
Netblock: 66.218.64.0 - 66.218.95.255

Maintainer: YAOO

Coordinator:
Admin, Netblock (NA258-ARIN) netblockadmin@yahoo-inc.com
1-408-349-7183

Domain System inverse mapping provided by:

NS1.YAHOO.COM	66.218.71.63
NS2.YAHOO.COM	209.132.1.28
NS3.YAHOO.COM	217.12.4.104
NS4.YAHOO.COM	63.250.206.138
NS5.YAHOO.COM	64.58.77.85

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Record last updated on 27-Jun-2002.
Database last updated on 9-Aug-2002 20:01:47 EDT.

Figure 1. Typical ARIN.net information

As mentioned above, some secure/closed networks do not have the luxury of the Internet ARIN.net function or other tools located on the Internet. These tools have to be recreated as best as possible. Sometimes a typical response from a secure/closed ARIN.net DNS type function consists of nothing at all, because there is no mechanism in place to provide this information. IA signature analysis is severely hampered since all the data is on the Internet and not available to the secure network. To assist in solving this problem, Internet security web sites are sometimes ported over to the secure/closed network; for instance, Snort.org and Whitehats.com, but they only contain a portion of the site and the updated information required by analysts. To solve this problem, a CCB should agree on a set of IA tools to be ported over or developed, such as Snort signature data from Whitehats.com, to be used on the secure network to facilitate the IA process. These issues have been around as long as the Internet and secure/closed networks. Through intensive lobbying by intrusion detection analysts, information assurance professionals, and recent events, the gap between the amount and type of tools available on open and secure/closed networks is closing.

Programs, Protocols, and Practices (PPP) – Not the PPP we grew up on

Working on an open network, an analyst sees many different types of traffic. Today, a majority of the traffic crossing both the open and secure/closed networks are web and email directed with a little DNS, Telnet, FTP, and various others peppered in. One might surmise that intrusion detection analysts working on secure/closed networks would not see the same traffic that an open network intrusion detection analyst would see on the Internet. On the contrary, some of the secure/closed network traffic might scare an open network intrusion detection analyst to death. Open networks typically see encryption used to conduct a majority of sensitive transactions. On secure/closed networks this is rarely the case. System administration is often conducted in the clear on critical systems. A couple examples would be administrators tweaking their web site using an unencrypted transfer protocol, or X-Windows administration of a DNS server that is also unencrypted. Internet Relay Chat is also used as one means of communications with some of the more widely dispersed road warriors. If these things were to take place on the Internet, or any other open network, it would not be long before the intrusion detection analysts would be actually detecting an intrusion. The reasons for this stance have been mentioned before; the users are considered to be more trustworthy and not malicious due to increased screening, and the network is closed so what could possibly happen on here. To combat these problems on the secure network, the same security mentality and best business practices should prevail, regardless of how much trust we put in our users. None of the precautions taken by any person or organization can reveal actions a person is willing to commit. A quick review of the Computer Security Institute/FBI Computer Crime and Security Survey will show you that insiders are the most dangerous.

Conclusion

Intrusion detection is a process that must be executed by knowledgeable and competent intrusion detection analysts in order to maintain secure networks. They must have the appropriate tools and training to accomplish their duties in a timely manner. A secure/closed network will challenge an analyst's abilities due to the different standards that govern those types of networks. An intrusion detection analyst must understand the importance of knowing their own network and the types of traffic they will be seeing. This is particularly difficult in secure/closed environments. An intrusion detection analyst will be able to win the whole information assurance battle by following what Sun Tzu initially described in the Art of War, if you know thy self and you know thy enemy, you have won the battle,.

References:

Computer Security Institute. “2002 Computer Crime and Security Survey.” (April 2002)
URL: http://www.gocsi.com/press/20020407.html
12 August 2002

Giles, Lionel. “SUN TSZU on the Art of War.” (19 November 2002)
URL: http://www.chinapage.com/sunzi-e.html
12 August 2002

Thomas, Benjamin. “Intrusion Detection Primer” (13 March 2000)
URL: http://www.linuxsecurity.com/feature_stories/feature_story-8.html
12 August 2002

Northcutt, Stephen. Network Intrusion Detection: An Analyst's Handbook. Indianapolis: New Riders, 2001
Schatz, Michael. “Learning Program Behavior profiles for Intrusion Detection”. (February 1999)
URL:http://www.usenix.org/publications/library/proceedings/detection99/full_papers/ghosh/ghosh_html/ 12 August 2002

Assignment 2 – Network Detects

The following alert listings and detects were generated by a Network Intrusion Detection System that uses SNORT in addition to other NIDS tools to detect anomalous activity in IP data flows. The Network Intrusion Detection System captures network data at the perimeter firewall. The alert listings and packet decodes shown are formatted by the Analysis Console for Intrusion Databases (ACID).

Alert listings have the following format:

Alert ID	Signature	Time Stamp	Source Address: Source Port	Dest. Address: Dest Port	Layer 4 Proto

The Packet decodes are divided into four sections:

Meta data: Items included in this section are the Alert ID, Time Stamp, Signature matched, Sensor that detected the packet, Sensor interface, and Alert Group.

IP header data: All fields in the header are labeled. An nslookup is performed

for IP and provided in the section titled “Fully Qualified Domain Name (FQDN).” Any IP options set are displayed in the last field in this section.

Alert ID	Signature	Time Stamp	Source Address	Dest. Address	Layer 4 Proto
#0 - NID-xyz	MIT cookie	2002-08-09 12:12:43	host.my.net.211:1741	Host.my.net.172:6000	TCP
#1 - NID-xyz	MIT cookie	2002-08-09 12:28:54	Host.my.net211:4330	Host.my.net172: 6000	TCP
#2 - NID-xyz	MIT cookie	2002-08-09 13:42:59	Host.my.net211:1783	Host.my.net.172: 6000	TCP
#3 - NID-xyz	MIT cookie	2002-08-09 13:42:59	Host.my.net211:1784	host.my.net.172: 6000	TCP
#4 - NID-xyz	MIT cookie	2002-08-09 13:42:59	host.my.net211: 1785	host.my.net.172: 6000	TCP
#5 - NID-xyz	MIT cookie	2002-08-09 13:42:59	host.my.net211:1786	host.my.net.172: 6000	TCP
#6 - NID-xyz	MIT cookie	2002-08-09 13:42:59	host.my.net211:1787	host.my.net.172: 6000	TCP

Layer 4 protocol data: All fields for the associated TCP, UDP or ICMP header are labeled in this section. Any TCP options set are also displayed in this section.

Payload: This section displays the packets data gram. The data gram is displayed in hexadecimal format with an ASCII conversion for the reader’s convenience.

Network detect (1):

Alert # 6 – Packet Decode:

Meta	ID # Time Triggered Signature 3-360963 2002-08-09 13:42:59 X11 MITcookie

	Sensor Name Interface Filter	NID-xyz hme0 *None*
	Alert Group	*None*
IP	Source addr Dest addr Ver Hdr Len TOS Length ID Flags Offset TTL Chksum	host.my.net.211 host.my.net.172 4 5 88 38941 62 51495
	FQDN Source Name Dest. Name	host.my.net.211 host.my.net.172

	Options
	None

TCP	Source port Dest port R 1 R 0 U R G A C K P S H R S T S Y N F I N Seq # ACK Offset res Window urp Chksum 1787 6000 X X 970432001 4071939527 5 32768 46903

	Options None
Payload	000 : 42 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....MIT- 010 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 MAGIC-COOKIE-1.I 020 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 z9HYBvsrVqmo1G6

1. Source of trace:

The traces above were collected from a Department of Defense network. The destination Internet Protocol addresses as well as the hex code for the DoD network, have been sanitized in both the alert listings and the packet decodes. Note also that the name of the sensor that generated each alert has been obfuscated in the alert listings and packet decodes. Furthermore; the Fully Qualified Domain Name (FQDN) for all DoD source or destination hosts has been changed to “host.my.net.xxx”.

2. Detect was generated by:

This detect was generated by a Network Intrusion Detection System which uses SNORT in addition to other NIDS tools to detect anomalous activity in IP data flows. The SNORT output plug-in is configured to use the XML logging format. The XML data is logged by the sensor and retrieved at regular intervals by an analysis server. The analysis server uses custom PERL scripts to parse the data and log it to a MySQL database. The alert listings and packet decodes shown above were formatted by the Analysis Console for Intrusion Databases (ACID). “The Analysis Console for Intrusion Databases (ACID) is a PHP-based analysis engine to search and process a database of security events generated by various IDSs, firewalls, and network monitoring tools.” - http://www.cert.org/kb/acid/ . The ACID application is open source and as such freely available to all. Please visit the URL above to learn more about this tool.

Snort Signature:

alert TCP $EXTERNAL any -> $INTERNAL 6000 (msg: "IDS396/x11_X-MITcookie"; flags: A+; content: "MIT-MAGIC-COOKIE-1"; classtype: system-attempt; reference: arachnids,396;)

3. Probability the source address was spoofed:

None. This type of attack requires a response and is typically part of an established TCP session. The fact that the ACK bit is set, there is a sequence number, and an acknowledgement number indicate that a three way handshake has taken place. The above packet revealed a locally owned IP which was generating this traffic. It was determined that packet was neither crafted nor spoofed.

4. Description of the attack:

X11 authentication is used to authenticate a user's X windows session by using the MIT-MAGIC-COOKIE-1 keys generated by the xdm module.
Keys are generated from 16 successive random numbers. MIT-MAGIC-COOKIE-1 can use 2 different methods of seeding the random number generator:

a) Using the process ID of xdm client & time of day in seconds
b) Using the time of day in seconds & time of day in microseconds.

If the xdm module does not support XDM-AUTHORIZATION-1 authentication built with the HasXdmAuth config option and your keys are generated by xdm module your MIT-MAGIC-COOKIE-1 key is insecure and guessable. This is a directed attack against systems running X-Terminal with xdm and the MIT Magic Cookie key generator. This attack is currently under consideration to become a CVE.

CAN-1999-0241 (under review): http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0241

5. Attack mechanism:

This is a directed attack against systems running X-Terminal with xdm and the MIT Magic Cookie key generator. This attack works by guessing the key generated by the xdm Magic Cookie key module. This attack is similar to initial sequence number guessing. On systems with poor pseudo-random number generators, the key may be guessable by remote users. By correctly guessing the cookie, the malicious user can masquerade as the original client and gain access to the remote system and possible root access.

According to description "a", located in "Description of Attack"; in order to crack a user's cookie, you need to find the process id of the xdm handling a display

and know the time the session was started. One way to determine the time the session was created: locate the file that contains the server's copy of the authority data, stat the file, and then use the creation time (st_ctime) as the time component of the seed. Such files can be found in the authDir named in the xdm-config under DisplayManager.authDir.

Cracking cookies generated by method “b” located in “Description of Attack” above: the time of day is easily predicted by guesswork or by statting the server's authfile, but the time of day in microseconds has to be guessed. Matters are made slightly easier by the fact that the time of day in milliseconds is left shifted by 16 bits, and hence is only a 16-bit factor to deal with. If a user has access to the machine, it will take at most 2*65536 == 131072 iterations. Chances are slim that a user will stay logged on for a single session that long.

The vulnerability is this: the low 8 bits of numbers produced by successive calls to rand() repeat in the same sequence with a period of 256. Consequently, under certain Operating Systems, there are only 256 unique magic cookies that can ever be generated.

6. Correlations:

CVE CAN-1999-0241 (under review): http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0241

http://www.iss.net/security_center/static/334.php

7. Evidence of active targeting:

This is verified active targeting of a specific host address within the network. Only one host was targeted and the attack is typically part of an established TCP session. The fact that this packet has the ACK bit set as well as an acknowledgement and sequence number indicates that this is part of an established TCP session between the two hosts. It should be noted that legitimate use of MIT Magic Cookies will also trigger a false positive, but this traffic should be verified.

