Global Information Assurance Certification Paper

GCIA

Practical Assignment

Greg Schultz
Assignment Version: 3.3

Submitted: 18 December 2003

Table of Contents

Abstract:

This paper was developed in three sections to meet the requirements of the GCIA certification. Section one is a paper written to describe a methodology for tuning an open source intrusion detection system (IDS). It is written from the point of view to provide a starting point for a new analyst. It should allow the analyst to begin asking the right questions and give them the ability to jump in and tune their IDS. Section two analyzes three detects. Two of which were taken from my production network IDS and one from the incident.org logs. The final section provides an analysis of five days worth of log files for a university network. The analysis reviews activity on the network based on log files received and provides recommendations for anomalies found during the analysis.

Part 1: Describe the State of Intrusion Detection

One Methodology for Tuning an Open-Source Intrusion Detection System (Based on Using Snort)

The purpose of this section is to provide a methodology for tuning the Intrusion Detection System (IDS). The paper will focus on Snort as a low cost IDS option used as an internal detection system for both signature based and anomaly based detection. This type of system can be used in any type of environment requiring an IDS. Since this is a low cost solution, it can be a primary, temporary, or proof-of-concept system.

The reason for writing this paper is to provide a methodology for tuning and managing the IDS. I accepted ownership of two IDS's, each deployed in an automated production environment. I began to look for documentation to learn how to tune the system to maximize its potential and reduce false positives. I also wanted to provide a secure monitored environment. Many articles that I found high-lighted one problem in particular, that IDS's are a management nightmare with false positives. This paper will provide the methodology that I used to tune my open source IDS. This is only one opinion of the method that I used and may not be the best. My purpose is to provide a baseline to help others get started in tuning their open source IDS.

My assumptions going forward are that a low cost IDS solution such as Snort has been or is being deployed. The analyst is reasonably new to the security field and is having difficulty trying to locate real-life documentation about tuning the IDS. It will not provide data on building the IDS as many sources for this information can be found on the Internet. A good place to start looking is on snort.org's website. This site will provide solutions for simply installing Snort to full-blown database installations. I assume that the analyst is receiving pages and or many alerts and are overwhelmed by the amount of incident handling required. This paper will attempt to provide methods to help reduce the alerts to a manageable amount. The goal of this paper is to reduce IDS alerts to real threats and still provide effective monitoring capabilities.

Foremost, the IDS owner must understand their network. One of the first steps in tuning your IDS should begin with identifying the critical systems in your enterprise. Most of the important servers on my network hang from the central production backbone, which is the most important component of the enterprise. My network has an interoperable mix of operating systems and has the potential to fail if any major disruption occurs at a critical point. Today, I keep vigilance on the Windows systems, but there are also exploits for other operating systems. Many secondary issues may arise as well. One example is a virus that floods the network and overwhelms the routers.

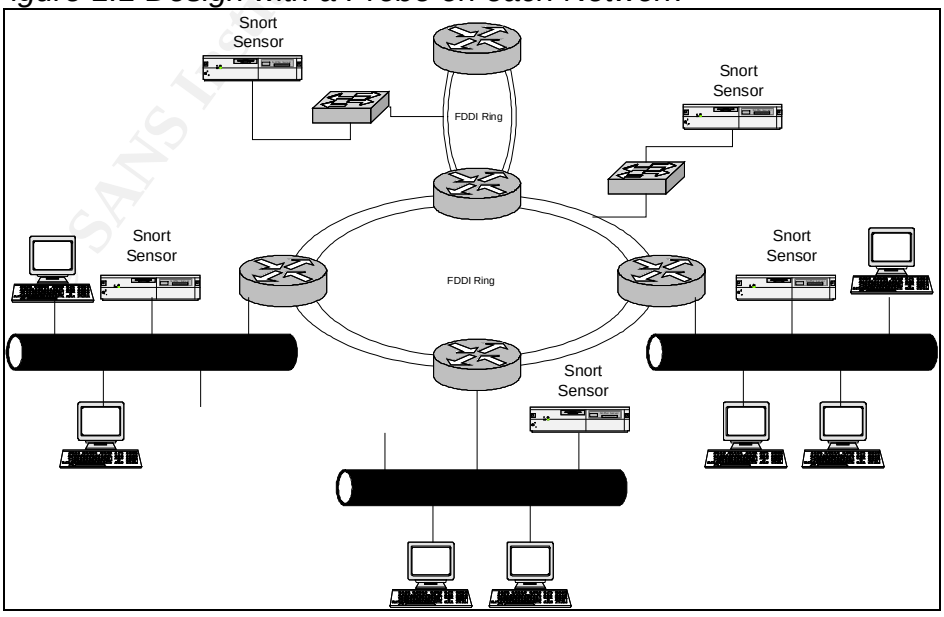
An important first step to pairing down potential false alarms and enhancing performance with Snort is to turn off rules that are not required for your network. A few examples of this are turning off Oracle rules if you do not run Oracle or if all web servers are Apache, turn off Microsoft IIS rules. This seems intuitive but is easy to overlook. Bottom-line; if your enterprise is not susceptible to the exploit, do not make Snort parse through the rules looking for a vulnerability that will never exist. Second, evaluate what you do not want or have no need to evaluate. As an example, I do not have a need to evaluate porn rules or chat rules. While it may be important to watch for this activity in some environments, management has not asked me to monitor for this type of activity to date. Hence these rule sets can be disabled as well.

This paper will not discuss in depth writing of Snort rules. There is plenty of data to be found on actual rule writing. The paper will aid in developing the mindset required to begin the daunting task of tuning. I will include a discussion on probe placement and a layered security model or defense in depth as it pertains to IDS.

Description of the IDS

My system runs the open source Snort application and currently I have version 1.9 deployed. There is a central server running Apache web services, SnortSnarf, and Sysmon for log monitoring. Sensors or probes run Snort and are located on all production subnets as well as choke points behind the firewall. See figure 1.1 for a design layout. In our environment, there are business reasons for having a probe on each subnet. One problem with this design is the amount of false alarms due to the activity seen on the network since we have the potential to view the same traffic multiple times.

Figure 1.1 Design with a Probe on each Network



So Where Do I Put Probes?

There are entire discussions on where the IDS probe or probes should be placed in the enterprise. Basically a good point to start is to identify your critical systems. These are systems that must remain running if the rest of the system fails. Another way to look at it is; how long your business can remain running if individual components in the system begin to fail. The systems that stop the business or must be restarted first are probably your critical systems. If you have a security policy, these systems should be identified in the policy. Another place to look may be the disaster recovery plan as system failure is not the only thing that can stall your business. Data integrity is also another key element. If an intruder is successful in corrupting a mission critical database, what is the mean-time-to-repair (MTTR) before you are operational?

Consider where the traffic enters and exits your network as these are the places where all traffic must pass and can easily be monitored. These are referred to as network choke points. Do not forget the business insider's ability to access your critical systems when considering choke points in your network. According to the Market research firm Gartner, by 2005, 60% of all cyber-terror attacks will come directly or indirectly from business insiders (Jaques, 2003).

Another consideration is the role that the IDS will play in a layered security model. The layered security model is also referred to as defense in depth. How does defense in depth play a role in where the IDS probe is place?

I look at my IDS as the last line of defense. If the intruder gets past the boundary router, through the demilitarized zone (DMZ), and past the firewall then I want to know what they are attempting to do. I am not particularly interested in how much they look through the windows and knock on the door. If this is your interest, you will spend a significant amount of time looking at mountains of alerts. If the attack gets through the systems I mentioned earlier, then the IDS should capture the event so I can perform continuous improvement (CIP) on my up front systems (Intranetworks). I will use the IDS data to review router ACL's and firewall rules. For insider activity, I may use the IDS alerts as a way to provide training if security gaps exist.

Where the IDS should really shine is in its ability to look at insider activity as mentioned earlier. Some insiders such as administrators or database analysts may have direct access to your systems. In this case what comes through the choke point is of no use. This is where a probe directly on the critical network may help. If the insider activity is malicious then the alerts may provide evidence.

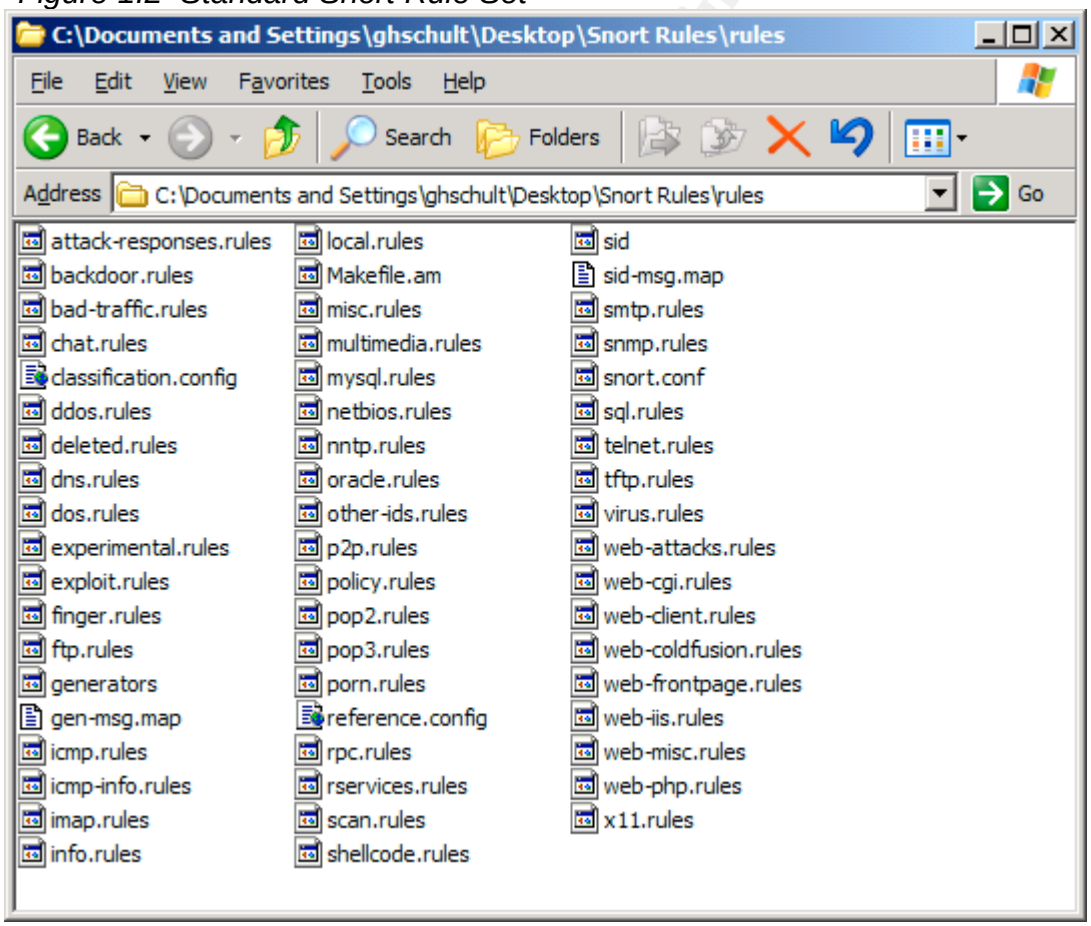
Standard Snort Rule Set

Use only the rule sets that are required to monitor your network. See the standard Snort rule set in Figure 1.2. Snort will test each available rule and end if

there is a match found. Running Snort with the –o switch set will read pass rules first. If a pass rule matches the packet then it will be discarded. Here is where writing precise rules becomes very important. You will only want to pass very specific traffic that has the potential to trigger an alert if it is common activity in your enterprise. Place the rules that you develop in the snort.conf file. This will allow centralized management of all locally developed rule sets.