8. Severity:

The formula used to calculate the severity of an incident is as follows:

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures)

Each category is assigned a value on a scale of (1) to (5). Five is the most significant value that may be assigned to each category.

Criticality:	*5*
The host targeted by this attack was a DNS server.	
Lethality:	*5*
The attack is lethal and could result in root level access.	
System Countermeasures:	*5*
Patches for this vulnerability were installed.	
Network Countermeasures:	*4*
This network uses a stateful firewall with application proxies. The application proxies should detect the protocol anomaly and discard the packets. Using X11 to connect to a DNS server is a poor security practice, but the risk has been accepted by the organization.	
Overall Severity:	**1**

* The chart above builds upon the work of Mr. James Conz. CGIA practical: http://www.giac.org/practical/James_Conz_GCIA.doc

9. Defensive recommendation:

Ensure XDM-AUTHORIZATION-1 authentication built with the HasXdmAuth config option is required on all X Windows systems. Have the latest version of the software installed and the appropriate software patches applied. Use host based IP filtering software to limit network communications to only authorized hosts. On this network X Windows is only allowed from internal admin workstations for managing the DNS servers, therefore not a high risk.

10. Multiple Choice Question:

What may be taking place if you see the following data: “MIT- MAGIC-COOKIE-

1.I dz9HYBvsrVqmo1G6listed”?

Port scan.
Windows attack.
Possible Magic Cookie attack.
Poison DNS attack.

Answer: Possible Magic Cookie attack.

Network detect (2): ICMP Redirect Host

An excerpt was taken out of the logs for purposes of brevity.

Alert # 9 – Packet Decode:

Alert ID	Signature	Time Stamp	Source Address	Dest. Address	Layer 4 Proto
#0 - NID-xyz	ICMP Redirect Host	2002-08-10 01:36:18	host.my.net.134:58698	Host.my.net.191.21:25	TCP
#1 - NID-xyz	ICMP Redirect Host	2002-08-10 01:36:27	host.my.net.134:58699	Host.my.net.191.28:25	TCP
#2 - NID-xyz	ICMP Redirect Host	2002-08-10 01:36:48	host.my.net.134:58699	Host.my.net.191.28:25	TCP
#3 - NID-xyz	ICMP Redirect Host	2002-08-10 01:37:17	host.my.net.134:58699	Host.my.net.191.28:25	TCP
#4 - NID-xyz	ICMP Redirect Host	2002-08-10 01:37:31	host.my.net.134:58699	Host.my.net.191.28:25	TCP
#5 - NID-xyz	ICMP Redirect Host	2002-08-10 01:37:38	host.my.net.134:58699	Host.my.net.191.28:25	TCP
#6 - NID-xyz	ICMP Redirect Host	2002-08-10 01:37:49	host.my.net.134:58698	Host.my.net.191.21:25	TCP
#7 - NID-xyz	ICMP Redirect Host	2002-08-10 01:37:58	host.my.net134:58698	Host.my.net.191.21:25	TCP
#8 - NID-xyz	ICMP Redirect Host	2002-08-10 01:37:11	host.my.net134:58698	Host.my.net.191.21:25	TCP
#9 - NID-xyz	ICMP Redirect Host	2002-08-10 01:37:33	host.my.net134:58698	Host.my.net.191.21:25	TCP

	Sensor
Name Interface Filter	NID-xyz hme0 *None*
	Alert Group
	None

IP	
Source addr	Host.my.net.174
Dest addr	Host.my.net.34
Ver	4
Hdr Len	5
TOS	192
Length	88
ID	25032
Flags	0
Offset	0
TTL	240
Chksum	55537

	FQDN
	Source Name Dest. Name grid-9-internet-solutions.LondonInt.cw.net host-on.my.net
	Options
	None

ICMP	Type Code Checksum Id Seq #
	(5) Redirect (1) 1130 0 0

Payload	length = 64 000 : 00 00 00 00 00 00 00 00 00 00 00 00 00 0 00 00 00 …… …….....<….... 010 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ……..........-4.. 020 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 …….......... 030 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ……..........;.W....

Whois resolution for source address: host.my.net.174
Server used for this query: [whois.arin.net]

Cable & Wireless USA (NET-CW-NETCS2)
9000 Regency Parkway, Suite 200
Cary, NC 27511
US

Netname: CW-NETCS2
Netblock: 166.63.0.0 - 166.63.255.255

Coordinator:
Cable & Wireless US (IA3-ORG-ARIN) ipadmin@clp.cw.net
1-800-977-4662

Domain System inverse mapping provided by:

NS.CW.NET	204.70.128.1
NS2.CW.NET	204.70.57.242
NS3.CW.NET	204.70.25.234
NS4.CW.NET	204.70.49.234

1. Source of trace:

The traces above were collected from a Department of Defense network. The destination Internet Protocol addresses, as well as the hex data, for the DoD network has been sanitized in both the alert listings and the packet decodes. Note also that the name of the sensor that generated each alert has been obfuscated in the alert listings and packet decodes. Furthermore, the Fully Qualified Domain Name (FQDN) for all DoD destination hosts has been changed to “host-on.my.net”.

2. Detect was generated by:

This detect was generated by a Network Intrusion Detection System that uses SNORT standard rules in addition to other NIDS tools to detect anomalous activity in IP data flows. The SNORT output plug-in is configured to use the XML logging format. The XML data is logged by the sensor and retrieved at regular intervals by an analysis server. The analysis server uses custom PERL scripts

to parse the data and log it to a MySQL database. The alert listings and packet decodes shown above were formatted by the Analysis Console for Intrusion Databases (ACID). "The Analysis Console for Intrusion Databases (ACID) is a PHP-based analysis engine to search and process a database of security events generated by various IDSs, firewalls, and network monitoring tools." - http://www.cert.org/kb/acid/ . The ACID application is open source and as such freely available to all. Please visit the URL above to learn more about this tool.

Snort Signature:

alert ICMP $EXTERNAL any -> $INTERNAL any (msg: "IDS199/icmp_icmp-redirect_net"; itype: 5; icode: 1; classtype: denialofservice; reference: arachnids,199;)

3. Probability the source address was spoofed:

Definitely: ICMP does not require a response. TCPdump analysis of ICMP payload data indicates, although sanitized here in the paper, the ICMP redirect packet arrived at the external interface with a source IP corresponding to an internal IP, clearly indicating spoofing. The new route embedded in the ICMP payload was to an invalid IP, 10.0.0.1, thus causing a DoS. This type of attack can also be used to exploit a trust relationship between hosts.

4. Description of the attack:

The redirect message is issued from a router to inform a host of a better route to a requested destination. The host then updates its routing table to include this route. This method of updating routing tables is an uncommon practice today. When non-specific network traffic of this type is encountered, shuning or disallowing the ICMP redirect would be the most prudent action, from a security perspective. These can also indicate a routing problem. This attack does not work on routers.

CVE : http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0265

5. Attack mechanism:

Attackers can use ICMP redirect messages to place incorrect routes into a target host's routing table, thereby enabling IP spoofing, data capture, or a denial-of-service attack. ICMP redirects are generated only by routers not by hosts. ICMP redirects are intended to be used by hosts not routers; therefore,

the attacker must spoof the source address of the ICMP packet so that it appears to come from the victim's normal router. The new router specified by the ICMP redirect message must be on the same LAN segment as the target host. Once the target receives the redirects, the kernel updates the routing tables. It should be noted, this alert could be caused by legitimate ICMP redirect traffic, where a router is actually informing a host a route is not optimal. Based on the ICMP payload data for the route, 10.0.0.1, this is not legitimate traffic.

6. Correlations:

A white paper by Yuri Volobuev dated 1997:
http://www.insecure.org/sploits/arp.games.html

http://www.whitehats.com/cgi/arachNIDS/Show?_id=ids199&view=protocol

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0265

7. Evidence of active targeting:

This appears to be active targeting of specific host addresses within the network. One host had its routing table altered by this attack. The alert listing above was reduced due to space constraints. The source host actually transmitted a total of 41 packets over a period of approximately two minutes. A host will disregard ICMP redirects until it reaches its default threshold which is unique to each operating system. The source is unaware of the number of ICMP redirects necessary to cause a target to change its routing table; therefore, the source sends out numerous redirects.

8. Severity:

The formula used to calculate the severity of an incident is as follows:

(Criticality + Lethality) – (System Countermeasures + Network Countermeasures)

Each category is assigned a value on a scale of (1) to (5). Five is the most significant value that may be assigned to each category.

Criticality:	*2*
The host targeted by this attack were workstations on the network.	
Lethality:	*5*

	The attack caused a temporary Denial of Service
System Countermeasures:	*1*
	The system will respond to ICMP redirect packets.
Network Countermeasures:	*1*
	There is currently no filtering of ICMP redirect packets.
Overall Severity:	**5**

* The chart above builds upon the work of Mr. James Conz. CGIA practical: http://www.giac.org/practical/James_Conz_GCIA.doc

9. Defensive recommendation:

Blocking incoming ICMP redirect messages at all border routers would be the most obvious and most effective defensive step. You would also want to do the same for the outgoing direction since you don't want to be the source of the attack either. To prevent spoofing, the router should drop all packets arriving at the external interface that have a source address corresponding to the internal network. If you can not block ICMP, strict checking should be performed on each ICMP redirect message received. For example, the new router must be directly connected to the network, the redirect must be from the current router, the ICMP redirect message can not tell the host to use itself as the router, and the route being modified must be an indirect route. The scope of redirect messages should be limited to a connection rather than to the global routing table. Administrators should be suspicious when a host is receiving excessive redirect messages. On a Cisco router, the following line on the ACL applied to the incoming direction on the Internet interface will block ICMP redirect messages coming from the Internet: access-list 101 deny icmp any any redirect. For Unix type operating systems complete the following steps as it applies to your specific OS:

Add the commands below to the /etc/rc.d/rc.local script

```
f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
echo 0 > $f
done
```

Edit the /etc/sysctl.conf file and add the following line:

Disable ICMP Redirect Acceptance
net.ipv4.conf.all.accept_redirects = 0

10. Multiple Choice Question:

The new route (IP) specified by the ICMP redirect packet can be which of the following?

The host's own IP.
Another router on the same LAN segment as the host.
Does not need to specify a new router.
None of the above.

Answer: Another router on the same LAN segment as the host

Network detect (3): Tiny Fragmentation

The following detect was submitted to incidents.org, as per the SANS GIAC GCIA Aministrivia. A period of two weeks had passed since I submitted the detect with only one peer, Dr. Anton Chuvakin, a GCIA Advisory board member providing any feed back as to my analysis of the detect. Dr. Chuvakin provided insightful feedback; however, I did not have a large amount of questions to choose from in order to select the three best to respond to. In fact I only received two questions. Dr. Chuvakin advised me to answer the questions I had received and that three questions were not set in stone. At his direction I have completed what I was able due to circumstances beyond my control.