One example of tuning that I ran into was a large amount of packets containing NOOP padding that comes from web servers. Most shellcode attacks contain NOOP padding as a method of ensuring the buffer is filled during buffer overflow attempts. NOOP padding is also found in normal web server traffic where binary files or image files are downloaded (Bassett, 2003). This is where a very specific pass rule is required as you will not want to pass shellcode attempts on systems in your environment.

Figure 1.2 Standard Snort Rule Set



Specific Rules Verses Generic Rules

When writing rules, be as specific as possible. It is always preferable to capture a packet and write the rule to trigger on as many unique attributes of the packet as possible. One example of a generic rule that masked the reconnaissance effort of the Nachi/Welchia worm in our environment is shown in figure 1.3. I found this rule in one system when I noticed that my second IDS had CyberKit 2.2 reconnaissance activity and the other did not.

Figure 1.3 ICMP Pass Rule

```
pass icmp any any -> $HOME_NET any (msg:"pass ping traffic";)
```

On our network, like most large enterprises, the administrative staff uses a series of administrative boxes to ping nodes that they manage to verify that the nodes are alive. The rule above was probably created at sometime to mask the ping activity that all the administrative nodes generated. The flaw in the pass rule above is that the rule passed all ping traffic. Using the recent example of the "Nachi" worm, which used a CyberKit 2.2 icmp packet padded with "a" to locate potential candidates to infect, passing all icmp traffic masked the reconnaissance activity. Poorly written pass rules can have a devastating effect by passing traffic that masks a potential exploit. The pass rule above has the effect of canceling all icmp rules in the Snort icmp.rules and icmp-info.rules rule set.

Here is the rule that alerted on Nachi reconnaissance activity. Snort alerted on the "ICMP ping CyberKit 2.2 Windows" rule shown in figure 1.4 during the "Nachi worm attack. The icmp packet used by Nachi contained the "a" padding shown in the content section of the Snort rule below.

Figure 1.4 CyberKit 2.2 Rule

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP
PING CyberKit 2.2 Windows";
content:"|aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa|";itype:8;depth:32;
reference:arachnids,154; sid:483;  classtype:misc-activity; rev:2;)
```

Set Up the Configuration File

Add variables to the snort.conf file as you are tuning your enterprise. Make variable names as descriptive as possible so when a rule is viewed there is no doubt what the variable represents. An example using the administrative nodes checking the health of systems on the network might be similar to the example in figure 1.5.

Figure 1.5 Variables

var ADMIN_NODES [192.168.1.100/32,192.168.1.120/32,192.168.1.125/32]
Rule: pass icmp $ADMIN_NODES any <> $HOME_NET any (msg:"Your message";)

Note the variables used in figure 1.4; HOME_NET and EXTERNAL_NET. These variables must be defined to tell Snort what IP addresses make up your network and the IP addresses that are considered anything outside your network. They are defined in the snort.conf file.

Data Roll Up

How will you view your data? There are many options depending on how you will use your system. Again a good place to look is on snort.org. They provide many options for viewing snort output. My IDS roll ups are done using SnortSnarf. While this might not be optimal, it provides a good overview for the period I wish to see. Consider rolling up detect data so that it can be correlated to the shifts that your facility operates. An example of this is as follows. If the production environment operates 7x24 on 12 hour shifts, then roll up the detect data every 12 hours. This will provide the best overview to correlate anomalous activity on individual shifts. Generally this will provide an avenue to view the support staff activity as they generally fall into the 8-5 day shift.

Figure 1.6 shows the SnortSnarf data roll up from the 18:00 to 6:00 shift. Comparing table 1.6 to table 1.7, there are no considerable differences for this particular day. One of the differences to point out is the RPC buffer overflow difference. In this case support personnel were bringing their laptops in without the current patch set. So, shift roll ups can provide interesting data.

Table 1.6
Earliest alert at **18:01:15**.304007 *on 08/23/2003*
Latest alert at **05:59:55**.194697 *on 08/24/2003*

Priority	Signature (click for sig info)	# Alerts	# Sources	# Dests	Detail link
0	DCE RPC Interface Buffer Overflow Exploit [sid] [BUGTRAQ]	22	1	20	Summary
0	TCP port 0 destination traffic [sid]	3	3	1	Summary
1	SHELLCODE x86 inc ebx NOOP [sid]	3	1	2	Summary
1	SHELLCODE x86 stealth NOOP [sid] [arachNIDS]	2	2	2	Summary
1	SHELLCODE x86 NOOP [sid]	1	1	1	Summary

2	NETBIOS SMB winreg access (unicode) [sid]	568	1	103	Summary
2	MISC Large UDP Packet [sid] [arachNIDS]	308	4	7	Summary
2	NETBIOS NT NULL session [sid]	121	3	3	Summary
2	ICMP Large ICMP Packet [sid] [arachNIDS]	24	1	1	Summary
2	ICMP L3retriever Ping [sid] [arachNIDS]	17	5	3	Summary
2	RPC portmap request NFS UDP [sid]	16	4	1	Summary
2	MISC xdmcp info query [sid]	14	8	2	Summary
2	NETBIOS SMB IPC$access [sid] [arachNIDS]	4	4	1	Summary
2	SHELLCODE x86 setuid 0 [sid] [arachNIDS]	1	1	1	Summary
3	ICMP Destination Unreachable (Undefined Code!) [sid]	250	6	9	Summary
3	BAD TRAFFIC bad frag bits [sid]	18	1	3	Summary
3	ICMP Destination Unreachable (Communication Administratively Prohibited) [sid]	14	4	3	Summary
3	TELNET access [sid]	1	1	1	Summary
N/A	(spp_stream4) possible EVASIVE RST detection	268	75	76	Summary
N/A	(spp_stream4) Multiple Acked Packets (possible fragroute)	13	5	7	Summary

Table 1.7
Earliest alert at **06:01:16**.974851 *on 08/24/2003*
Latest alert at **17:59:28**.550399 *on 08/24/2003*

Priority	Signature (click for sig info)	# Alerts	# Sources	# Dests	Detail link
0	DCE RPC Interface Buffer Overflow Exploit [sid] [BUGTRAQ]	54	3	44	Summary
1	FTP PASS overflow attempt [sid]	1	1	1	Summary
2	NETBIOS SMB winreg access (unicode) [sid]	568	1	103	Summary
2	MISC Large UDP Packet [sid]	470	4	7	Summary

	[arachNIDS]				
2	NETBIOS NT NULL session [sid]	124	4	4	Summary
2	ICMP Large ICMP Packet [sid] [arachNIDS]	26	1	1	Summary
2	RPC portmap request NFS UDP [sid]	24	6	3	Summary
2	MISC xdmcp info query [sid]	19	14	2	Summary
2	ICMP L3retriever Ping [sid] [arachNIDS]	6	1	1	Summary
2	NETBIOS SMB IPC$access [sid] [arachNIDS]	3	3	1	Summary
2	RPC mountd UDP export request [sid]	1	1	1	Summary
2	RPC mountd TCP export request [sid] [arachNIDS]	1	1	1	Summary
3	ICMP Destination Unreachable (Undefined Code!) [sid]	305	6	14	Summary
3	BAD TRAFFIC bad frag bits [sid]	38	2	5	Summary
3	ICMP Destination Unreachable (Communication Administratively Prohibited) [sid]	11	3	4	Summary
N/A	(spp_stream4) possible EVASIVE RST detection	217	64	67	Summary
N/A	(spp_stream4) Multiple Acked Packets (possible fragroute)	12	3	6	Summary

As pointed out earlier, you should be able to tell by viewing the alert roll up that this is an interoperable environment. You should readily pick out some attributes of both Windows and UNIX systems. In case you can not, some Windows attributes are RPC Buffer overflows, NetBios activity, and possibly the L3 pings. For UNIX note the NFS portmap request, xdmcp query, and Telnet. This is very important data as the analyst must know what their network traffic looks like.

Would you ever see telnet attempts in an all Windows environment? Should you see SMB traffic in a UNIX or Linux environment? Should you see packets with the DF and MF bits set in your environment? The answer to these questions is yes and no. This is where knowing your network is critical. What Operating Systems (OS) are running, what applications do they run, and what hardware is running. This just touches the surface, but this knowledge will help the analyst ask the right questions when it comes to whether the traffic is normal to the network.

So you have the data and hopefully you have more than one day of data. Where do you begin? Look at the priority column. You are probably receiving

pages from the priority ones. Evaluate and disposition all priority one activity. This way you will receive a page only when there is important activity worth knowing about immediately.

Second, look at the priority twos. If your environment is similar to mine, this is where much of the noise is intertwined with possible attack profiles. Notice my roll up logs; most alerts seem to be coming from the priority twos. Assuming that your low cost solution does not have infinite disk space and processor power, you should focus on this area next. Priority twos and portscan activity can be quite noisy, which can consume considerable disk space.

Finally you can use SnortSnarf to roll up some period of interest for trending purposes. The daily or hourly view will not show your trends in the system. As an example, my network sees a flurry of activity every Monday morning from a push done by the administrative staff. It kicks off three serious rules that I used to receive pages for. This or other activity can be seen in trending charts over a period of time.

Once your system is tuned, do not forget about it. This is where the fun begins. You should now be able to develop your CIP plan. Remember the IDS is just one element of your layered security model. Use the IDS data to make the entire model rock solid.

Conclusion

My attempt was to provide a single methodology on how to use and tune an open source IDS. I could find little information on how to tune and setup my IDS after I took ownership. While this is not an all inclusive operations manual its intent was to provide solutions and generate thoughts for the newer IDS owner. The paper discussed methods for determination of placing probes in the network. This determination and what the IDS will be used for will help the owner understand the rules required for effective monitoring. The paper looked at rule specificity and provided examples of how poorly written rules can be detrimental to the monitoring process. Finally the paper discussed a methodology for using SnortSnarf as a roll up utility for monitoring activity. I provided an overview of two roll up time frames to see immediate and trending activity in the monitored network. I hope this served an information purpose for someone trying to get their IDS running.

References for Section 1:

Bassett, Greg. “Intrusion Detection: An Inside Look”. 21 September 2003. URL: http://www.giac.org/practical/GCIA/Greg_Bassett_GCIA.pdf. (14 December 2003).

Jaques, Robert. “Most cyber-attacks will come from insiders”. 06 March 2003. URL: http://nl2.vnunet.com/News/1141354. (15 December 2003).

Intranetworks. “The Security Rules Have Changed: Are You Equipped to Play?”. URL: http://www.inkra.com/solutions/security.html. (15 December 2003).

Part 2: Network Detects

Detect 1: Reconnaissance

Source of Trace

The source of detect 1 was on my production network. The data was primarily collected on 08/18/03, which was the day that the Nachi worm was moving rapidly across the Internet.

Detect was Generated by:

The detect was captured by our Snort IDS running Snort version 1.9. Actual packets and Snort rules that were used to capture the packets are shown below. The icmp packet contains the “AA” padding as seen below in the reconnaissance ICMP packet. Snort alerted and logged the packet based on the snort icmp rule which triggers on the CyberKit 2.2 windows ping.