1. Source of Trace:

Source data is from a Snort tcpdump log from http://www.incidents.org/logs/Raw/2002.6.8. The tcpdump output for this detect is as follows:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=
09/08-07:56:32.204488 80.1.36.6 -> 46.5.237.133
TCP TTL:111 TOS:0x0 ID:49885 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0   Frag Size: 0x5A8
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=
09/08-07:59:44.724488 80.1.36.6 -> 46.5.237.133
TCP TTL:111 TOS:0x0 ID:62656 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0   Frag Size: 0x5A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=
09/08-08:03:44.814488 80.1.36.6 -> 46.5.237.133
TCP TTL:111 TOS:0x0 ID:13456 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0   Frag Size: 0x5A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=
09/08-08:07:45.554488 80.1.36.6 -> 46.5.237.133
TCP TTL:111 TOS:0x0 ID:29434 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0   Frag Size: 0x5A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=
```

2. Detect Generated By:

The Snort alert and log data are generated by Snort IDS 1.8.3 Win32 version with the Snort 1.8.6 rule set. The actual snort rule set is unknown. Snort may have initially hit on the "MF" and "DF" flags both being set or that the fragment offset is zero, indicating first fragment, but there is no protocol information. Protocol information is only included in the first packet of a fragment, thus this indicates this is a subsequent packet.

3. Probability the Source Address Was Spoofed:

Low. In order for the source to glean any useful information using this approach, they need to receive a response from the target. The source may be looking to ascertain what operating system is on the target, what services are running, or the MTU of the path from source to destination, all which require a response. If the source is able to sniff the return path of the packets, they could spoof these packet's IPs and still be able to gather the data on their return.

4. Description of Attack:

With many IP implementations it is possible to impose an unusually small fragment size on outgoing packets. If the fragment size is made small enough to force some of a TCP packet's TCP header fields into the second fragment, filter rules that specify patterns for those fields will not match. If the filtering implementation does not enforce a minimum fragment size, a disallowed packet

might be passed because it didn't hit a match in the filter. This alert is symbolic of an Nmap scan to determine target OS.

5. Attack Mechanism:

This attack works by attempting a couple of things. First, by setting both the “Don't Fragment” and the “More Fragments” bits, it is possible to confuse an IDS or firewall and avoid detection. Second, although the fragment offset indicates this is the first packet, there is no protocol information, which would indicate this is a subsequent packet. The attacker crafted the IP fragment packets with bad data. Attackers using this technique were able to penetrate older firewalls and filtering routers. Firewalls and IDSs would assume that this is just another fragment that has already passed their access control lists. If the targets did not exist, a router would send back a host unreachable message. Taking the inverse of the host unreachable messages would reveal all the live hosts.

6. Correlations:

The links listed below deal with how fragmentation is handled in firewalls and intrusion detection systems. Once the packets reach the target host they are reassembled in the IP stack and the malicious code, which would have been stopped by the firewall or IDS had it not been fragmented, is executed. In older systems, as well as systems that do not keep packet state, fragmentation is used to bypass them.

http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0804

http://www.checkpoint.com/techsupport/alerts/list_vun.html

http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0602

http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0588

7. Evidence of Active Targeting:

This attack indicates the same source and destination in all the packets. This indicates that a specific system was targeted.

8. Severity:

Since this attack was captured from an unknown network using an unknown rule set, the following is assumed: The attack is old and wide known. The

network is using Stateful intrusion detection systems and firewalls.

Criticality: 3

Lethality: 2

System Countermeasures: 5

Network Countermeasures: 5

Severity = (criticality + lethality) - (system countermeasures + network countermeasures) = -5.

9. Defensive Recommendation:

A router can prevent this sort of attack by enforcing certain limits on fragments passing through. Reassemble incoming packets before making filtering and/or intrusion detection decisions. For example, the first fragment should be large enough to contain all the necessary header information. Employing Stateful firewalls and intrusion detections systems will increase your security posture. Be aware, however, they take up more system resources due to the fact that they must keep packet state.

10. Multiple Choice Question:

If you analyze a TCP packet where the first fragment contains only eight octets of data (the minimum fragment size) what type of packet did you receive?

The first packet
The last packet
A blank packet
A crafted packet

Answer: D. A crafted packet.

Question 1: What about other correlations, such as by IP, by time, etc?

Answer: Alternate correlations to this detect, such as those indicated in the above question, were considered but came to no fruition. A review of logs

immediately before and after did not reveal any information pertaining to this detect. Correlation by IP or time was not able to be done in this case. Due to this situation, the four packets were evaluated on their own merit. This is not to say that there was no data to be extracted in any of the logs posted, but that I only reviewed one set before and one set after the initial log set. This was done for time sake. A more meticulous review of all the logs could have been done were there more time.

Question 2: Can you come up with any other information other than the above statement for active targeting?

Answer: After receiving this question, I went back and reviewed the logs for additional data. I conducted a review of the logs trying to identify prior similar activity from the same IP or subnet. There was not convincing activity that I was able to discern from the logs, such as prior activity from the same IP or subnet. Again, as stated above, the four packets were evaluated on their own merit.

Analysis Summary: My analysis of this detect was based solely on the four packets by themselves. A review of the logs immediately preceding and following these logs was done. No pertinent additional data was found pertaining to this detect. A review of all the logs on the web site may reveal more data about these packets, but was unable to be done due to time constraints and the amount of data. Based on the packets alone, it is evident that these packets were crafted for malicious intent.

Assignment 3 – "Analyze This" Scenario

The following practicals were consulted during this analysis

http://www.giac.org/practical/Hee_So_GCIA.doc
http://www.giac.org/practical/Royans_Tharakan_GCIA.doc
http://www.giac.org/practical/Trenton_Riddell_GCIA.doc

Executive Summary

An information assurance analysis was done on the educational network my.net.x.x. The network seems to be fairly open with limited restrictions, as is evident with the peer to peer activity and suspicious virus/Trojan traffic. It should be noted, however, that this type of configuration is common at many educational institutions. Upon finishing the intrusion analysis, the following are

recommendations for the customer:

- Install Antivirus applications on all systems and perform regular updates. In addition, investigate identified systems in report for compromises.
- Develop and implement security policies, processes, and procedures that address peer to peer application use.
- Install an intrusion detection system to facilitate your network information assurance posture.
- Establish a DMZ for authorized web services.
- Establish strong access control lists on the network border routers.
- Determine which systems are running vulnerable processes, i.e. TFTP, and replace them.

Overview:

The following report summarizes my analysis of the data collected by a SNORT intrusion detection system for five consecutive days from the campus network of an undisclosed university. The data was collected between September 5th and September 9th, 2002. It is important to note that network topology for this network was not provided. Furthermore, the analyst was not provided access to network resources (such as server, web, or firewall logs) to correlate any findings. Therefore, some of the analysis is inconclusive and requires further investigation.

The format of this analysis will be as follows:

Prioritized list of alerts detected with supporting analysis
    Top 10 Talkers
    Top 5 Talkers Information

Prioritized list of scanning activity with supporting analysis
    Top 10 Talkers

Prioritized list of OOS (out of spec) activity with supporting analysis
    Top 10 Talkers
    Top 5 Talkers Information

Defensive recommendations
    Analysis process

The logs files used in this analysis are as follows:

Alert Files	Scan Files	OOS Files
alert.020905	scans. 020905	oos_Sep.05.2002
alert.020906	scans.020906	oos_Sep.06.2002
alert.020907	scans.020907	oos_Sep.07.2002
alert.020908	scans.020908	oos_Sep.08.2002
alert.020909	scans.020909	oos_Sep.09.2002

Detect Summary:

List of alerts detected with supporting analysis:

The alerts in the list below are prioritized by number of occurrences

Signature	# Alerts	# Sources	# Dests
spp_http_decode: IIS Unicode attack detected	54, 328	125	641
connect to 515 from inside	32, 269	84	6
SMB Name Wildcard	57	23	5
Samba Client access	32,842	26	312
NIMDA - Attempt to execute cmd from campus host	16,224	25	23
NMAP TCP ping!	14,843	15	6
DDOS shaft client to handler	5,631	51	50
EXPLOIT x86 NOOP	4,945	71	201
Null scan!	2,536	148	15
SNMP public access	2,327	5	30
SUNRPC highport access!	1,873	38	51
Watchlist 000222 NET-NCFC	1.475	37	51
Watchlist 000220 IL-ISDNNET-990517	1,021	21	5
High port 65535 udp - possible Red Worm - traffic	1,110	12	7
Queso fingerprint	842	24	408
Possible trojan server activity	642	15	27
TFTP - External UDP connection to internal tftp server	512	4	7
spp_http_decode: CGI Null Byte attack detected	204	4	3
Tiny Fragments - Possible Hostile Activity	707	30	15
STATDX UDP attack	754	21	340
Totals: Signatures matched 20	87,014	547	1,551

Supporting analysis

spp_http_decode: IIS Unicode attack detected

Unicode is a standard for providing identifiers for characters in every language to assist in uniform computer representation of the characters used all over the world, see http://www.unicode.org. The vulnerability of IIS is it doesn't decode large Unicode representations until after checking the path, which allows a directory transversal attack, see http://www.securityfocus.com/archive/1/140091. This vulnerability has been categorized as CAN-2000-0884. The IDS recorded 52,161 attempts from 95 sources to 572 destinations. Many of these are likely false positives. The following IPs require further investigation in order to fully determine traffic intentions.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
154.124.62.71	148	274	3	21
65.145.3.25	32	71	6	7
162.248.45.42	21	42	4	3
203.229.99.7	18	37	2	2
214.229.78.54	12	8	6	7
214.229.78.85	7	25	5	5
214.229.78.83	4	21	1	1

Snort Signature

alert TCP $EXTERNAL any -> $INTERNAL 80 (msg: "IDS432/web-iis_http-iis-unicode-traversal"; flags: A+; uricontent: "..|25|c1|25|1c"; nocase; classtype: system-attempt; reference: arachnids,432;)

Correlation:

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0884
http://www.iss.net/security_center/advice/Intrusions/2000639/default.htm

Recommendations:

Apply the patches named by iis_promisc as soon as possible to protect the web servers with IIS 4 and 5. In addition, apply security patch q293826 with the title Windows 2000 Security Patch: Superfluous decoding operation could allow command execution via IIS would fix the following vulnerabilities per Microsoft:

1. A vulnerability that could enable a malicious user to run operating system commands on an affected server.

2. A vulnerability that could allow a malicious user to enter a File Transfer Protocol (FTP) command, which can cause IIS 5.0 to fail. FTP is the protocol used for copying files to and from remote computer systems on a network.

3. A vulnerability that can enable a malicious user to access a guest account using the FTP service.

Connect to 515 from inside

This rule appears to be locally generated. Source MY.NET.121.23 scanned 147 other MY.NET hosts for printer ports. The system administrator will want to check this host out as possibly being compromised. Given the vulnerability on LRPng (CV-E2000-0917), this would indicate a threat.

Sources triggering this attack signature

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total))
MY.NET.121.23	147	1023	1	1
MY.NET.78.121	15	15	1	1
MY.NET.42.87	6	9	2	2
MY.NET.253.14	5	5	1	1
MY.NET.70.38	5	5	1	1
MY.NET.99.244	2	2	1	1
MY.NET.60.16	1	1	1	1
MY.NET.179.78	1	1	1	1
MY.NET.219.122	1	8	1	2
MY.NET.219.194	1	1	1	1
MY.NET.163.17	1	1	1	1

Recommendations:

Disable the printer spooler service on all hosts not intended to act as a printer server. Ensure all print servers have up to date operating system patches. Use ingress/egress filtering for port TCP/UDP 515 to prevent LPR request from entering or leaving the network.

SMB Name Wildcard

Windows machines typically send these types of queries in normal operation, particularly when file sharing is active, to determine NetBIOS names when only IP addresses are known. This type of query, when originating from an external network, is usually a pre-attack probe to gather netbios name table information such as workstation name, domain, and a list of currently logged in users. By accessing system name table information, individuals can obtain information which can be used to launch an attack. Information available includes: 1. The NetBIOS name of the server. 2. The Windows NT workgroup domain name. 3. Login names of users who are logged into the server. 4. The name of the administrator account if they are logged into the server. There were 1,896 alerts with this signature. This equates to 0.78% all alerts observed. The alerts were to and from addresses within the network. The majority of these are most likely false positives encountered during normal operations. However, MY.NET.23.58 was responsible for 425 of the alerts. Further analysis revealed that this host is also generating ICMP Echo Request L3retriever Pings. This host should be check for signs of compromise or misuse.

Correlation:

http://www.sans.org/newlook/resources/IDFAQ/port_137.htm
http://www.geocrawler.com/archives/3/4890/2000/12/0/4883899/

Recommendations:
It is considered best practice to ensure that users outside of your network are not permitted to access the NetBIOS name service. This is usually accomplished by configuring packet filters to drop UDP traffic to port 137. Establish filters on network border devices (firewalls, routers) to filter external request for NETBIOS protocols.

Samba client access

Samba is a client/server system that implements network resource sharing for

Linux and other UNIX computers. With Samba, UNIX files and printers can be shared with Windows clients and vice versa. Samba supports the Session Message Block (SMB) protocol. Nearly all Windows computers include SMB support with their internal network subsystems (NetBIOS in particular). This event indicates an attempt to access the Netbios service using the Unix Samba client. The packet that caused this event is normally a part of an established TCP session, indicating that the source IP address has not been spoofed. If you are using a firewall that supports stateful inspection, and are not vulnerable to sequence number prediction attacks, then you can be fairly certain that the source IP address of the event is accurate. It has been noted that the intruder is likely to expect or desire a response to their packets, so it may be likely that the source IP address is not spoofed.

Snort Signature

alert TCP $EXTERNAL any -> $INTERNAL 139 (msg: "IDS341/netbios_NETBIOS-Samba-clientaccess"; flags: A+; content: "|00|Unix|00|Samba"; classtype: suspicious; reference: arachnids,341;)

Correlation:

http://www.ciac.org/ciac/bulletins/l-105.shtml
http://www.stanford.edu/group/itss-ccs/security/Advisories/98-0190.html

Recommendations:

A temporary fix is to edit your smb.conf configuration file and remove all occurrences of the macro "%m". Replacing occurrences of %m with %I is probably the best workaround for most sites. For a permanent fix you need to upgrade to the current version of SAMBA.

NIMDA - Attempt to execute cmd from campus host

The worm, primarily takes advantage of poorly maintained Microsoft software running on computers that are connected to a network. Computers that are easily compromised by Nimda include; PCs running vulnerable Internet Explorer, servers running vulnerable Internet Information Server (IIS), computers that are configured with insecure “shares” and computers that have not been cleaned to get rid of "root.exe" that was left behind by “Code Red II” or Sadmind worms. Email systems infected with Nimda will collect email addresses, through the MAPI service then it sends multiple messages containing copies of its code as an attachment named (readme.exe) to all the email addresses it has collected. Infection is accomplished automatically by the worm if the user is running a

vulnerable version of Internet Explorer and views or previews the infected message. A user can become infected by double clicking on the attachment. These systems should be examined for possible infection.

Correlation:

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0154
http://www.cert.org/advisories/CA-2001-26.html

Signature:

alert tcp $EXTERNAL_NET any -> $HOME_NET 139 (msg:"NETBIOS nimda .eml"; content:"|00|E|00|M|00|L"; flow:to_server,established; classtype:bad-unknown; reference:url,www.datafellows.com/v-descs/nimda.shtml; sid:1293; rev:6;)

Recommendations:

Protecting your computing landscape against virus and worm attack is highly important, to large organizations as well as small ones. A computer can be infected by a floppy, CD, keyboard, LAN, WAN, modem, or any I./O device. Any input device that communicates with the computer has the potential to be a source for a virus infection. Everyone should have a licensed virus detection application running on all platforms and operating systems. Servers and workstations should have a mechanism that keeps the virus detection and cleaning definitions up to date, automatically if possible. All software and networking equipment should have the latest patches and service packages applied as soon as they become available. As your network increases in size you must scale up the virus protection. If you have an email server it should be protected by network virus scanning software. Educate your users to think before clicking on an attachment. Get them to read the extension of the attached file, and know file extensions that are dangerous. Keep current by joining a security email list. These messages could let you know what to do before a virus or worm attacks your system. Remain vigilant keep yourself and your users educated and updated. Develop a suitable "back-up" schedule for mission critical computers. Develop a strong virus protection policy and appropriate procedures.

NMAP TCP ping!

Nmap is a utility for network exploration or security auditing. It supports ping

scanning (determine which hosts are up), many port scanning techniques (determine what services the hosts are offering), and TCP/IP fingerprinting (remote host operating system identification). Nmap also offers flexible target and port specification, decoy/stealth scanning, sunRPC scanning, and more. Most UNIX and Windows platforms are supported in both GUI and command-line modes. Several popular handheld devices are also supported, including the Sharp Zaurus and the iPAQ. The TCP ping is performed by sending a TCP ACK to a host and listening for a TCP RST. If a TCP RST is received, it is reasonable to assume the host is up.

[**] IDS028 - PING NMAP TCP [**]
09/06-12:51:53.245934 195.54.105.6:80 -> my.net.net:.34 2347
TCP TTL:38 TOS:0x0 ID:10650
******A* Seq: 0x362 Ack: 0x0 Win: 0x578

[**] IDS028 - PING NMAP TCP [**]
09/06-12:51:53.245934 195.54.105.6:80 -> my.net.net:34 2347
TCP TTL:38 TOS:0x0 ID:10650
******A* Seq: 0x362 Ack: 0x0 Win: 0x578

[**] IDS028 - PING NMAP TCP [**]
09/06-12:51:53.245934 195.54.105.6:80 -> my.net.net:34 2347
TCP TTL:38 TOS:0x0 ID:10650
******A* Seq: 0x362 Ack: 0x0 Win: 0x578

Correlation:

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0523
http://www.iss.net/security_center/advice/Intrusions/2000310/default.htm

Snort Signature:

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap TCP";flags:A;ack:0; reference:arachnids,28; classtype:attempted-recon; sid:628; rev:1;)

Recommendations:

Implement a packet filter and firewall to deny all packets connection requests originating from outside our network, and block all known trojan ports. Filters should be put in place to block malformed packets (XMAS, FIN/SYN scans, etc)

and IDS signatures put in place. Additionally, we should double check which daemons are running on our workstations, turn off unneeded ones, and replace needed ones with more secure replacements if possible (i.e. use secure-shell instead of ftp and telnet, use TCP Wrappers, etc.). Finally, we may want to verify patching and logging procedures are being followed.

DDOS shaft client to handler

Shaft belongs in the family of tools such as Trinoo, TFN, Stacheldraht, and TFN2K. Like in those tools, there are handler (or master) and agent programs. The Shaft network is made up of one or more handler programs, Shaftmaster, and a large set of agents, shaftnodes. The attacker uses a telnet program to connect to and communicate with the handlers. This event indicates possible control traffic from the Shaft master to the Shaft handlers. If you must verify that this event represents control traffic, your host may be compromised. Shaft is a distributed denial of service (DDoS) tool

Correlation:

http://www.cert.org/reports/dsit_workshop.pdf
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0138

Signature:

alert tcp $EXTERNAL_NET any -> $HOME_NET 20432 (msg:"DDOS shaft client to handler"; flags: A+; reference:arachnids,254; classtype:attempted-dos; sid:230; rev:1;)

Recommendations:

Keep systems updated with current anti-virus products and block known Trojan ports. In addition, monitor all UDP packets on shared Ethernet segments and look for the tell tale signs of communication between master(s) and daemon(s) as described in http://security.royans.net/info/posts/bugtraq_ddos3.shtml. Unfortunately, this would only occur during an attack, which would likely become known by network throughput degradation and/or reports of denial of service attacks from victim sites.

EXPLOIT x86 NOOP

The rule is looking for the no-op opcode in the ix86 processor architecture. A common technique in buffer overflows where the exact return address may not be known, is to put a long strings of no-ops (one byte instructions to the processor that tell it to do nothing). No matter where in this string of no-ops the execution resumes, the processor will run through a bunch of them and then execute the trojan code at the end. Snort saw the string of "90" bytes and assumed this was an overflow attempt. There is a chance of false positives when MIME encoded traffic and large binary transfers are encountered.

Correlation:

Name	Description
CVE-1999-0139	Buffer overflow in Solaris x86 mkcookie allows local users to obtain root access.
CVE-2000-0316	Buffer overflow in Solaris 7 lp allows local users to gain root privileges via a long -d option.
CVE-2000-0337	Buffer overflow in Xsun X server in Solaris 7 allows local users to gain root privileges via a long -dev parameter.
CAN-1999-1014	** CANDIDATE (under review) ** Buffer overflow in mail command in Solaris 2.7 and 2.7 allows local users to gain privileges via a long -m argument.
CAN-1999-1026	** CANDIDATE (under review) ** aspppd on Solaris 2.5 x86 allows local users to modify arbitrary files and gain root privileges via a symlink attack on the /tmp/.asppp.fifo file.

Snort Signature:

alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"IDS362/EXPLOIT x86 NOPS"; content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90| ";reference:arachnids,362;)

Sample packet

[**] EXPLOIT x86 NOOP [**]
09/07-17:03:18.488815 205.188.252.121:80 -> my.net.net.100:3345

TCP TTL:36 TOS:0x0 ID:57358 IpLen:20 DgmLen:576
A* Seq: 0x25CA3454 Ack: 0x62184F92 Win: 0x7FFF TcpLen: 20

Recommedations:

Remove system from the network and perform forensic analysis. Add source IPs to a block list ACL on the border router.

Null scan!

The Null scan alert indicates that a packet was received without any of the TCP flags (SYN, ACK, RST, FIN, PSH, URG, R0,R1) set. This is definitely an anomalous packet. This scan is a member of the stealth family of scans. The general concept behind the scan is that an open port will drop the packet where as a closed port will generate a TCP RST response. Because there is no defined way to responds to this type of request, individual operating systems will generate unique responses to this type of scan. One advantage to this scan is that in addition to simply mapping ports, the scanner may be able to determine the operating system of the remote host by examining the response received. A total of 2354 NULL scans were detected. of those, 172 of these were from MY.NET.168.34.

Correlation:

http://www.nwconnection.com/2001_03/cybercrime/
http://www.synnergy.net/downloads/papers/portscan.txt

Snort Signature:

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN NULL";flags:0; seq:0; ack:0; reference:arachnids,4; classtype:attempted-recon; sid:623; rev:1;)

Recommendations:

A stateful inspection or application gateway firewall in conjunction with an IDS would go a long way in detecting and preventing these scans. In addition, establish filters on network border devices to filter external requests.

SNMP public access

This alert is triggered by the presence of the “public” in the datagram of a packet destined for TCP or UDP port 161. The Simple Network Management Protocol (SNMP) protocol uses community strings to authenticate access to Management Information Base (MIB) objects. In the default configuration of many SNMP implementations the community string for read access is “public” and community string for write access is “private” (SANS Institute). Unchanged, this presents a vulnerability to SNMP enabled devices. An attacker can exploit this vulnerability to read configuration information from and write configuration information to a network device. SNMP agents are a default part of the installation for many network devices. There are several vulnerabilities in SNMP as stated in http://www.cert.org/advisories/CA-2002-03.html.

In my analysis a total of 34, 439 events with this signature were noted. This amounts to 14.71% of all events noted. There were 17 sources and 151 destinations, all within the MY.NET.X.X network. Without network architecture information it is impossible to say which of these are legitimate SNMP requests verses illegitimate access attempts. Regardless, this is a poor security practice. All legitimate SNMP agents should be reconfigured to use proprietary community strings.

Correlation:

http://www.sans.org/newlook/resources/IDFAQ/SNMP.htm
http://www.cert.org/advisories/CA-2002-03.html

Recommendations:

Disable SNMP agents on systems where it is not required. In cases where SNMP is required, ensure that the default community strings are changed and strong passwords are utilized. Establish an ACL that limits the use of SNMP to only authorized devices on the network.

Snort Signatures:

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP public

access udp"; content:"public"; reference:cve,CAN-2002-0012; reference:cve,CAN-2002-0013; sid:1411; rev:1; classtype:attempted-recon;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP public access tcp"; content:"public"; reference:cve,CAN-2002-0012; reference:cve,CAN-2002-0013; sid:1412; rev:1; classtype:attempted-recon;)

Attempted Sun RPC high port access
SUNRPC highport access!

These alerts correspond to access or access attempts to high numbered ports commonly used by RPC services. The Port mapper service (TCP/111) is used to map remote procedural calls to the appropriate port for the service requested. These services run at high ports such as TCP 32773 (rpc.ttdbserverd), 32776 (rpc.spray), 32777 (rpc.walld) and 32779 (rpc.cmsd). Many exploits exist for RPC services. Included among these are statd, ttdbserverd, cmsd and ypupdated. Of course, RPC has many legitimate uses as well.

Correlation:

http://www.cert.org/advisories/CA-2000-17.html
http://www.stanford.edu/group/itss-ccs/security/Advisories/99-0010.html
http://online.securityfocus.com/advisories/1721

Snort Signature:

The latest distribution of SNORT signatures has refined this signature into individual signature for each RPC service. Please download the signature distribution and grep “rpc” * to view them.

Recommendations:

Disable RPC services if they are not required. This can be accomplished by editing the /etc/inetd.conf file. Install the latest patches for any services you cannot remove. Regularly search the vendor patch database for new patches

and install them right away. Block the RPC port (port 111) at the border router or firewall. Block the RPC "loopback" ports, 32770-32789 (TCP and UDP). Enable a non-executable stack on those operating systems that support this feature.

Watchlist 000222 NET-NCFC
&
Watchlist 000220 IL-ISDNNET-990517

The table below shows the source addresses that triggered this alert. These are obviously part of a custom Watch list. The Watch list is not part of the current SNORT signature distribution. Notice that each of the addresses below shows signs of KaZaA activity. KaZaA is a peer-to-peer media distribution application and as such subject to the vulnerabilities that are inherent to peer-to-peer services.

Source	# Alerts (sig)	Destination Addresses	Activity Noted	Net-Name / Location
212.179.35.118	315	MY.NET.153.178 MY.NET.153.162 MY.NET.153.148 MY.NET.153.143 MY.NET.153.163 MY.NET.152.19	HTTP KaZaA	IL-ISDNNET-990517 Petach Tikvah, Israel
212.179.35.119	71	MY.NET.153.178 MY.NET.153.162 MY.NET.153.148 MY.NET.153.143 MY.NET.153.163	KaZaA	IL-ISDNNET-990517 Petach Tikvah, Israel
212.179.127.75	52	MY.NET.88.162	KaZaA	ARAVA-DEVELOPMENT-COMPANY-LTD Petach-Tikva, Israel
212.179.27.6	8	MY.NET.150.133	KaZaA	ADI-ASSOCIATION Petach Tikvah, Israel
212.179.28.133	8	MY.NET.5.97 MY.NET.5.128	HTTP	BS-COMPUTERS Petach Tikvah, Israel

212.179.34.114	5	MY.NET.150.133	KaZaA	IL-ISDNNET-990517 Petach Tikvah, Israel
212.179.51.77	3	MY.NET.150.133	KaZaA	SELA-GROUP Petach Tikvah, Israel
212.179.45.195	3	MY.NET.150.133	KaZaA	KIBBUTZ-MATZUVA Petach Tikvah, Israel
212.179.48.2	1	MY.NET.150.143	Unknown	NESS-TECHNOLOGIES Petach Tikvah, Israel
212.179.38.251	1	MY.NET.150.145	KaZaA	IMFOMALL-LTD Petach Tikvah Israel
212.179.45.205	1	MY.NET.150.133	KaZaA	KIBBUTZ-MATZUVA Petach Tikvah, Israel

Correlation:

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

http://www.sans.org/y2k/practical/Mike_Worman_GCIA.doc

Recommendations:

Install and update antivirus applications on all systems. In addition, block all known P2P ports at border routers and firewalls. Check the destination hosts listed in the chart above for signs of compromise.

High port 65535 udp - possible Red Worm - traffic

This alert indicates that a UDP packet with a source or destination port of 65535 has traversed the network. At least two known Trojans use this port and protocol for communication: the Adore worm, and the RC1 Trojan. While Worms and Trojans quite frequently use this port, it is normal for a non-malicious UDP packet to use this port. This increases the opportunity for an IDS to generate false positives. However my analysis revealed that there are a large number of hosts communicating frequently with this port and protocol. Many of the alerts observed used 65535 as the source and the destination port of the packet. The communication patterns show internal to internal, internal to external, and external to internal network communications. The table below lists five hosts that should be considered compromised and taken off the network for forensic evaluation.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.83.50	1249	1138	58	63
MY.NET.83.146	842	651	57	52
MY.NET.116.107	712	712	51	51
MY.NET.6.48	548	521	34	34
MY.NET.6.65	305	241	62	62

Correlation:

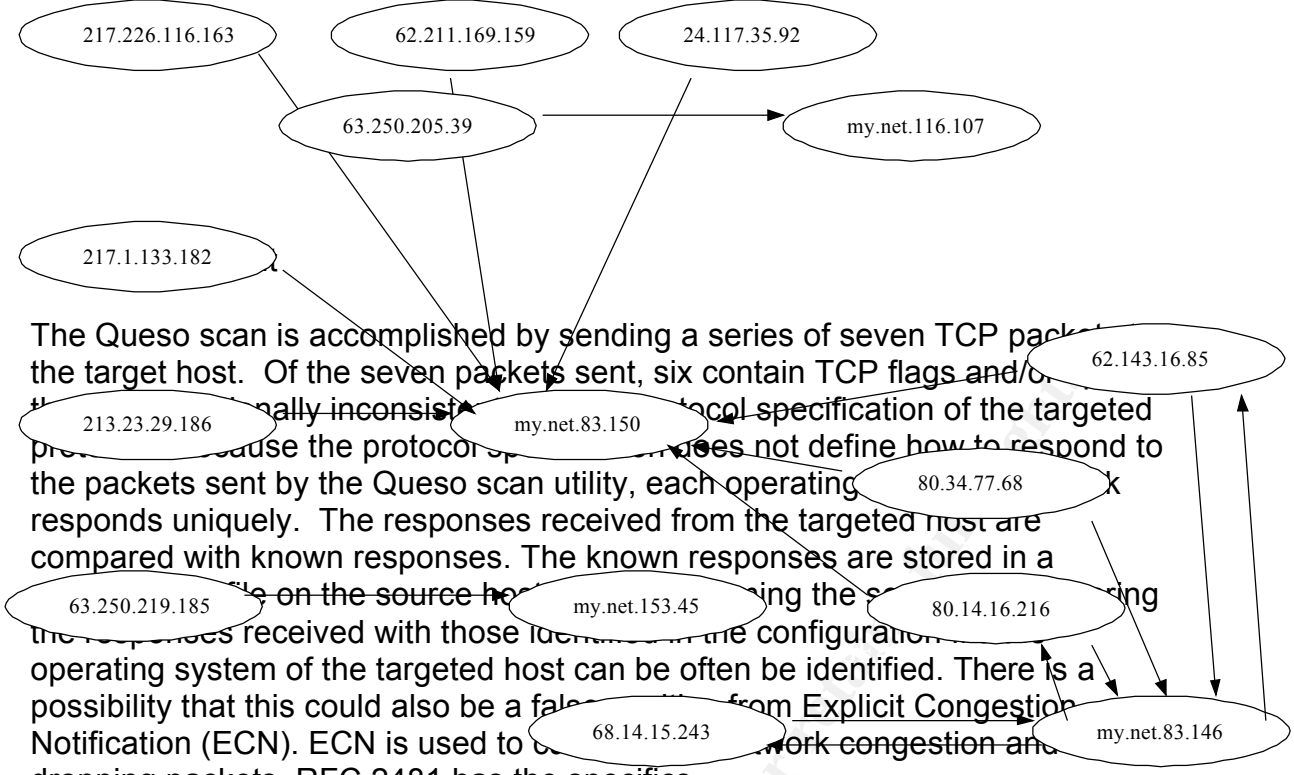
http://rr.sans.org/threats/mutation.php
http://andrew.triumf.ca/ports/sophos.html
http://www.sans.org/y2k/adore.htm
http://www.blackcode.com/trojans/details.php?id=1073

Recommendations:

Remove compromised hosts form the network. Reinstall original OEM applications, DO NOT restore from backups. Use Antivirus software on all systems and update virus signatures regularly. Scan all files before saving them. Develop policy, procedures, and education plan to combat this issue.

Link Graph: High port 65535 udp possible Red Worm - traffic

The link graph depicts the volume of activity going to several hosts on my.net.x.x from both inside and outside the network. The nodes with the highest amount of activity should be examined for possible compromise. These systems comprised a majority of the overall alerts.



The Queso scan is accomplished by sending a series of seven TCP packets to the target host. Of the seven packets sent, six contain TCP flags and/or ... nally inconsistent ... tocol specification of the targeted ... because the protocol ... does not define how to respond to the packets sent by the Queso scan utility, each operating ... responds uniquely. The responses received from the targeted host are compared with known responses. The known responses are stored in a ... file on the source host ... ing the ... ring the responses received with those identified in the configuration ... operating system of the targeted host can be often be identified. There is a possibility that this could also be a false ... from Explicit Congestion Notification (ECN). ECN is used to ... work congestion and dropping packets. RFC 2481 has the specifics.

Correlation:

http://www.wi2600.org/mediawhore/nf0/defcon_archive/SCANNERS/QUESO_980903.TXT
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454
http://www.faqs.org/rfcs/rfc2481.html

Snort Signature:

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"IDS029/SCAN queso fingerprint attempt";flags:S12; reference:arachnids,29;)

Possible trojan server activity

The following hosts were detected communicating to internal as well as external network addresses on port 27374 and port 555. Port 555 has been known to serve the following trojans: Ini-Killer, Net Administrator, Phase Zero, or Stealth Spy. This scan is strictly trying to find hosts running these trojans. Port 27374 is indicative of Bad Blood, SubSeven, SubSeven 2.1 Gold, Subseven 2.1.4 DefCon8.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.114.127	52	52	12	12
MY.NET.9.156	26	10	15	8
12.63.112.7	17	5	1	1

64.78.59.235	10	10	1	1
MY.NET.6.38	7	7	1	1
MY.NET.114.89	5	5	1	1
216.32.120.133	4	7	10	6
MY.NET.114.180	4	4	1	1
MY.NET.114.167	2	2	1	1
195.153.253.43	2	2	1	1

Correlation:

http://www.simovits.com/sve/nyhetsarkiv/1999/nyheter9902.html

Recommendations:

These hosts should be removed from the network and tested for the presence of the above referenced Trojans. Note that the chart also includes four external network addresses. These addresses should be monitored and if this activity continues they should be added to the block list ACL on the network premise router.

TFTP - External UDP connection to internal tftp server

This protocol is a simple form of FTP without the login/password requirements. TFTP uses UDP port 69. TFTP can be used to read from and write files (including configuration files) to the system running the TFTP server. TFTP is a very poor security practice. This alert was generated for seven connections that came from four external source addresses. The destination addresses involved are MY.NET.114.189 and MY.NET.114.167. The TFTP servers on these internal systems should be disabled immediately. These hosts should also be considered compromised due to ease of exploiting this type of vulnerability.

Correlation:

http://www.faqs.org/rfcs/rfc1350.html
http://www.webopedia.com/TERM/T/TFTP.html
http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1123.html

Snort Signature:
alert udp $EXTERNAL_NET any -> $HOME_NET 69 (msg:"TFTP Write"; content:"|00 02|"; depth:2; reference:cve,CVE-1999-0183;

reference:arachnids,148; classtype:bad-unknown; sid:518; rev:2;)

alert udp $EXTERNAL_NET any -> $HOME_NET 69 (msg:"TFTP parent directory"; content:".."; reference:arachnids,137; reference:cve,CVE-1999-0183; classtype:bad-unknown; sid:519; rev:1;)

alert udp $EXTERNAL_NET any -> $HOME_NET 69 (msg:"TFTP root directory"; content:"|0001|/"; reference:arachnids,138; reference:cve,CVE-1999-0183; classtype:bad-unknown; sid:520; rev:2;)

spp_http_decode: CGI Null Byte attack detected

This alert indicates that the string %00 was found in a packet destined for TCP port 80/443 (http/https). This is indicative of the CGI null byte exploit. The exploit is accomplished by strategically inserting the null character %00 into the URL request to a CGI script. PERL accepts null characters as a valid part of string variables. The C libraries that handle system calls interpret the null character as a delimiter. You may see false positives with sites that use cookies with url encoded binary data, or if you're scanning port 443 and picking up SSL encrypted traffic.

For example if the string "../../etc/passwd%00.txt.something.else" when passed to a CGI script written in PERL will be interpreted as "../../etc/passwd\0.txt.something.else". The C libraries that process the system calls from the PERL script will interpret the null character as a delimiter, thus the string becomes "../../etc/passwd" (Rain Forrest Puppy). Cookies and encrypted SSL traffic contribute to the many false positives generated by this alert.

Correlation:

http://www.snort.org/docs/faq.html#4.12
http://www.phrack.com/phrack/55/P55-07

Recommendations:

Suggestions for improving the security of a Common Gateway Interface can be found at http://rr.sans.org/threats/CGI_basics.php

Tiny Fragments - Possible Hostile Activity

Tiny fragmented packets typically are up to no good. This attack uses small fragments to force some of the TCP header information into the next fragment. They can be used for Denial of Service type attacks or for reconnaissance mapping. They can also be used to avoid detection since most IDS don't reassemble the packets to thoroughly examine the payload. The Snort alert keys on the fact that packet fragments are less than 25 bytes.

Correlation:

http://www.digital-minds.org/Network/fragmentation_attacks.pdf
http://archives.neohapsis.com/archives/snort/2000-05/0103.html
http://www.securent-2000.com/article.php?sid=46

Signature:

alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"MISC Tiny Fragments"; fragbits:M; dsize: < 25; classtype:bad-unknown; sid:522; rev:1;)

STATDX UDP attack

Statd exploits Unix flavored systems. It works by exploiting a buffer overflow through rpc and drops you into root on a remote system. This event is specific to a particular exploit and is detected based on a particular string of characters found in the packet payload. Signatures for this event are very specific. SANS Institute lists this as number 3 on the Top Ten list. CVE-1999-0018 and CVE-1999-0019 report this attack.

Correlation:

http://www.cert.org/advisories/CA-97.26.statd.html
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0018
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0019

Signature:
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"RPC EXPLOIT

statdx"; content: "/bin|c74604|/sh"; reference:arachnids,442; classtype:attempted-admin; sid:1282; rev:1;)

Top ten talkers:

The most active source IPs are shown. Rank is determined by the number of alerts with that IP as the source. Within a rank, IPs are sorted by # of signatures, then by IP number

Rank	Total # Alerts	**Source IP**
rank #1	27083 alerts	my.net.81.37
rank #2	11472 alerts	3.0.0.99
rank #3	8375 alerts	194.98.189.139
rank #4	2670 alerts	212.179.66.17
rank #5	2426 alerts	140.90.198.134
rank #6	2109 alerts	63.21.4.50
rank #7	1362 alerts	my.net.111.231
rank #8	1360 alerts	my.net.111.230
rank #9	1349 alerts	my.net.109.105
rank #10	1332 alerts	my.net.111.219

The most active destination IPs are shown. Rank is determined by the number of alerts with that IP as the destination. Within a rank, IPs are sorted by # of signatures, then by IP number.

Rank	Total # Alerts	**Destination IP**
rank #1	27083 alerts	216.241.219.28
rank #2	11472 alerts	10.0.0.1
rank #3	5403 alerts	192.168.0.216
rank #4	2514 alerts	my.net.154.27
rank #5	1837 alerts	my.net.104.204
rank #6	1541 alerts	my.net.153.196
rank #7	1312 alerts	my.net.117.137
rank #8	1258 alerts	my.net.153.159
rank #9	625 alerts	my.net.163.107
rank #10	415 alerts	my.net.110.224

Alert TOP 5 Registration Information

The following is information about the top 5 of the top 10 Talkers based on number of alerts.

OrgName: My.net
OrgID: my.net

NetRange: my.net.0.0 - my.net.255.255
CIDR: my.net.0.0/16
NetName: my.net
NetHandle: NET-my-net-0-0-1
Parent: NET-my-0-0-0-0
NetType: Direct Assignment
NameServer: my5.net.EDU
NameServer: my4.net.EDU
NameServer: my3.net.EDU
Comment:
RegDate: 1988-07-05
Updated: 2000-03-17

TechHandle: JJS41-ARIN
TechName: Suess, John
TechPhone: +1-xxx-xxx-xxxx
TechEmail: my@net.edu

OrgName: General Electric Company
OrgID: GENERA-9

NetRange: 3.0.0.0 - 3.255.255.255
CIDR: 3.0.0.0/8
NetName: GE-INTERNET
NetHandle: NET-3-0-0-0-1
Parent:
NetType: Direct Assignment
NameServer: ns.ge.com
NameServer: ns1.ge.com
NameServer: ns2.ge.com
Comment:
RegDate: 1988-02-23

Updated: 2002-09-26

TechHandle: GET2-ORG-ARIN
TechName: General Electric Company
TechPhone: +1-518-612-6672
TechEmail: genictech@ge.com

---

inetnum: 194.98.189.128 - 194.98.189.143
netname: INGENCYS-NET1
descr: INGENCYS
country: FR
admin-c: DR5-RIPE
tech-c: JB371-RIPE
status: ASSIGNED PA
remarks: abuse@fr.uu.net
mnt-by: IWAY-NOC
changed: frederic.martzel@mciworldcom.fr 20010924
source: RIPE
route: 194.98.0.0/16
descr: UUNET-BLOCK1
descr: UUNET France Block 1
origin: AS702
remarks: **********
remarks: For all spamming or hacking problems
remarks: please send your requests directly to
remarks: abuse@fr.uu.net
remarks: **********
notify: net-adm@mciworldcom.fr
mnt-by: IWAY-NOC
changed: net-adm@iway.fr 19981109
changed: frederic.martzel@mciworldcom.fr 20011114
source: RIPE
role: technical contact
address: UUNET FRANCE
address: 215, Avenue Georges Clemenceau
address: F-92024 NANTERRE Cedex
phone: +33 1 56 38 22 00
fax-no: +33 1 56 38 22 01
e-mail: net-adm@mciworldcom.fr
admin-c: VP1616-RIPE
admin-c: FM7174-RIPE
admin-c: AW7486-RIPE

tech-c: ZM321-RIPE
tech-c: AH6610-RIPE
tech-c: TC334-RIPE
nic-hdl: JB371-RIPE
remarks: ----------
remarks: For all spamming or hacking problems
remarks: please send your requests directly to
remarks: abuse@fr.uu.net
remarks: ----------
mnt-by: IWAY-NOC
changed: frederic.martzel@mciworldcom.fr 20010828
source: RIPE
person: Monsieur De Royer
address: INGENCYS
address: 4, Rue de la Madeleine
address: 45140 ST JEAN DE LA RUELLE, France
phone: +33 2 37 25 12 00
fax-no: +33 2 37 25 12 00
nic-hdl: DR5-RIPE
mnt-by: IWAY-NOC
changed: frederic.martzel@mciworldcom.fr 20010924
source: RIPE

---

inetnum: 212.179.66.16 - 212.179.66.31
netname: IMESH
mnt-by: INET-MGR
descr: IMESH-LAN
country: IL
admin-c: MR916-RIPE
tech-c: ZV140-RIPE
status: ASSIGNED PA
notify: hostmaster@bezeqint.net
changed: hostmaster@bezeqint.net 20020825
source: RIPE
route: 212.179.64.0/18
descr: ISDN Net Ltd.
origin: AS8551
notify: hostmaster@bezeqint.net
mnt-by: AS8551-MNT
changed: hostmaster@bezeqint.net 20020618
source: RIPE
person: Miri Roaky

address: Bezeq Inernational
address: hashacham 40
address: Petach Tiqua
address: Israel
phone: +972-3-9203010
phone: +972-3-9203005
e-mail: hostmaster@bezeqint.net
nic-hdl: MR916-RIPE
changed: hostmaster@bezeqint.net 20020502
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

OrgName: National Oceanic and Atmospheric Administration
OrgID: NOAA-8

NetRange: 140.90.0.0 - 140.90.255.255
CIDR: 140.90.0.0/16
NetName: NOAA-NET
NetHandle: NET-140-90-0-0-1
Parent: NET-140-0-0-0-0
NetType: Direct Assignment
NameServer: NEWNS.NOAA.GOV
NameServer: NWRNS.NOAA.GOV
NameServer: SERNS.NOAA.GOV
NameServer: MERNS.NOAA.GOV
Comment:
RegDate: 1990-04-09
Updated: 2002-01-09

TechHandle: JK79-ARIN
TechName: Kyler, John
TechPhone: +1-301-713-0600
TechEmail: John.C.Kyler@noaa.gov

Network Scanning Activity

Total network scans recorded amounted to 6,708,295. The breakdown is listed below.

Top 5 network scans:

The chart below shows the top 5 scans recorded by the network IDS during the five day period for which analysis was conducted.

Signature (click for sig info)	# Alerts	# Sources	# Dests	Scan Signature
UDP scan	2873008	624	42156	Various UDP scans
TCP ******S* scan	315246	471	27548	SYN Scan
TCP *******F scan	415	8	451	FIN Scan
TCP ******** scan	309	53	14	NULL Scan
TCP **UAPRSF scan	356	128	13	XMAS

UDP Scan

In order to find UDP ports, the attacker generally sends empty UDP datagrams at the port. If the port is listening, the service will send back an error message or ignore the incoming datagram. If the port is closed, then the operating system sends back an "ICMP Port Unreachable" message.

SYN Scan

A TCP SYN scan is the most common form of scanning. The TCP SYN scan takes advantage of the basic TCP three-way handshake. The scanner transmits a SYN packet and waits for the SYN/ACK response from the target. If a SYN/ACK response is received, it means that a system is listening on that port. Responses are recorded for possible exploitation at a later time.

FIN Scan

The FIN scan attempts to close a connection that isn't open. If no service is listening at the target port, the operating system will generate an error message. If a service is listening, the operating system will silently drop the incoming packet. Therefore, no response indicates a listening service at the port. However, since packets can be dropped accidentally on the wire or by firewalls, this isn't a very effective scan.

NULL Scan

The Null scan alert indicates that a packet was received without any of the TCP flags (SYN, ACK, RST, FIN, PSH, URG, R0,R1) set. This is definitely an anomalous packet. This scan is a member of the stealth family of scans. The general concept behind the scan is that an open port will drop the packet where as a closed port will generate a TCP RST response. Because there is no defined way to responds to this type of request, individual operating systems will generate unique responses to this type of scan. One advantage to this scan is that in addition to simply mapping ports, the scanner may be able to determine the operating system of the remote host by examining the response received.

XMAS Scan

The XMAS scan sets all TCP flags (ACK, FIN, RST, SYN, URG, PSH). This is also of form of inverse scanning, meaning an open port will drop the packet and a closed port responds with a TCP RST. This scan has the added advantage of TCP OS fingerprinting. This packet should never be seen in normal TCP operation.

Top 10 Talkers

The most active source IPs are shown. Rank is determined by the number of alerts with that IP as the source. Within a rank, IPs are sorted by # of signatures, then by IP number.

Rank	Total # Alerts	Destination IP	# Signatures triggered	Originating sources
rank #1	3797 alerts	my.net.75.114	1 signatures	63.210.46.141
rank #2	3541 alerts	my.net.184.23	1 signatures	(6 source IPs)
rank #3	3378 alerts	my.net.182.91	2 signatures	(7 source IPs)
rank #4	2986 alerts	my.net.111.194	2 signatures	(7 source IPs)
rank #5	2951 alerts	my.net.146.15	2 signatures	(7 source IPs)

rank #6	2416 alerts	my.net.106.154	2 signatures	(7 source IPs)
rank #7	2407 alerts	my.net.86.28	2 signatures	(8 source IPs)
rank #8	2347 alerts	my.net.178.41	2 signatures	(7 source IPs)
rank #9	2329 alerts	my.net.145.198	2 signatures	(7 source IPs)
rank #10	2204 alerts	my.net.116.69	1 signatures	(6 source IPs)

The most active destination IPs are shown. Rank is determined by the number of alerts with that IP as the destination. Within a rank, IPs are sorted by # of signatures, then by IP number.

Rank	Total # Alerts	Destination IP	# Signatures triggered	Originating sources
rank #1	3797 alerts	my.net.75.114	1 signatures	63.210.46.141
rank #2	3541 alerts	my.net.184.23	1 signatures	(6 source IPs)
rank #3	3378 alerts	my.net.182.91	2 signatures	(7 source IPs)
rank #4	2986 alerts	my.net.111.194	2 signatures	(7 source IPs)
rank #5	2951 alerts	my.net.146.15	2 signatures	(7 source IPs)
rank #6	2416 alerts	my.net.106.154	2 signatures	(7 source IPs)
rank #7	2407 alerts	my.net.86.28	2 signatures	(8 source IPs)
rank #8	2347 alerts	my.net.178.41	2 signatures	(7 source IPs)
rank #9	2329 alerts	my.net.145.198	2 signatures	(7 source IPs)
rank #10	2204 alerts	my.net.116.69	1 signatures	(6 source IPs)

OOS log Analysis

The OOS (out of spec) logs were small enough to be analyzed without automated tools. The OOS logs reviewed included data from September 5th ~ September 9th. An item of interest is that a majority of the traffic during these five days was an attempt at TCP/IP stack fingerprinting.

The first and most obvious series of alerts occurred on September 5th between 00:05:01 and 21:56:42. The first alert is shown below for reference.

=+
=
09/05-00:05:01.429911 198.186.202.147:50839 -> 172.21.253.52:113

TCP TTL:47 TOS:0x0 ID:33000 DF
21S***** Seq: 0x8C64C1FA Ack: 0x0 Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 113066258 0 EOL EOL EOL EOL
=+
=

Queso fingerprint

This is a Queso scan. It is accomplished by sending a series of seven TCP packets to the target host. Of the seven packets sent, six contain TCP flags and/or options that are intentionally inconsistent with the protocol specification of the targeted protocol. Because the protocol specification does not define how to respond to the packets sent by the Queso scan utility, each operating system's IP stack responds uniquely. The responses received from the targeted host are compared with known responses. The known responses are stored in a configuration file on the source host that is performing the scan. By comparing the responses received with those identified in the configuration file the operating system of the targeted host can be often be identified. It is interesting to note that this is probably a Queso variant since the original Queso scan did not include End of Line (EOL) data. There is a possibility that this could also be a false positive from Explicit Congestion Notification (ECN). ECN is used to cut down on network congestion and routers dropping packets. RFC 2481 has the specifics.

Correlation:

http://www.wi2600.org/mediawhore/nf0/defcon_archive/SCANNERS/QUESO_980903.TXT
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454
http://www.faqs.org/rfcs/rfc2481.html

Snort Signature:

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"IDS029/SCAN queso fingerprint attempt";flags:S12; reference:arachnids,29;)

SCAN XMAS

The second pattern to emerge from the logs was an XMAS tree scan. The XMAS scan sets all TCP flags (ACK, FIN, RST, SYN, URG, PSH). This is also of

form of inverse scanning, meaning an open port will drop the packet and a closed port responds with a TCP RST. This scan has the added advantage of TCP OS fingerprinting. This packet should never be seen in normal TCP operation. It is interesting to note that this is probably a variant of the XMAS scan. The original XMAS scan set the sequence number to 0. This packets sequence number is not. The first alert is shown below for reference.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=
09/05-23:32:51.273249 24.249.218.238:0 -> 172.21.225.182:6346
TCP TTL:113 TOS:0x0 ID:41657  DF
21SFRPAU Seq: 0xDDE0161   Ack: 0xD1063E0   Win: 0x5018
TCP Options => EOL EOL
18 CC 65 2E 5D 0F 00 00 00 74                  ..e.]....t
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=
```