```
Snort – icmp.rules
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP
PING CyberKit 2.2 Windows";
content:"|aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa|";itype:8;depth:32;
reference:arachnids,154; sid:483;  classtype:misc-activity; rev:2;)
```

```
Reconnaissance ICMP packet
[**] ICMP PING CyberKit 2.2 Windows [**]
09/05-23:39:34.493934 68.6.98.49 -> 68.3.163.107
ICMP TTL:120 TOS:0x0 ID:17953 IpLen:20 DgmLen:92
Type:8   Code:0   ID:512   Seq:51362   ECHO
```

```
0x0000: 00 06 25 7F D8 D8 00 30 B8 04 2D 40 08 00 45 00  ..%.....0..-@..E
0x0010: 00 5C 46 21 00 00 78 01 6E 00 00 00 00 00 00 00  .\F!..x.n.D/b1D.
0x0020: 00 6B 08 00 D0 00 02 00 C0 A2 AA AA AA AA AA AA  ..k.............
0x0030: AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA  ................
0x0040: AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA  ................
0x0050: AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA  ................
0x0060: AA AA AA AA AA AA AA AA AA AA                    ..........
```

```
icmp packet used to find targets
[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:51.204914 192.168.83.132 -> 192.168.31.161
ICMP TTL:123 TOS:0x0 ID:3700 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:8997 ECHO
[Xref => arachnids 154]
```

Probability the Source Address was Spoofed

The addresses in the captures shown above and below are not spoofed as they are addresses on my network. The CyberKit 2.2 ping is a reconnaissance packet looking for live Windows hosts on the network. The packet capture shows an icmp packet, which is a type 8 request packet.

However, if the CyberKit 2.2 is to be used for malicious activity, the requesting address very well could be spoofed. Still if it were spoofed, the CyberKit utility is a reconnaissance tool and is looking for a specific response from a target set of tools. With that stated, there would have to be some mechanism for collecting the reconnaissance data. The description of the attack section will discuss how the tool works and the response it is attempting to elicit.

Description of the Attack

Nachi worm infected boxes ping nodes starting with a B class network or at a random start point. The Snort captures below show an infected node starting at 192.168.31.69 and iterating up one node per ping on subnet 31. Note that there is no target hit until host 160 is targeted. All nodes pinged until 160 are not Windows operating systems. The first Windows node on subnet 31 is host 160 at which point 192.168.83.132 attempts a buffer overflow to infect the host. All boxes on the production subnets were patched and not affected during the propagation of the Nachi/Welchia worm.

```
[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:49.767884 192.168.83.132 -> 192.168.31.69
ICMP TTL:123 TOS:0x0 ID:3548 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:50980 ECHO
[Xref => arachnids 154]

[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:49.783536 192.168.83.132 -> 192.168.31.70
ICMP TTL:123 TOS:0x0 ID:3550 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:51236 ECHO
[Xref => arachnids 154]

[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:49.798705 192.168.83.132 -> 192.168.31.71
ICMP TTL:123 TOS:0x0 ID:3552 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:51492 ECHO
[Xref => arachnids 154]
```

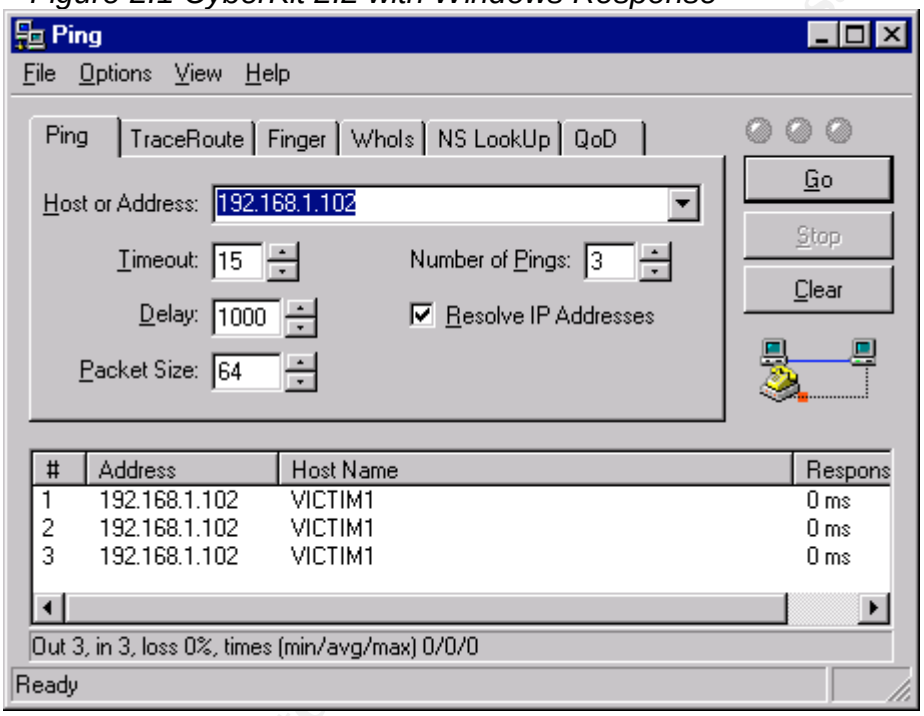
```
[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:49.814318 192.168.83.132 -> 192.168.31.72
ICMP TTL:123 TOS:0x0 ID:3554 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:51748 ECHO
[Xref => arachnids 154]

[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:49.845760 192.168.83.132 -> 192.168.31.74
ICMP TTL:123 TOS:0x0 ID:3557 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:52260 ECHO
[Xref => arachnids 154]

[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:49.861185 192.168.83.132 -> 192.168.31.75
ICMP TTL:123 TOS:0x0 ID:3559 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:52516 ECHO
[Xref => arachnids 154]

[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:49.877238 192.168.83.132 -> 192.168.31.76
ICMP TTL:123 TOS:0x0 ID:3561 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:52772 ECHO
[Xref => arachnids 154]

[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:49.923589 192.168.83.132 -> 192.168.31.79
ICMP TTL:123 TOS:0x0 ID:3565 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:53540 ECHO
[Xref => arachnids 154]

[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:50.018925 192.168.83.132 -> 192.168.31.85
ICMP TTL:123 TOS:0x0 ID:3574 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:55076 ECHO
[Xref => arachnids 154]

[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:50.033064 192.168.83.132 -> 192.168.31.86
ICMP TTL:123 TOS:0x0 ID:3576 IpLen:20 DgmLen:92
```

```
Type:8 Code:0 ID:512 Seq:55332 ECHO
[Xref => arachnids 154]

[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:50.048787 192.168.83.132 -> 192.168.31.87
ICMP TTL:123 TOS:0x0 ID:3578 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:55588 ECHO
[Xref => arachnids 154]

[**] [1:0:1] DCE RPC Interface Buffer Overflow Exploit [**]
[Priority: 0]
08/18-13:44:51.214403 192.168.83.132:2774 -> 192.168.31.161:135
TCP TTL:123 TOS:0x0 ID:3704 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x3A9846B9 Ack: 0x1871C075 Win: 0x4434 TcpLen: 20
[Xref => bugtraq 8205]

[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:51.189473 192.168.83.132 -> 192.168.31.160
ICMP TTL:123 TOS:0x0 ID:3692 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:8741 ECHO
[Xref => arachnids 154]

[**] [1:0:1] DCE RPC Interface Buffer Overflow Exploit [**]
[Priority: 0]
08/18-13:44:51.198954 192.168.83.132:2773 -> 192.168.31.160:135
TCP TTL:123 TOS:0x0 ID:3696 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x3A9791B7 Ack: 0x233EDD4 Win: 0x4434 TcpLen: 20
[Xref => bugtraq 8205]

[**] [1:483:2] ICMP PING CyberKit 2.2 Windows [**]
[Classification: Misc activity] [Priority: 3]
08/18-13:44:51.204914 192.168.83.132 -> 192.168.31.161
ICMP TTL:123 TOS:0x0 ID:3700 IpLen:20 DgmLen:92
Type:8 Code:0 ID:512 Seq:8997 ECHO
[Xref => arachnids 154]