Correlation:
http://www.synnergy.net/downloads/papers/portscan.txt

Snort Signature:
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN XMAS";flags:SRAFPU; reference:arachnids,144; classtype:attempted-recon; sid:625; rev:1;)

SYN/FIN SCAN

The last pattern to emerge from the five days of log analysis was a scan of the SYN/FIN type. This SYN-FIN scan is similar to a null scan. This probe sets both the SYN flag and the FIN flag in a TCP packet. This traffic does not occur naturally and indicates an intentional probe. It is probably part of single-packet OS detection. Most SYN-FIN scans are sent as fragments in the hope that they may slip by simple packet filters or firewalls. This is also a good candidate for a variant of the SYN/FIN scan. The original SYN/FIN scan did not include EOL data. The first alert is shown below for reference.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
=
09/05-11:05:17.755937 151.38.84.194:27960 -> 172.21.235.94:27970
TCP TTL:113 TOS:0x0 ID:21370  DF
*1SFR*** Seq: 0x3D6E1C   Ack: 0x35230000   Win: 0x9F
```

TCP Options => EOL EOL EOL EOL EOL EOL SackOK NOP NOP TS: 0 0 EOL EOL EOL EOL
=+
=

Correlation:
http://www.ece.stevens-tech.edu/sd2k_old/grp25/Final_Report.htm
http://www.nwconnection.com/2001_03/cybercrime/

Snort Signature:
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN";flags:SF; reference:arachnids,198; classtype:attempted-recon; sid:624; rev:1;)

OOS TOP 10 Talkers

The OOS top 10 talkers are based on number of alerts received for each IP.

198.186.202.147	24.4.160.163
199.183.24.194	24.28.69.5
128.46.156.155	213.23.38.240
24.19.97.122	24.101.253.5
212.124.64.22	129.74.148.26

OOS TOP 5 Registration Information

The following is information about the top 5 of the top 10 Talkers based on number of alerts.

Country: UNITED STATES
OrgName: Dandelion Digital
OrgID: DANDEL

NetRange: 198.186.200.0 - 198.186.203.255
CIDR: 198.186.200.0/22
NetName: NETBLK-DANDELION-C
NetHandle: NET-198-186-200-0-1

Parent: NET-198-0-0-0-0
NetType: Direct Assignment
NameServer: NS1.VALINUX.COM
NameServer: NS2.VALINUX.COM
Comment:
RegDate: 1993-09-17
Updated: 2001-07-18

TechHandle: LNZ-ARIN
TechName: Zubkoff, Leonard
TechPhone: +1-775-832-1068
TechEmail: lnz@dandelion.com

Country: UNITED STATES
OrgName: ICG NetAhead, Inc.
OrgID: ICGN

NetRange: 199.183.16.0 - 199.183.143.255
CIDR: 199.183.16.0/20, 199.183.32.0/19, 199.183.64.0/18, 199.183.128.0/20
NetName: ICG-BLK-BLK4-C
NetHandle: NET-199-183-16-0-1
Parent: NET-199-0-0-0-0
NetType: Direct Allocation
NameServer: AS1.ICG.NET
NameServer: AS2.ICG.NET
Comment: Addresses within this block are non-portable
RegDate:
Updated: 2001-07-26

TechHandle: ST452-ARIN
TechName: Taylor, Stacy
TechPhone: +1-408-579-5000
TechEmail: abuse@icgcom.com

Country: UNITED STATES
OrgName: Purdue University
OrgID: PURDUE

NetRange: 128.46.0.0 - 128.46.255.255
CIDR: 128.46.0.0/16
NetName: PURDUE-ECN-NET
NetHandle: NET-128-46-0-0-1
Parent: NET-128-0-0-0-0

NetType: Direct Assignment
NameServer: HARBOR.ECN.PURDUE.EDU
NameServer: MOE.RICE.EDU
NameServer: NS.PURDUE.EDU
NameServer: PENDRAGON.CS.PURDUE.EDU
Comment:
RegDate: 1985-01-14
Updated: 1999-05-24

TechHandle: JMM118-ARIN
TechName: Moya, James
TechPhone: +1-765-494-2349
TechEmail: moyman@ecn.purdue.edu

24.19.97.122

The Address space for this IP belongs to ARIN as a Cable Block. It was previously owned by IANA for the same purpose. It was transferred in May 01. I was unable to obtain any additional information on it.

Country: BULGARIA (high)

ARIN says that this IP belongs to RIPE; I'm looking it up there.

% 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/ripencc/pub-services/db/copyright.html

inetnum: 212.124.64.0 - 212.124.67.255
netname: INTERNETBG
descr: INTERNET Bulgaria Ltd. is in operation since 1996 as one of
descr: the first Internet Service Provider for Bulgaria.
country: BG
admin-c: LD57-RIPE
tech-c: LD57-RIPE
tech-c: SCC53
tech-c: BNL1-RIPE
status: ASSIGNED PA
mnt-by: INTERNETBG-MNT

```
notify:       scc@internet-bg.net
notify:       bla@internet-bg.net
changed:      scc@internet-bg.net 20011214
source:       RIPE

route:        212.124.64.0/22
descr:        Internet Bulgaria
origin:       AS9154
mnt-by:       INTERNETBG-MNT
notify:       scc@internet-bg.net
changed:      scc@internet-bg.net 20011214
source:       RIPE

person:       Latchezar Dinev
address:      Internet Bulgaria Ltd.
address:      60,Milin kamak str. ap.1
address:      Sofia, Bulgaria
phone:        +359 2 9631094
phone:        +359 2 9631466
phone:        +359 2 9631752
fax-no:       +359 2 9631552
nic-hdl:      LD57-RIPE
notify:       scc@internet-bg.net
notify:       lnd@internet-bg.net
mnt-by:       INTERNETBG-MNT
changed:      scc@internet-bg.net 20011122
source:       RIPE

person:       Svilen Canov Canov
address:      Internet Bulgaria Ltd.
address:      60,Milin kamak str. ap.1
phone:        +35929631094
fax-no:       +35929631552
e-mail:       scc@internet-bg.net
nic-hdl:      SCC53
notify:       scc@internet-bg.net
mnt-by:       INTERNETBG-MNT
changed:      scc@internet-bg.net 20011122
source:       RIPE

person:       Blagovest Nikolov Lazarov
address:      Internet Bulgaria Ltd.
address:      Sofia, Bulgaria
```

address: 60 , Milin kamak str.
phone: +35929631094
fax-no: +35929631552
e-mail: bla@internet-bg.net
nic-hdl: BNL1-RIPE
mnt-by: INTERNETBG-MNT
notify: bla@internet-bg.net
changed: scc@internet-bg.net 20020607
source: RIPE

Defensive Recommendations

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. Traditionally, the atmosphere surrounding universities is one that is open and encourages experimentation in pursuit of education. Today's Internet culture simply will not allow that mentality to exist with regard to network security. The mindset of those who administer this network must be one of vigilance and ingenuity. This network's security posture could be greatly enhanced through implementing the ideas that follow.

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.

A firewall should be installed at all points where this network is connected to the Internet. If there are any firewalls currently installed on the network their rule sets should be adjusted. Specifically recommendations include:

Establish a deny all, allow by exception policy

Establish a list of trusted external hosts and limit the protocols those hosts may use to access this network.
Establish a rule set that supports internal connections to external entities for common protocols such as HTTP, HTTPS, SMTP, Telnet, FTP, SSH, etc. This rules set should serve as the general policy governing authorized protocols. The use of any other services or protocols should be granted on an as required basis. Justification for the requirement should be submitted for review by the security staff.
Establish a DMZ for authorized web services.

Establish strong access control lists on the network border routers. Specific recommendation included:

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.

Drop all in coming http port/80 requests not destined for authorized web servers in the DMZ.

Filter in coming ICMP packets with code 0, 8 and 30.
Establish an ACL that serves as a block list. As the sources of offending traffic are identified, add them to this ACL

Disable unnecessary services on all systems (i.e. RPC services, SNMP, FTP, anonymous FTP, etc.)

Install Anti-Virus software on all network systems and update virus definitions frequently. Due to the size of this network, a site license for this product should be considered. The Universities network usage policy should mandate the use of Anti-virus software if it does not already do so.

Install the latest vendor system and security patches to all systems on the network. Procedures should be developed to do this at predefined intervals. In addition procedures should be established for do this on an as required basis (i.e. system patches in response to security advisories).

The systems identified in this analysis as being possibly compromised, should be removed from the network. These systems should be formatted and restored from the last known good backup. If backups do not exist these systems should

be rebuilt entirely.

Configure system logging on all servers.

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.

Establish a password policy that enforces the use of strong passwords.

Continue to use IDS to monitor all networks. In addition to IDS logs, review firewall and systems logs at regular intervals.

Analysis Process

The first step in my analysis process was to decide which tools would be used to analyze the data. I decided that I would use SnortSnarf (v. 010821.1) and Snort_stat.pl (v.1.15.2.6) to analyze the alert files and that the portscan logs may require custom shell or perl scripts. The OOS logs appeared small enough to be analyzed manually.

I knew, from reviewing previous practical exams, that it would be necessary to convert all "MY.NET." references to a standard IP address format. I chose the network address "172.21.". To do this I used a sed script from Mr. James Conz's GCIA practical. The script was modified ever so slightly for use in a Solaris environment. I used this script to convert each of the logs (alert, scan and oos) from their original "MY.NET" format to the "172.21" format appending .new to each new file in order to preserve the original data.

Next I created a single log for each of the logs types (alert, scans, and oos) that encompassed the entire five day period to be analyzed. To do this I simply modified the script above to append to a single file vice creating a new file for each log.