[**] [1:0:1] DCE RPC Interface Buffer Overflow Exploit [**]
[Priority: 0]
08/18-13:44:51.214403 192.168.83.132:2774 -> 192.168.31.161:135
TCP TTL:123 TOS:0x0 ID:3704 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x3A9846B9 Ack: 0x1871C075 Win: 0x4434 TcpLen: 20
[Xref => bugtraq 8205]
```

Is the CyberKit 2.2 packet a stimulus or a response? The captures shown above are request pings, so it can be clearly stated that this activity is a stimulus. So, why were these packets seen at this time and is it malicious activity or data collection? First, what does the CyberKit 2.2 software do?

CyberKit 2.2 is a ping utility designed to elicit a response from a live Windows machine. This paper will only review the ping utility built into CyberKit 2.2.

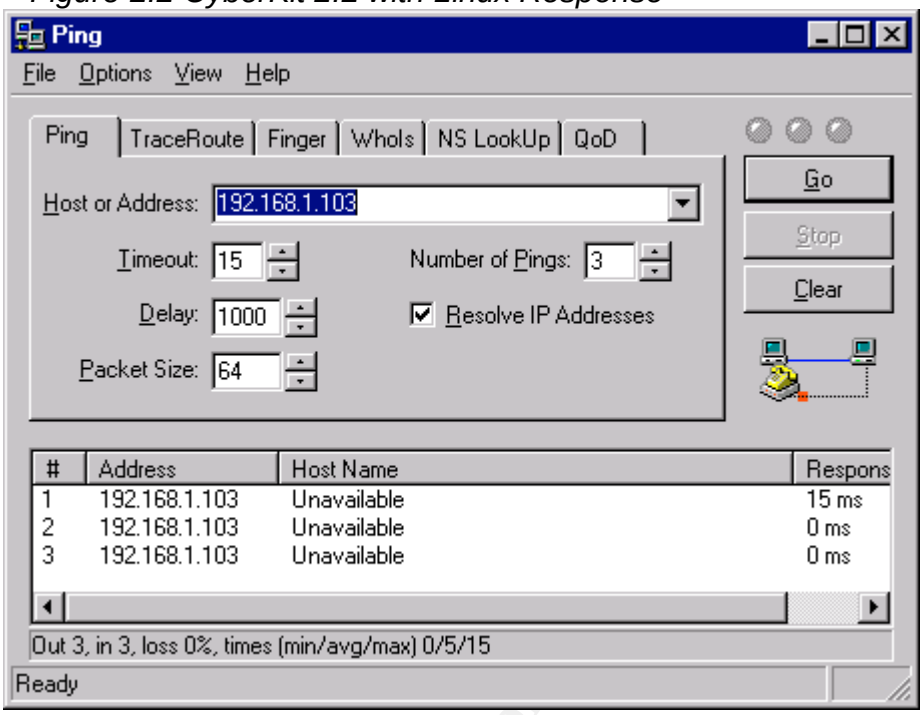
Figure 2.1 CyberKit 2.2 with Windows Response



Shown in Figure 2.1 is CyberKit 2.2 sending a series of pings to 192.168.1.102. A valid response is shown in the lower collection window with three replies including the hostname. Compare this response with Figure 2.2. Sending pings to a UNIX base host does not elicit the same response. In the response window, the host is shown as unavailable.

Back to why was this activity seen at this time? My network was being attacked with the Nachi/Welchia worm, so these were reconnaissance ping packets looking for live Windows hosts on the network. This is not to say that there are not other applications that could put the same or similar type packet on the wire. CyberKit like packets could be used for network mapping and other applications like “Big Brother”, which is a ping utility for tracking live nodes, are valid uses of this footprint.

Figure 2.2 CyberKit 2.2 with Linux Response



Attack Mechanism

By itself the CyberKit 2.2 ping is not an attack mechanism. As stated earlier there are valid reasons why this packet could be seen on the network. In this case, the packets captured for this analysis are clearly a reconnaissance effort for a pending attack.

With this being known activity of a pending attack, if enough nodes on a given network become infected, the fallout could overwhelm the network routers thus making this an attack method.

Correlations

There are many questions on the Internet to correlate the CyberKit 2.2 packet as reconnaissance activity for the Nachi/Welchia worm. Many packets were collected all over the world as this worm spread. One correlation is from the amount of packets that my home IDS captured. As the worm spread across the internet many nodes on my internet service provider became infected. My home IDS collected much of the reconnaissance activity from these infected systems.

Other sites across the world had similar activity. Two correlations are shown below to support the evidence.

Dr Medina indicates that the following snort capture was seen on their network on Sep 11 2003 (Medina, 2003).

> Sep 6 12:27:56 linuxserver snort: [1:483:2] ICMP PING CyberKit 2.2 Windows [Classification: Misc activity] [Priority: 3]: {ICMP} 200.95.132.194 -> 200.95.123.16

Blackburn on Monday, August 18, 2003 at 3:25 AM asks incidents.securityfocus.com if "this is the start of something naughty" with the capture seen below (Blackburn, 2003).

> Aug 18 10:46:14 thunder snort: [1:483:2] ICMP PING CyberKit 2.2 Windows
> [Classification: Misc activity] [Priority: 3]: {ICMP} 80.253.133.136 -> xx.xx.xx.120/123/125/127

There is considerable evidence that there was wide spread reconnaissance activity on the internet by doing a simple google search for CyberKit 2.2 activity.

Evidence of Active Targeting

This activity was not directed at any particular host. However, as shown in the description of attack section above, the CyberKit 2.2 ping can ping any node, but elicits the response it wants from a Windows node. For the Nachi/Welchia worm this is a critical response as Windows nodes are what the worm targets.

Severity

The values provided in the formula below are rated on a scale from 1 to 5 where 1 is the lowest and 5 is the highest.

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

Criticality = 2 due too the fact that the production environment requires a few Windows nodes to continue operations

Lethality = 1 due to the fact that this is reconnaissance activity

System countermeasures = 4 because 100% of production nodes were patched but 3 nodes became infected for some reason. Production nodes in most cases have full redundancy. An IDS was in place and able to detect activity.

Network countermeasures = 2 since the firewall was unloaded, but ACLs to stop echo replies were applied to production routers

Severity = (2 + 1) – (4 + 2)
Severity = -3

Defensive Recommendation

The best defense for this type of reconnaissance activity is to block icmp packets greater than 90 bytes at the firewall or boundary router. The boundary router ACL could block either ingress or egress response or reply packets. Before doing this, perform an icmp capture to make sure that valid traffic can still get through if there is a business requirement. Second, if this is valid reconnaissance activity, it would be ideal to be patched to the current patch set if it is available. Patching may not always be an option, but in the case of Nachi/Welchia the patch was out prior to the attack.

Multiple Choice Test Question

What does the CyberKit 2.2 application return if it attempts a hostname lookup on a Linux node?

a. a valid hostname
b. unavailable
c. icmp host unreachable packet
d. none of the above

Answer b

References for detect 1:

Blackburn, Charles. CyberKit 2.2 activity. 18 August 2003. URL: http://www.securityfocus.com/archive/75/333858/2003-08-14/2003-08-20/0.html (04 October 2003).

Medina, Aldo. CyberKit 2.2 activity. 11 September 2003. URL: http://lists.insecure.org/lists/security-basics/2003/Sep/0576.html (04 October 2003).

Detect 2: Buffer Overflow Attempt on Target

Source of Trace

The source of this detect was my production network. The detect was captured using Snort version 2.0.

```
Shellcode.rules
alert ip $EXTERNAL_NET any -> $HOME_NET $SHELLCODE
(msg:"SHELLCODE x86 NOOP"; content: "|90 90 90 90 90 90 90 90 90
90 90 90 90 90|"; depth: 128; reference:arachnids,181;
classtype:shellcode-detect; sid:648;  rev:5;)

18:12:26.137980 10.1.xx.xx.1022 > 192.168.xx.xx.3610: P [tcp sum ok]
1634227749:1634229209(1460) ack 1847795 win 17520 (DF) (ttl 53, id
29464, len 1500)
```

```
0x0000   4500 05dc 7318 4000 0006 2000 0002 0009        E...s.@.5./.....
0x0010   0010 000e 03fe 0e1a 6168 5625 001c 31f3        ..vn....ahV%..1.
0x0020   5018 4470 0000 0000 830e 04a8 8074 0383        P.Dpj;.......t..
0x0030   0e08 a801 7403 830e 10a8 0474 0383 0e20        ....t......t....
0x0040   f6c4 0874 078b 0e80 c980 890e f6c4 1074        ...t...........t
0x0050   078b 0680 cc02 8906 5fb8 0100 0000 5ec2        ........_.....^.
0x0060   0800 8bc7 5f5e c208 0090 9090 8b44 2404        ...._^.......D$.
0x0070   5650 33f6 ff15 0cf1 4200 83f8 ff74 09a8        VP3.....B....t..
0x0080   10b8 0100 0000 7502 8bc6 5ec2 0400 9090        ......u...^.....
0x0090   9090 9090 9090 9090 9090 9090 566a 0c6a        ............Vj.j
0x00a0   4233 f6ff 1528 f142 0050 ff15 2cf1 4200        B3...(.B.P..,.B.
0x00b0   85c0 740f 8b4c 2408 5e89 01b8 0100 0000        ..t..L$.^.......
0x00c0   c204 008b c65e c204 0090 9090 568b 7424        .....^......V.t$
0x00d0   0885 f674 2a8b 0685 c074 2457 8b3d 30f1        ...t*....t$W.=0.
0x00e0   4200 50ff d750 ff15 34f1 4200 8b06 50ff        B.P..P..4.B...P.
0x00f0   d750 ff15 38f1 4200 c706 0000 0000 5f5e        .P..8.B......._^
0x0100   c204 0090 9090 9090 9090 9090 8b4c 2404        .............L$.
0x0110   33c0 85c9 7435 8b54 2408 85d2 742d 8b41        3...t5.T$...t-.A
0x0120   04c7 4204 0000 0000 8902 8b41 0485 c074        ..B........A...t
0x0130   0389 5004 8b01 8951 0485 c075 0289 118b        ..P....Q...u....
0x0140   4108 4089 4108 b801 0000 00c2 0800 9090        A.@.A...........
0x0150   9090 9090 9090 9090 9090 9090 8b4c 2404        .............L$.
0x0160   33c0 85c9 742e 8b51 0885 d274 278b 018b        3...t..Q...t'...
0x0170   5004 85d2 8911 7406 c702 0000 0000 8b51        P.....t........Q
0x0180   084a 8951 0875 0dc7 4104 0000 0000 c701        .J.Q.u..A.......
0x0190   0000 0000 c204 0090 9090 9090 8b4c 2404        .............L$.
0x01a0   33c0 85c9 7403 8b41 08c2 0400 8b54 2404        3...t..A.....T$.
0x01b0   8b4c 2408 5633 c08b 7208 3bce 5e73 0c8b        .L$.V3..r.;.^s..
0x01c0   0285 c974 068b 4004 4975 fac2 0800 9090        ...t..@.Iu......
0x01d0   9090 9090 9090 9090 9090 9090 8b54 2404        .............T$.
0x01e0   33c0 568b 4a08 85c9 745b 8b74 240c 3bf1        3.V.J...t[.t$.;.
0x01f0   7353 8b02 85f6 750d 8b48 0485 c989 0a74        sS....u..H.....t
0x0200   1789 31eb 1349 3bf1 7508 8b4a 048b 0989        ..1..I;.u..J....
0x0210   4a04 8b40 044e 75fa 8b4a 0849 894a 0875        J..@.Nu..J.I.J.u
0x0220   0dc7 4204 0000 0000 c702 0000 0000 8b08        ..B.............
0x0230   85c9 7406 8b50 0489 5104 8b48 0485 c974        ..t..P..Q..H...t
0x0240   048b 1089 115e c208 0090 9090 83ec 1053        .....^.........S
0x0250   5657 33db 680c 0200 006a 4033 ff89 5c24        VW3.h....j@3..\$
0x0260   1cc7 4424 1401 0000 0089 5c24 18ff 1528        ..D$......\$...(
0x0270   f142 0050 ff15 2cf1 4200 8bf0 3bf3 0f84        .B.P..,.B...;...
0x0280   7402 0000 8b44 2420 5568 0002 0000 5056        t....D$.Uh....PV
```

```
0x0290   e8c7 2700 008b fe83 c9ff 33c0 83c4 0cf2        ..'.......3.....
0x02a0   aef7 d149 8beb 898e 0002 0000 0f84 3a02        ...I..........:.
0x02b0   0000 8bfe b901 0000 0089 7c24 1c89 4c24        ..........|$..L$
0x02c0   248a 078a 1598 2a43 003a c20f 85ba 0000        $.....*C.:......
0x02d0   0039 5c24 180f 85f4 0100 008b 4424 24c6        .9\$........D$$.
0x02e0   0700 8b8e 0002 0000 8bd8 3bc1 0f83 8500        ..........;.....
0x02f0   0000 8a0d 982a 4300 380c 3375 0b8b 9600        .....*C.8.3u....
0x0300   0200 0043 3bda 72f0 3bd8 766b 3b9e 0002        ...C;.r.;.vk;...
0x0310   0000 7358 8d14 3383 c9ff 8bfa 33c0 f2ae        ..sX..3.....3...
0x0320   f7d1 4951 8d4c 2e01 5251 e89d 5400 008b        ..IQ.L..RQ..T...
0x0330   8600 0200 008b d52b d383 c40c 428b 7c24        .......+....B.|$
0x0340   1c03 c2c7 4424 1001 0000 0089 8600 0200        ....D$..........
0x0350   0033 dbc6 0430 008b 8600 0200 00b9 0100        .3...0..........
0x0360   0000 c644 3001 00e9 6301 0000 89ae 0002        ...D0...c.......
0x0370   0000 c644 2e01 00c7 4424 1001 0000 0033        ...D....D$.....3
0x0380   dbb9 0100 0000 e944 0100 003a 0599 2a43        .......D...:..*C
0x0390   000f 85f8 0000 0039 4c24 1875 778b 8600        .......9L$.uw...
0x03a0   0200 008b 5424 243b d077 698a 1598 2a43        ....T$$;.wi...*C
0x03b0   008d 4c2e 0138 118b 5424 1474 2383 fa01        ..L..8..T$.t#...
0x03c0   0f85 b300 0000 3944 2424 731d c607 00c7        ......9D$$s.....
0x03d0   4424 1001 0000 0089 5c24 14e9 9900 0000        D$......\$......
0x03e0   83fa 010f 8590 0000 002b c550 5157 e8d9        .........+.PQW..
0x03f0   5300 008b 8600 0200 008b 4c24 3083 c40c        S.........L$0...
0x0400   484d 4989 8600 0200 0089 4c24 244f 895c        HMI.......L$$O.\
0x0410   2414 eb65 394c 2410 7511 8b86 0802 0000        $..e9L$.u.......
0x0420   895c 2410 4089 8608 0200 008d 5c2e 0153        .\$.@.......\..S
0x0430   e8d7 0000 0083 c404 85c0 753b 8b44 2418        ..........u;.D$.
0x0440   85c0 7533 8b86 0002 0000 2bc5 5053 57e8        ..u3......+.PSW.
0x0450   7853 0000 8b9e 0002 0000 8b4c 2430 83c4        xS.........L$0..
0x0460   0c4b 4d49 c744 2414 0100 0000 899e 0002        .KMI.D$.........
0x0470   0000 894c 2424 4f33 db8b 4424 1833 c93b        ...L$$O3..D$.3.;
0x0480   c30f 94c1 894c 2418 b901 0000 00eb 403a        .....L$.......@:
0x0490   059a 2a43 0074 213a 059b 2a43 0074 1939        ..*C.t!:..*C.t.9
0x04a0   4c24 1075 2a8b 8608 0200 0089 5c24 1040        L$.u*.......\$.@
0x04b0   8986 0802 0000 eb17 394c 2410 7511 8b86        ........9L$.u...
0x04c0   0402 0000 895c 2410 4089 8604 0200 008b        .....\$.@.......
0x04d0   4424 2445 4047 8944 2424 8b86 0002 0000        D$$E@G.D$$......
0x04e0   3be8 897c 241c 0f82 d5fd ffff 5d8b c65f        ;..|$.......]..._
0x04f0   5e5b 83c4 10c2 0400 8bc7 5f5e 5b83 c410        ^[........_^[...
0x0500   c204 0090 9090 9090 9090 9090 8b44 2404        .............D$.
0x0510   85c0 7501 c38a 008a 0d9a 2a43 003a c174        ..u.......*C.:.t
0x0520   0b3a 059b 2a43 0074 0333 c0c3 b801 0000        .:..*C.t.3......
0x0530   00c3 9090 9090 9090 9090 9090 8b44 2404        .............D$.
0x0540   8b80 0402 0000 c204 0090 9090 5355 568b        ............SUV.
0x0550   7424 1033 ed57 803e 0074 418b 5c24 188a        t$.3.W.>.tA.\$..
0x0560   068a 0d9a 2a43 003a c174 083a 059b 2a43        ....*C.:.t.:..*C
0x0570   0075 118d 4601 5350 e8ef 2700 0083 c408        .u..F.SP..'.....
0x0580   85c0 7421 8bfe 83c9 ff33 c0f2 aef7 d149        ..t!.....3.....I
0x0590   8a44 0e01 8d74 0e01 84c0 75c3 5f8b c55e        .D...t....u._..^
0x05a0   5d5b c208 005f 5e5d b801 0000 005b c208        ][..._^].....[..
0x05b0   0090 9090 9090 9090 9090 9090 568b 7424        ............V.t$
0x05c0   0885 f674 1c57 8b3d 30f1 4200 56ff d750        ...t.W.=0.B.V..P
0x05d0   ff15 34f1 4200 56ff d750 ff15                  ..4.B.V..P..
```

Detect was Generated by:

This detect was an alert to a shellcode buffer overflow rule found in the standard Snort rule set. The detect was generated by Snort version 2.0 with a standard rule with some modifications to the local rules. See rule shown below.

```
Shellcode.rules
alert ip $EXTERNAL_NET any -> $HOME_NET $SHELLCODE
(msg:"SHELLCODE x86 NOOP"; content: "|90 90 90 90 90 90 90 90 90
90 90 90 90|"; depth: 128; reference:arachnids,181;
classtype:shellcode-detect; sid:648;  rev:5;)
```

The alert came from the rule looking for the “90 90 90 90 90” NOOP padding that is high-lighted in the packet capture above.

Probability the Source Address was Spoofed

There was no chance the source address was spoofed for this capture. In this case the packet came from my local network and the source address was from a local proxy server.

Description of the Attack

The packets captured for this detect and analysis was sent in response to a web query on a web server or in this case through a proxy server. The reason that I chose this packet for analysis is the fact that it alerts as if it is a shellcode attack. Shellcode attacks by default in Snort version 2.0 are priority 1, which sends a page to the IDS analyst. Other web traffic or actual shellcode attacks can send similar alerts and all need analysis for understanding. See the different packet subsets shown below. The shellcode attack is outlined in the attack mechanism section.

Actual Attack (Welchia/Nachi) - Packet subset shown for brevity

```
0x00F0: 00 00 00 00 00 00 01 10 08 00 CC CC CC CC C8 00  ................
0x0100: 00 00 4D 45 4F 57 E8 05 00 00 D8 00 00 00 00 00  ..MEOW..........
0x0110: 00 00 02 00 00 00 07 00 00 00 00 00 00 00 00 00  ................
0x0120: 00 00 00 00 00 00 00 00 00 00 C4 28 CD 00 64 29  ...........(..d)
0x0130: CD 00 00 00 00 00 07 00 00 00 B9 01 00 00 00 00  ................
0x0140: 00 00 C0 00 00 00 00 00 00 00 46 AB 01 00 00 00  ..........F.....
0x0150: 00 00 C0 00 00 00 00 00 00 00 46 A5 01 00 00 00  ..........F.....
0x0160: 00 00 C0 00 00 00 00 00 00 00 46 A6 01 00 00 00  ..........F.....
0x0170: 00 00 C0 00 00 00 00 00 00 00 46 A4 01 00 00 00  ..........F.....
0x0180: 00 00 C0 00 00 00 00 00 00 00 46 AD 01 00 00 00  ..........F.....
0x0190: 00 00 C0 00 00 00 00 00 00 00 46 AA 01 00 00 00  ..........F.....
0x01A0: 00 00 C0 00 00 00 00 00 00 00 46 07 00 00 00 60 00  ..........F....`.
0x01B0: 00 00 58 00 00 00 00 90 00 00 00 00 40 00 00 00 20 00  ..X.........@... .
0x0360: 00 00 30 00 2E 00 00 00 00 00 00 00 00 00 00 00 00 00  ..0.............
0x0370: 00 00 00 00 00 00 01 10 08 00 CC CC CC CC 68 00  ..............h.
0x0380: 00 00 0E 00 FF FF 68 8B 0B 00 02 00 00 00 00 00  ......h.........
0x0390: 00 00 00 00 00 00 86 01 00 00 00 00 00 00 86 01  ................
0x03A0: 00 00 5C 00 5C 00 46 00 58 00 4E 00 42 00 46 00  ..\.\.F.X.N.B.F.
0x03B0: 58 00 46 00 58 00 4E 00 42 00 46 00 58 00 46 00  X.F.X.N.B.F.X.F.
0x03C0: 58 00 46 00 58 00 46 00 58 00 9D 13 00 01 CC E0  X.F.X.F.X.......
0x03D0: FD 7F CC E0 FD 7F 90 90 90 90 90 90 90 90 90 90  ................
0x03E0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x03F0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0400: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0410: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0420: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0430: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
```

```
0x0440: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0450: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0460: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0470: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0480: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x0490: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x04A0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x04B0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x04C0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x04D0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x04E0: 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
0x04F0: 90 90 90 90 90 90 90 90 90 90 90 90 EB 10 5A 4A  ..............ZJ
0x0500: 33 C9 66 B9 76 01 80 34 0A 99 E2 FA EB 05 E8 EB  3.f.v..4........
0x0510: FF FF FF 70 61 99 99 99 C3 21 95 69 64 E6 12 99  ...pa....!.id...
0x0520: 12 E9 85 34 12 D9 91 12 41 12 EA A5 9A 6A 12 EF  ...4....A....j..
0x0530: E1 9A 6A 12 E7 B9 9A 62 12 D7 8D AA 74 CF CE C8  ..j....b....t...
0x0540: 12 A6 9A 62 12 6B F3 97 C0 6A 3F ED 91 C0 C6 1A  ...b.k...j?.....
0x0550: 5E 9D DC 7B 70 C0 C6 C7 12 54 12 DF BD 9A 5A 48  ^..{p....T....ZH
```

Web traffic showing the "43 43 43 43" x86 padding - Packet subset shown for brevity