```
for file in `ls alert.*.new`
do
cat $file >> master.alerts
done
```

I then used SnortSnarf to process each day's alert log and the five day alert log. The command used to do this was:

Snortsnarf.pl –d /data/snarf/alerts/alert.020905 –homenet 172.21.0.0/16 –split=50 –top=10 /data/alert/alert.020905

In addition to SnortSnarf, I used Snort_stat.pl to create statistics for each days log and the five day log. The command used to do this was:

Cat /data/alert/alert.020905 |snort_stat.pl -f -h /data/snarf/alerts/stats/stats.020905.html

After converting the logs to an html format with the Perl scripts above, I was able to analyze the alert data via a web browser.

I also used SnortSnarf to identify the top TCP scans. After parsing the data with the scripts above, I imported it into Excel so that I could manipulate it a bit more easily. I used Excel to further sort and sum the data that resulted in the charts and tables included in this report.

Said a prayer to help keep me sane through all this.

Appendix A.

```
#!/usr/local/bin/perl
#
#   Start mainline code
while (<>) {
#
#  Check for blank line, if so process next line
#
```

```
   if ( $_ eq "" )  { next };
#
#  Check for spp_portscan, if it is get the next record
#
#   Tokenize the string so we can use it
#
   if ($_ =~ m/^\w{3}\s+\d+\s+\d+\:\d+\:\d+\s+([\w\d\.]+)\:(\d+)\s+\-\>\s+([\d\
w\.]+)\:(\d+)\s+UDP/) {

       $saddr  =      $1;
       $sport  =      $2;
       $daddr  =      $3;
       $dport  =      $4;
       $source{$saddr}++;
   }  # end if

   if ($_ =~ m/^\w{3}\s+\d+\s+\d+\:\d+\:\d+\s+([\w\d\.]+)\:(\d+)\s+\-\>\s+([\d\
w\.]+)\:(\d+)\s+([-\w]+)\s+[\*1PUSFAR]+\s+/) {

       $saddr  =      $1;
       $sport  =      $2;
       $daddr  =      $3;
       $dport  =      $4;
       $descrp =      $5;
       $source{$saddr}++;
   }  # end if

}  # while

foreach $num ( sort keys(%source) ) {
       $strings = $source{$num};
       foreach $string (split(' ', $strings)) {
              print "$string\t$num\n";
       }
}
```

Appendix B.

```
#!/usr/local/bin/perl
#
#   Start mainline code
while (<>) {
#
#  Check for blank line, if so process next line
```

```
#
   if ( $_ eq "" )  { next };
#
#  Check for spp_portscan, if it is get the next record
#
#   Tokenize the string so we can use it
#
   if ($_ =~ m/^\w{3}\s+\d+\s+\d+\:\d+\:\d+\s+([\w\d\.]+)\:(\d+)\s+\-\>\s+([\d\
w\.]+)\:(\d+)\s+UDP/) {

      $saddr  =      $1;
      $sport  =      $2;
      $daddr  =      $3;
      $dport  =      $4;
      $volume{"$saddr $daddr"}++;
   }  # end if

   if ($_ =~ m/^\w{3}\s+\d+\s+\d+\:\d+\:\d+\s+([\w\d\.]+)\:(\d+)\s+\-\>\s+([\d\
w\.]+)\:(\d+)\s+([-\w]+)\s+[\*1PUSFAR]+\s+/) {

      $saddr  =      $1;
      $sport  =      $2;
      $daddr  =      $3;
      $dport  =      $4;
      $descrp =      $5;
      $volume{"$saddr $daddr"}++;
   }  # end if

}  # while

   foreach $pair (sort keys(%volume)) {
      $parts = $volume{$pair} ;
      foreach $number (split(' ', $parts)) {
            print "$number\t$pair\n";

      }
}
```

References:

ISS. “Serious flaw in Microsoft IIS Unicode translation” (26 October 2000)
URL: http://www.iss.net/security_center/alerts/advise68.php
15 February 2002

SANS Institute "Alert: Increased probes to TCP port 515" (20 November 2000)
URL: http://www.sans.org/newlook/alerts/port515.htm
15 February 2002

Romanski James. "Using SNMP for Reconnaissance" (12 August 2000)
URL: http://www.sans.org/newlook/resources/IDFAQ/SNMP.htm
16 February 2002

CERT/CC. "Distributed Denial of Service Tools" (18 Nov 99), (15 January 2001)
URL: http://www.cert.org/incident_notes/IN-99-07.html
16 February 2002

Roesch, Marty. "SNORT FAQ v 1.13" (15 March 2002)
URL: http://www.snort.org/docs/faq.html#4.12
16 February 2002

Puppy, Rain Forest. "Poison NULL byte" (09 October 1999)
Phrack Magazine --- Vol. 9 | Issue 55
URL: http://www.phrack.com/phrack/55/P55-07
18 February 2002

Dell, J. Anthony. "Adore Worm – Another Mutation" (6 April 2001)
URL: http://rr.sans.org/threats/mutation.php
18 February 2002

Daviel, Andrew. "Internet ports and Trojans" (January 2002)
URL: http://andrew.triumf.ca/ports/sophos.html
19 February 2002
Fearnow, Matt. "Adore Worm" (12 April 2001)
URL: http://www.sans.org/y2k/adore.htm
19 February 2002

sili@l0pht.com. "L0pht Security Advisory" (11 August 1999)
URL: http://www.l0pht.com/research/advisories/1999/rdp.txt
19 February 2002

Postel, Jon. "ICMP TYPE NUMBERS" (September 1995), (27 August 2001)
URL: http://www.iana.org/assignments/icmp-parameters
19 February 2002

Deering, S. "Request for Comments: 1256" (September 1991)
URL: http://www.faqs.org/rfcs/rfc1256.html
20 February 2002

Cohen, Dr. Frederick B. "Packet Fragmentation Attacks" (1996)
URL: http://www.all.net/journal/netsec/1995-09.html
20 February 2002

Anderson, Jason. "An Analysis of Fragmentation Attacks" (15 March 2001)
URL: http://rr.sans.org/threats/frag_attacks.php
20 February 2002

Undernet.org "Undernet Scans for Insecure Wingates and Proxies" (Unknown)
URL: http://help.undernet.org/proxyscan/
21 February 2002

Russell, Dan. "Intrusion Detection Practical Assignment" (January 2002)
30 September 2002

Chappell, Laura. "You're Being Watched: Cyber-Crime Scans" (March 2001)
URL: http://www.nwconnection.com/2001_03/cybercrime/
21 February 2002

dethy@synnergy.net. "Examining port scan methods - Analysing Audible Techniques" (2001)
URL: http://www.synnergy.net/downloads/papers/portscan.txt
21 February 2002

Group 25. "SIT Systems Security Upgrade" (5 December 2000)
URL: http://www.ece.stevens-tech.edu/sd2k_old/grp25/Final_Report.htm
21 February 2002

ISS. "Resurgence of "Code Red" Worm Derivatives" (6 August 2001)
URL: http://www.iss.net/security_center/alerts/advise90.php/
22 February 2002

Terhesiu, Dan. "SHELLCODE x86 NOOP" (4 October 2001)
URL: http://www.der-keiler.de/Mailing-Lists/securityfocus/incidents/2001-10/0020.html
22 February 2002

FitzGerald, Nick. "RE: SHELLCODE x86 NOOP" (4 October 2001)
URL:http://www.der-keiler.de/Mailing-Lists/securityfocus/incidents/2001-

10/0032.html
22 February 2002

Von Braun, Joakim, “The Trojan List” (07 December 2001)
URL: http://www.simovits.com/sve/nyhetsarkiv/1999/nyheter9902.html
22 February 2002

CERT/CC. “CERT* Summary CS-97.05” (27 August 1997)
URL: http://www.cert.org/summaries/CS-97.05.html
24 February 2002

Security Focus “Microsoft IIS 5.0 "Translate: f" Source Disclosure Vulnerability” (14 August 2000)
URL: http://online.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=1578
1 March 2002

NIPC Cyernotes “Bugs, Holes and Patches” (3 February 1999) Issue 3-99
URL: http://www.nipc.gov/cybernotes/1999/cyberissue3.pdf
2 March 2002

The yahoo Team. “Common WWW Error Messages” (1994-1996)
URL: http://docs.yahoo.com/docs/writeus/error.html
4 March 2002

Internetqa.com. “HTTP Error Codes Summary” (Unknown)
URL: http://www.internetqa.com/web_tests/links/error_info.htm
4 March 2002

Linux Security.com “NetBSD Security Advisory 2001-004” (5 April 2001)
URL: http://www.linuxsecurity.com/advisories/netbsd_advisory-1255.html
5 March 2002

Frasunek, Przemyslaw. “ntpd =< 4.0.99k remote buffer overflow” (4 April 2001)
URL: http://online.securityfocus.com/archive/1/174011
5 March 2002

Gelman, Herschel. “SANS DC 2000 Practical” (2000)
URL: http://www.sans.org/y2k/practical/Herschel_Gelman.html#overflows
6 March 2002

Jankowski, Rich. “Scanning and Defending Networks with Nmap” (20 February 2000)

URL: http://www.linuxsecurity.com/feature_stories/feature_story-4.html
6 March 2002

CERT/CC. "CERT® Advisory CA-2000-17 Input Validation Problem in rpc.statd" (6 September 2000)
URL: http://www.cert.org/advisories/CA-2000-17.html
6 March 2002

Donahoo. "Port Scanning" (Uknown)
URL: http://cs.baylor.edu/~donahoo/NIUNet/portscan.html
7 March 2002

Worman, Mike. "Re: Connection from unknown" (23 October 2000)
URL: http://lists.insecure.org/incidents/2000/Oct/0141.html
8 March 2002

Heath, Andrew. "I Was rooted" (17 July 2000)
URL: http://archives.neohapsis.com/archives/incidents/2000-07/0081.html
8 March 2002

Murgó, Jordi. "INTRODUCTION TO QueSO." (3 September 1998)
URL:
http://www.wi2600.org/mediawhore/nf0/defcon_archive/SCANNERS/QUESO_980903.TXT
01 February 2002

CERT/CC. "CERT® Advisory CA-1996-26 Denial-of-Service Attack via ping" (5 December 1997)
URL: http://www.cert.org/advisories/CA-1996-26.html
13 March 2002

Microsoft.com "Packets May Be Dropped When A Very Large Number of Fragmented UDP Packets Are in Use (Q255593)" (19 May 2001)
URL: http://support.microsoft.com/default.aspx?scid=kb;EN-US;q255593
13 March 2002
Rieger, Gerhard. "Protocol scan with nmap" (28 May 2000)
URL: http://lists.insecure.org/nmap-hackers/2000/Apr-Jun/0119.html
14 March 2002