```
0x0020   5018 4470 08ce 0000 4854 5450 2f31 2e30        P.Dp....HTTP/1.0
0x0030   2032 3030 204f 4b0d 0a44 6174 653a 2054        .200.OK..Date:.T
0x00c0   0d0a 4163 6365 7074 2d52 616e 6765 733a        ..Accept-Ranges:
0x00d0   2062 7974 6573 0d0a 436f 6e74 656e 742d        .bytes..Content-
0x00e0   4c65 6e67 7468 3a20 3130 3233 0d0a 436f        Length:.1023..Co
0x00f0   6e74 656e 742d 5479 7065 3a20 696d 6167        ntent-Type:.imag
0x0100   652f 6a70 6567 0d0a 582d 4361 6368 653a        e/jpeg..X-Cache:
0x0110   204d 4953 5320 6672 6f6d 2070 726f 7879        .MISS.from.proxy
0x01c0   0013 0f0f 1711 1725 1616 252f 241d 242f        .......%..%/$.$/
0x01d0   2c24 2323 242c 3a32 3232 3232 3a43 3d3d        ,$##$,:22222:C==
0x01e0   3d3d 3d3d 4343 4343 4343 4343 4343 4343        ====CCCCCCCCCCCC
0x01f0   4343 4343 4343 4343 4343 4343 4343 4343        CCCCCCCCCCCCCCCC
0x0200   4301 1417 171e 1a1e 2418 1824 3324 1e24        C.......$..$3$.$
0x0210   3342 3329 2933 4243 423e 323e 4243 4343        3B3))3BCB>2>BCCC
0x0220   4343 4343 4343 4343 4343 4343 4343 4343        CCCCCCCCCCCCCCCC
0x0230   4343 4343 4343 4343 4343 4343 4343 4343        CCCCCCCCCCCCCCCC
0x0240   4343 ffc0 0011 0800 3200 3203 0122 0002        CC......2.2..".."
0x0250   1101 0311 01ff c400 8a00 0002 0301 0101        ................
```

Web traffic showing the "61 61 61 61" x86 padding - Packet subset shown for brevity

```
0x0020   5010 4000 b6d6 0000 4854 5450 2f31 2e31        P.@.....HTTP/1.1
0x0030   2032 3030 204f 4b0d 0a41 6765 3a20 3230        .200.OK..Age:.20
0x0040   3039 3532 360d 0a41 6363 6570 742d 5261        09526..Accept-Ra
0x0050   6e67 6573 3a20 6279 7465 730d 0a44 6174        nges:.bytes..Dat
0x00e0   6e22 0d0a 5365 7276 6572 3a20 4d69 6372        n"..Server:.Micr
0x00f0   6f73 6f66 742d 4949 532f 352e 300d 0a78        osoft-IIS/5.0..x
0x0100   2d69 6973 3562 6173 6562 7569 6c64 2d69        -iis5basebuild-i
0x0110   6465 6e74 6966 6965 723a 2049 5745 2050        dentifier:.IWE.P
0x0120   3130 3020 4949 5320 352e 3020 4261 7365        100.IIS.5.0.Base
0x01d0   1110 1318 2719 1816 1618 3022 241c 2739        ....'.....0"$.'9
0x01e0   323c 3b38 3237 363f 475a 4c3f 4355 4436        2<;8276?GZL?CUD6
0x01f0   374e 6b4f 555d 6065 6665 3d4b 6f77 6e62        7NkOU]`efe=Kownb
0x0200   765a 6365 61ff db00 4301 1112 1218 1518        vZcea...C.......
0x0210   2e19 192e 6141 3741 6161 6161 6161 6161        ....aA7Aaaaaaaaa
0x0220   6161 6161 6161 6161 6161 6161 6161 6161        aaaaaaaaaaaaaaaa
0x0230   6161 6161 6161 6161 6161 6161 6161 6161        aaaaaaaaaaaaaaaa
0x0240   6161 6161 6161 6161 6161 ffc0 0011 0800        aaaaaaaaaa......
0x0250   7c00 7803 0122 0002 1101 0311 01ff c400        |.x.."..........
0x0260   1b00 0002 0301 0101 0000 0000 0000 0000        ................
0x0270   0000 0405 0003 0602 0107 ffc4 0037 1000        .............7..
0x0280   0201 0302 0403 0605 0304 0300 0000 0001        ................
```

All of the packets shown above alert the IDS analyst that there is a shellcode attack happening. They are, with the exception of the Nachi/Welchia packet, normal traffic that can be seen in any environment where web traffic is considered normal. The point I want to make is that the “crying wolf syndrome” can easily make real threats go by without proper analysis.

Attack Mechanism

The DCOM RPC attack works once a response is returned from a valid Windows based target. A data packet containing shellcode is then sent to udp port 135, which exploits the DCOM RPC process or tcp port 80 on a node running IIS 5.0, which exploits the WebDav process. The victim then contacts the attack system on port range 666 to 765 to receive instruction to start the tftp process. The attack system then transfers svchost.exe and dllhost.exe to the \%systemroot%\wins directory as in the case of Welchia/Nachi (Symantec Security Response, p.2-3).

The DCOM RPC exploit works due to the way that RPC handles malformed tcp packets. The vulnerability affects an RCP interface, which handles DCOM object activation requests sent from the client to server. The malformed packet is generally sent to port 135, 139, 445, or any port configure as an RPC port. If the exploit is successful, local system privileges will be gained allowing the attacker the ability to perform activities granted to the privilege set (Aharoni, p.1).

The WebDav/ntdll.dll vulnerability is a buffer overflow that exists on Windows boxes with IIS installed. The IIS application does not necessarily need to be running to be susceptible to this exploit. The attack is successful if long data is sent to IIS, which passes the data to ntdll.dll. The ntdll.dll system component does not perform bounds checking on the data, which can allow a buffer overrun. If the exploit is successful, the attacker may gain default local system privileges (Symantec Security Response, p.1).

The purpose of the padding shown in the exploit packet is an attempt to overflow the buffer of some vulnerable code. During a buffer overflow attempt, the attacker wants to fill the allocated buffer space with NOOP’s to a point where the instructions they want to run are the next thing the program sees. So the purpose of the NOOP padding is to fill the buffer or to go slightly past the last allocated point (Hinckley, 2003). Buffer overflow attacks are not specific to Windows operating systems. Any program that does not provide bounds checking on the data passed to it is susceptible.

Correlations

Bassett discusses details about web traffic, NOOP padding, and some causes in his practical (Bassett, 2003).

The Nahci exploit packet is a variant of the RPC DCOM buffer overflow attack on the Windows OS. The worm is patched by applying Microsoft patch MS03-026. The Common Vulnerabilities and Exposures number for the latest shellcode exploits is CAN-2003-0352, which covers Blaster/Lovesan and Nachi/Welchia (CVE).

Evidence of Active Targeting

The RPC DCOM attack looks for target victims in a somewhat random pattern based on the attack nodes IP address and the algorithm used by the attack. For the NOOP packet, which is a false positive featured in this detect, there is no active targeting as the packet is a response to a query.

Severity

Below is a summation of severity for the web traffic containing X86 NOOP padding seen in my environment. The values provided in the formula below are rated on a scale from 1 to 5 where 1 is the lowest and 5 is the highest.

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

Criticality = 1

Lethality = 1

System countermeasures = 2 because one counter measure is the IDS and poor tuning causes alarms to analysts

Network countermeasures = 1

Severity = (1 + 1) – (2 + 1)
Severity = -1

The real problem with these NOOP packets is that improper IDS tuning can make the IDS analyst numb to real attacks. Also, care must be taken when writing pass rules for shellcode false positives, so real attacks do not pass along with the web traffic.

Defensive Recommendation

The best defensive method is to carefully examine the packets and write pass rules. The pass rules should be as specific as possible so real shellcode NOOP attacks are not masked.

Multiple Choice Test Question

Only Microsoft software is susceptible to buffer overflow attacks?

a. True
b. False

b. False – any program that does not do bounds checking on data passed to the buffer is susceptible

References for detect 2:

Aharoni, Mati. “Window DCOM RPC Exploit.” 14 August 2003. URL: http://sysadminnews.com/sysadminnews-32-20030814WindowsDCOMRPCExploit.html (01 September 2003).

Bassett, Greg. “Intrusion Detection: An Inside Look”. 21 September 2003. URL: http://www.giac.org/practical/GCIA/Greg_Bassett_GCIA.pdf. (14 December 2003).

Common Vulnerabilities and Exposures. “CAN-2003-0352 (under review).” URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name-CAN-2003-0352 (12 December 2003).

Hinckley, Kee. “Definition: Buffer Overflow”. (14 September 2003). URL: http://commons.somewhere.com/buzz/2000/Definition.Buffer.Overfl.html (16 December 2003).

Symantec Security Response. “Microsoft Windows 2000 WebDAV/ntdll.dll Buffer Overflow Vulnerability.” 17 March 2003. URL: http://securityresponce.symantec.com/avcenter/security/Content/3.17.2003.html (01 September 2003).

Symantec Security Response. “W32.Welchia.Worm.” 29 August 2003. URL: http://securityresponce.symantec.com/avcenter/venc/data/w32.welchia.worm.html (01 September 2003).

Detect 3: Looking For A Way To Hide

Source of Trace

This trace came from the incidents.org/logs/raw detect file. The primary file used for this detect was 2002.10.3. Based on the Ethernet header, the packets

contain a destination MAC of 00:00:0c:04:b2:33 and a source MAC of 00:03:e3:d9:26:c0, which places the capture tool or IDS between two Cisco routers. Best guess is that the destination router faces the WAN and/or the Internet and the source router is the first layer LAN distribution of some environment. Below is a diagram of the router relationship to the IDS probe.

```
                Router 1              IDS              Router 2
Internet Facing  X----------------------o--------------------X  → Internal Network
```

Detect was Generated by:

The snort rule that was triggered by the port 1080 scan is shown below. This rule is from the Snort version 2.0 stable rule set.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 1080 (msg:"SCAN
SOCKS Proxy attempt"; flags:S,12;
reference:url,help.undernet.org/proxyscan/; classtype:attempted-recon;
sid:615; rev:4;)
```

Nine packets from the capture are shown below. Ethereal was used to view and analyze the packet capture.

```
0.000000   216.77.219.195   >   207.166.233.11      TCP     48839 >
1080 [SYN] Seq=1769720505 Ack=1769720505 Win=1024 Len=0

0000  00 00 0c 04 b2 33 00 03 e3 d9 26 c0 08 00 45 00   .....3....&...E.
0010  00 28 5d 00 00 00 31 06 09 59 d8 4d db c3 cf a6   .(]...1..Y.M....
0020  e9 0b be c7 04 38 69 7b ca b9 69 7b ca b9 50 02   .....8i{..i{..P.
0030  04 00 5d 01 00 00 00 00 00 00 00 00                ..]........

66.390000   216.77.219.195   >   207.166.50.15      TCP     63990 >
1080 [SYN] Seq=897349751 Ack=897349751 Win=1024 Len=0

0000  00 00 0c 04 b2 33 00 03 e3 d9 26 c0 08 00 45 00   .....3....&...E.
0010  00 28 7f 60 00 00 31 06 9d f5 d8 4d db c3 cf a6   .(.`..1....M....
0020  32 0f f9 f6 04 38 35 7c 78 77 35 7c 78 77 50 02   2....85|xw5|xwP.
0030  04 00 e5 51 00 00 00 00 00 00 00 00                ...Q........

132.760000  216.77.219.195   >   207.166.16.247     TCP     13700 >
1080 [SYN] Seq=1536807 Ack=1536807 Win=1024 Len=0

0000  00 00 0c 04 b2 33 00 03 e3 d9 26 c0 08 00 45 00   .....3....&...E.
0010  00 28 48 31 00 00 31 06 f9 39 d8 4d db c3 cf a6   .(H1..1..9.M....
0020  10 f7 35 84 04 38 00 17 73 27 00 17 73 27 50 02   ..5..8..s'..s'P.
0030  04 00 43 44 00 00 00 00 00 00 00 00                ..CD........
```

```
199.150000  216.77.219.195   >  207.166.133.140      TCP      6469 >
1080 [SYN] Seq=1067936866 Ack=1067936866 Win=1024 Len=0

0000  00 00 0c 04 b2 33 00 03 e3 d9 26 c0 08 00 45 00   .....3....&...E.
0010  00 28 62 f6 00 00 31 06 67 e0 d8 4d db c3 cf a6   .(b...1.g..M....
0020  85 8c 19 45 04 38 3f a7 6c 62 3f a7 6c 62 50 02   ...E.8?.lb?.lbP.
0030  04 00 77 58 00 00 00 00 00 00 00 00                ..wX........

265.560000  216.77.219.195   >  207.166.26.20       TCP      62881 >
1080 [SYN] Seq=131862696 Ack=131862696 Win=1024 Len=0

0000  00 00 0c 04 b2 33 00 03 e3 d9 26 c0 08 00 45 00   .....3....&...E.
0010  00 28 02 01 00 00 31 06 35 4f d8 4d db c3 cf a6   .(....1.5O.M....
0020  1a 14 f5 a1 04 38 07 dc 10 a8 07 dc 10 a8 50 02   .....8........P.
0030  04 00 2e 80 00 00 00 00 00 00 00 00                ............

331.940000  216.77.219.195   >  207.166.213.46      TCP      47201 >
1080 [SYN] Seq=883477552 Ack=883477552 Win=1024 Len=0

398.310000  216.77.219.195   >  207.166.219.166      TCP      39364 >
1080 [SYN] Seq=800273912 Ack=800273912 Win=1024 Len=0

464.640000  216.77.219.195   >  207.166.207.78       TCP      3625 >
1080 [SYN] Seq=1930540870 Ack=1930540870 Win=1024 Len=0

531.040000  216.77.219.195   >  207.166.111.25       TCP      38286 >
1080 [SYN] Seq=1595556944 Ack=1595556944 Win=1024 Len=0
```

Probability the Source Address was Spoofed

It is difficult to say definitively that the source address is or is not spoofed. There are a few possibilities that will be discussed about the fact that the source address could be spoofed.

The address may be spoofed, but everything in the packet looks normal on the surface with the exception of the incorrect IP and TCP checksums, which have been changed to protect the potentially innocent. To dig a little deeper into the packet using passive operating system fingerprinting attributes, there does seem to be a discrepancy.

Using the following attributes, let's examine the sending OS.

- Packet TTL
  - All packets in the capture have a TTL of 49
- Window size
  - All packets in the capture have a Window size of 1024
- Defrag bit set

    - None of the packets have the DF flag set
- Type of service
    - The TOS is 0x00 for all packets

What does this say about the OS of the sending box? Using the OS fingerprinting chart created by the honeynet project, a comparison of the above data will be made.

The TTL of 49 would indicate that the packets originated at a node that begins its TTL with 64. The rational for this is that 64 – 49 = 15 and 15 hops from source to destination is a very acceptable hop count. This is difficult to test since a traceroute can not be sent to verify the hop count. The window size is the most troubling piece of data in the packet as the data collected during the honeynet project contains no OSs with window sizes less than 2100. The packets do not have the DF flag set and the TOS flags are 0x00. With this data in hand there is no clear OS that stands out. This leads me to believe that some sort of packet crafting is going on with the packets. One final check was to perform with nslookup and a subsequent whois on the address, which returned an owner of BellSouth.

An anomaly that lends support to packet crafting can be found with the SYN and ACK sequence numbers. Reviewing the packets shown above, the SYN and ACK sequence numbers are the same. Since most TCP stacks choose beginning sequence values randomly it is highly unlikely that the values could be the same without packet crafting. This is particularly true for successive connections as seen above. When the packets were examined closer, the ACK flag was not set and hence should not have a value.

The question of rather this packet is a stimulus or a response must be examined to make a more educated guess. A stimulus would indicate that the originator would hope to gain something in return for their effort. Clearly this scan is indicative of a valid scan of a port that could be useful for unscrupulous activity. This type of scan could be either a reconnaissance scan to find an exploitable port or an attempt to warn the user of pending problems. A reply found on Insecure.org stated that Undernet.org would scan for port 1080 and then warn node owners that they had an open socks server live on the Internet prior to allowing them to connect to their IRC service (Fulton, 2000). There are no response packets sent back to 216.77.219.195 so it is difficult to analyze rather the expected return was of any value. As for the stimulus or response, it seems that it could be classified as a stimulus as the scanning system would expect to gain a list of open 1080 ports.

My conclusion is that this is a reconnaissance scan for future activity that could be considered either for the purpose of good or bad. Either way, a reconnaissance scan has a primary purpose of collecting data. In this case it is to find open connections on port 1080. With that stated, my assumption is that the source addressed is not spoofed due to the fact that the individual performing the

scan would like to collect and correlate the scan responses for some purpose. However with that stated, the packets do not conform to any OS found in the honeynet project collection. I believe that the packets contain some amount of crafting to further mask the scan. It is also highly likely that the packet could be sent through an exploited node under the attacker's control. From this point forward the paper will assume that this scan is an attempt to locate an active socks server to be exploited.

The address 216.77.219.195 resolves to a BellSouth user. See the bold line in the whois search below. The address may be a customer of their ISP service or an internal user.

Search results for: bellsouth.net

BellSouth.Net (BELLSO-1)
Bellsouth.Net (BELLSO-8)
BellSouth.net Inc. (BELL)
Bellsouth.Net (AS7891) BELLSOUTH-NET-BLK2 7891 - 7894
Bellsouth.Net (AS8060) BELLSOUTH-NET-BLK3 8060 - 8063
BellSouth.net Inc. (AS6380) BELLSOUTH-NET-BLK 6380 - 6389
BellSouth.Net BS-NOLN2 (NET-207-205-114-0-1) 207.205.114.0 - 207.205.115.255
BellSouth.net Inc. BELLSNET-BLK1 (NET-205-152-0-0-1) 205.152.0.0 - 205.152.255.255
BellSouth.net Inc. BELLSNET-BLK4 (NET-209-214-0-0-1) 209.214.0.0 - 209.215.255.255
BellSouth.net Inc. BELLSNET-BLK5 (NET-216-76-0-0-1) 216.76.0.0 - 216.79.255.255
BellSouth.net Inc. BELLSNET-BLK7 (NET-208-60-0-0-1) 208.60.0.0 - 208.63.255.255
BellSouth.net Inc. BELLSNET-BLK8 (NET-66-20-0-0-1) 66.20.0.0 - 66.21.255.255
BellSouth.net Inc. BELLSNET-BLK9 (NET-65-80-0-0-1) 65.80.0.0 - 65.83.255.255
BellSouth.net Inc. BELLSNET-BLK10 (NET-66-156-0-0-1) 66.156.0.0 - 66.157.255.255
BellSouth.net Inc. BELLSNET-BLK12 (NET-67-32-0-0-1) 67.32.0.0 - 67.35.255.255
BellSouth.net Inc. BELLSNET-BLK13 (NET-68-16-0-0-1) 68.16.0.0 - 68.19.255.255
BellSouth.net Inc. BELLSNET-BLK14 (NET-68-152-0-0-1) 68.152.0.0 - 68.159.255.255
BellSouth.net Inc. BELLSNET-BLK15 (NET-68-208-0-0-1) 68.208.0.0 - 68.214.255.255
BellSouth.net Inc. BELLSNET-BLK11 (NET-206-223-128-0-1) 206.223.128.0 - 206.223.128.255
BellSouth.net Inc. BELLSNET-BLK2 (NET-207-203-0-0-1) 207.203.0.0 - 207.203.255.255
BellSouth.net Inc. BELLSNET-BLK3 (NET-209-149-0-0-1) 209.149.0.0 - 209.149.255.255

Description of the Attack

The scan would not be considered an attack but rather some type of reconnaissance. Should the reconnaissance provide what the attacker hopes to find, in this case an open tcp 1080 port, then the scan activity could turn into an attack. The scan was performed by sending a SYN packet. A valid response in the form of a SYN ACK would tell the attacker that the port is open.

One interesting element of the scan is that the time between packets in the scan is 66.4 seconds. What can be deduced from the time gap between packets? This will be discussed in the evidence of active targeting section.

SOCKS port 1080 is generally used to tunnel Internet traffic through the firewall using a single IP address. It should be configure to only let traffic out, but

poorly configured firewalls can pass traffic both ways. Hoyt points out that the popular Windows personal firewall WinGate is often misconfigured this way (HoytDuff, 2003). Older versions of WinGate had a default install to allow traffic both in and out of the firewall. This would allow the attacker access to victims behind the firewall.

A second attack, which is more likely, given the October 2002 time frame of this detect, is a buffer overflow vulnerability with an AnalogX Proxy, which can be exploited through tcp port 1080. The vulnerability was discovered and released in July of 2002. It has a CVE number of CAN-2002.1001.

Attack Mechanism

While the scan is not an attack mechanism in and of itself, the ability of the scanner to find open SOCKS proxy ports will most likely allow the attacker to use systems on the victim network. If the attacker can locate open proxies, then anonymity for clandestine activity is probably the goal. The activity can be as simple as using the bounce site for IRC connectivity to launching a full scale attack. The attack can come from a node behind the firewall or through the proxy itself. This can effectively mask the attacker in the short-term until sites and authorities work together to track the actual source (Jatt). Many spammers are using the SOCKS port exploit gained through scanning to mask their spamming activity. This allows the spammer to send their material virtually undetected as the source address is masked via the proxy sending the data.

As I pointed out earlier, given the time frame, the scan was probably looking for an open tcp port 1080 to attempt an AnalogX proxy overflow. The exploit on tcp port 1080 is done by sending a SOCKS 4A request with a hostname containing more than 140 characters. This would cause a write access violation and the application would error (Ahmad, 2002). The attacker could then execute arbitrary code on the system with the privileges held by the server (X-Force, 2002).

Correlations

The CVE number for the tcp port 1080 exploit for the AnalogX proxy is CAN-2002-1001 (CVE, 2002).

Information on the AnalogX proxy exploit was also found on Security Focus with Ahmad's report and on Internet Security Systems site with the X-Force research report.

Many scans of port 1080 can be found on the Internet using "port 1080" and SYN as search criteria on google. I was able to correlate some events with this search during the time frame that these packets were collected.

Evidence of Active Targeting

The scan appears to be evidence of some attacker attempting to locate a target. The detect contains 864 captures from the 216.77.219.119 source address. The scan seems to randomly pick IP addresses and is slowly iterating through potential targets. There is an approximate 66.4 second gap between each packet. This could indicate that the scan is trying to avoid detection or that the 864 detects in this capture are a small subset of a larger scan. Given the 66.4 second gap between all packets in the scan, the probability of not find two IP addresses within the random scope of the local addresses thus creating variability in the time is highly unlikely. My assumption is that this scan was given a scope and set to scan, and then log the response every 60 seconds. So, could the scan be attempting to avoid detection given the discussion above? Yes, it is a possibility, but assuming the person performing the scan is reasonably intelligent, a quick search of IDS rule sets would provide the answer. I may be giving to much credit to the potential attacker, but most detection systems will log this type of activity no matter how slow, if the target is running an IDS and monitoring the logs.

Severity

The values provided in the formula below are rated on a scale from 1 to 5 where 1 is the lowest and 5 is the highest.

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

Criticality = 1 due to the random nature of the chosen IP addresses

Lethality = 1 due to the fact that the scan is considered reconnaissance

System countermeasures = 1 assuming systems are patched or not vulnerable

Network countermeasures = 3 assuming that there are few counter measures ahead of the detection sensor. My assumption is based on the large volume of different types of detections contained in the collection

Severity = (1 + 1) – (1 + 3)
Severity = -2

Defensive Recommendation

Make sure the SOCKS proxy configuration is configure to exit the network only. The proxy should have no provisions for passing traffic from the internet to the LAN. If the network is running AnalogX proxy versions earlier than 4.12, the software must be patched or upgraded.

Multiple Choice Test Question

The SOCKS proxy buffer overflow exploit in the AnalogX proxy is performed by sending malformed packets on which port?

a. udp 1080
b. tcp 1080
c. padded icmp packet
d. none of the above

answer is b

Top Three Questions or Comments From intrusions@incidents.org
First Posting on 12/07/03 – Received no comments
Second Posting on 12/11/03 – The top three questions were from T. Hudak

1. In reference to the probability of packet spoofing; Does everything look normal? Look at the ACK number. Is it usually set on a SYN packet?
    a. This I overlooked in the packets when discussing spoofing. Tyler's comment made me take another look at the SYN and ACK sequence numbers. The fact that the packet has a SYN flag set and an ACK flag not set indicates that there should be no ACK value. Another tell-tale sign of packet crafting is that the SYN and ACK sequence numbers are the same. It is highly unlikely that one packets SYN and ACK sequence values could be the same and definitely not possible with successive connections.
2. In reference to the Undernet; Undernet is an IRC network that would scan your computer for open proxy ports (SOCKS included) before they let you connect. It may just be the way I'm reading your statement above, but it sounds like you are saying that Undernet was a group that would randomly scan the Internet looking for open SOCKS ports and warn their owners.
    a. A clarification seems to be needed. Undernet as a provider does not want vulnerable systems connected to their service, so they will provide information to the user prior to connecting to their IRC network.
3. In reference to the description of the attack: With this scan, how will the attacker know if the port is open or closed?
    a. By sending a SYN packet to tcp port 1080, if the port was open the expected response would be a SYN ACK. On the other side, if the port was closed, no response would be sent.

References for detect 3:

Ahmad, Dave. “FS
Advisory ID: FS-070102-23-AXPR.” URL:
http://archives.neohapsis.com/archives/bugtraq/2002-07/0006.html (05 October 2003).

Common Vulnerabilities and Exposures. “CAN-2002-1001 (under review).” URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name-CAN-2002-1001 (05 October 2003).

Fulton, Russell. “Security Incidents: Re: Socks port 1080.” 20 Jan 2000. URL: http://lists.insecure.org/lists/incidents/2000/Jan/0170.html (21 September 2003).

Honeynet Project. “Lists of fingerprints for passive fingerprint monitoring.” 23 May 2000 URL: http://project.honeynet.org/papers/finger/traces.txt (21 September 2003).

HoytDuff. “Re: attack alert on port 1080.” 16 Aug 2003. URL: http://www.mail-archive.com/redhat-list@redhat.com/msg123732.html (21 September 2003).

Jatt, Martin. “Proxy/WinGate/SOCKS.” URL: http://rrfn.promodtecnologies.com /K-base/Proxy-WinGate-SOCKS_Tutorial.html (21 September 2003).

X-Force Research. “Analogx Proxy long Socks4a request buffer overflow.” URL: http://xforce.iss.net/xforce/xfdb/9456 (05 October 2003).

Part 3: Analyze This

Executive Summary

The period of analysis for this report is from November 11th, 2003 to November 15th, 2003. A total of fifteen files containing five Snort alert logs, five scan logs, and five out of spec logs were provided and used during the analysis of the university's enterprise environment. The top ten alerts by volume were analyzed along with other alerts that appeared to be potential vulnerabilities. The data indicates that most of the activity in the top ten events is due to poor IDS tuning, selection of IDS location, or both. The top ten alerts make up greater than 97% of the alert activity captured for the five day period.

A Priority was assigned to each event, which indicates the potential of the vulnerability. Of the top ten alerts analyzed, four are considered a low priority and six are considered a medium priority. The medium priorities should be review by following the recommendations for each alert.

The analysis includes summaries of scan data and several top talker charts. The top talker charts provide data on source and destination IP's as well as scan data by destination IP and port activity.

A final recommendation was created from a compilation of the individual recommendations in the alert analysis. The primary recommendations include a general tightening of the Snort rule set as the mountain of data supplied was excessive. An anti-virus plan should be implemented and monitored to provide current anti-virus engines and dat files. The anti-virus logs should be monitored on critical systems for virus activity. Finally, the boundary router's access control lists should be tightened to encompass both ingress and egress activity. The IDS should be used as the final defense in a layered security model. Firewall rules should be reviewed and tightened. A plan to use the IDS activity as continuous improvement for the firewall and router ACL's should be developed and implemented.

The Analysis

The following files were used for the "analyze this" section.

Alert Files		
File Name	Download File Size	Unpacked File Size
alert.031111.gz	2988841	33850037
alert.031112.gz	2275129	26011757
alert.031113.gz	2475280	28374014
alert.031114.gz	3410545	40453583
alert.031115.gz	2897508	33412551

Scan Files		
File Name	Download File Size	Unpacked File Size
scan.031111.gz	23880479	210432184
scan.031112.gz	15677724	140991912
scan.031113.gz	18529420	167236556
scan.031114.gz	21361284	196006847
scan.031115.gz	22959104	210279780

Out of Spec Report		
File Name	Download File Size	Unpacked File Size
oos_report_031111	2859008	N/A
oos_report_031112	2859008	N/A
oos_report_031113	2859008	N/A
oos_report_031114	2859008	N/A
oos_report_031115	2859008	N/A

The method used to roll up the alert files was SnortSnarf. The application parses all data and assembles the output in an easy to read web page. It also provides a top twenty source and destination output. Many practicals that I read had problems using SnortSnarf due to a multitude of reasons. In order for my attempt to use SnortSnarf, the alert data was reduced to a manageable file size by removing all portscan data. The scan logs will be used to supplement the portscan data that was removed. The five reduced alert files were concatenated and then parsed with SnortSnarf. To allow SnortSnarf to efficiently analyze the logs, “MY.NET” was replaced with “130.85”. Any IP address containing “130.85” from this point forward is considered to be the local University network. “130.85” was chosen as the network prefix upon review of the scan logs provided. All data in the scan logs contained the prefix and an nslookup returned valid university fully qualified names. Using SnortSnarf output, the top ten alerts by volume will be analyzed first, with interesting alerts highlighted second.

Since the portscan data was removed from the alert files, all scan data and Out of Spec (OOS) files will be analyzed and used to support conclusions drawn from the alert data. An interesting element to point out about the OOS logs is that all five logs are the same size. See the Out of Spec Report chart above. All five OOS files contain data that starts at 10/27-00:06:01 and ends at 10/29-14:30:59. Since they contain no data for the period of time that the alert and scan logs cover, I am not sure how they will help.

Viewing the shear amount of data, two hypotheses can be made. First, Snort sensors are improperly placed in the environment. If Snort sensors are placed outside the perimeter of the local network, they will produce many alerts for

packets that may not be able to reach internal systems. This scenario can create many false positives. Second, little or no tuning seems to have taken place. Or, the rules that Snort uses are too generic and must be more specific to the environment. The majority of alerts trigger on general activity instead of specific vulnerabilities. This would seem to be the reason for so many false positives. In the five day period, there were 15,051,717 triggers by combining the alert and scan data, considerable system resources are being wasted. I would assume the system has intermittent problems under the volume of data. According to a meeting with SourceFire representatives, approximately one-million alerts equal one gigabyte of disk space. In this case approximately fifteen gigabytes are being consumed every five days.

The volume of alerts and false positives probably leads to the “cry wolf syndrome” where serious alerts may be overlooked. The volume shown in the SnortSnarf output that follows could not possibly be managed with any efficiency by IDS analysts. This further supports the belief that system tuning would go a long way in pairing down the amount of data that is collected by the IDS.

The top 10 alerts by volume in section one will be given a priority. This priority will be based on severity to the enterprise environment. The recommendation following each analysis will provide an insight to the severity and why the priority was set at low, medium, or high. The three priority categories with an explanation of each category are shown in table 3.1.

Table 3.1 Priority Explanations

Priority	Comments
High	Requires immediate attention as this item presents a real threat to the enterprise or environment
Medium	Requires attention as this item presents a short-term or long-term pending threat to the enterprise or environment
Low	Should be reviewed, could be vulnerable at some point

Table 3.2 contains all SnortSnarf output sorted from the alert files. SnortSnarf output in the webpage was placed in the chart by alert volume in ascending order. To emphasis the top ten alerts in bold print, I placed the data shown below in a descending order so the top ten alerts for the first section of analysis are at the top. The rest of the data is shown for reference purposes only as greater than 97% of the IDS activity is in the top ten alerts.

Table 3.2 Alert Roll up Created by SnortSnarf

Signature (click for sig info)	# Alerts	# Sources	# Dests
130.85.30.4 activity	**104330**	287	1
SMB Name Wildcard	**13232**	198	10037
Incomplete Packet Fragments Discarded	**8103**	59	118
connect to 515 from inside	**5172**	2	2
High port 65535 tcp - possible Red Worm - traffic	**3981**	105	125

High port 65535 udp - possible Red Worm - traffic	**3939**	189	85
EXPLOIT x86 NOOP	**3711**	275	112
130.85.30.3 activity	**3695**	78	1
SUNRPC highport access!	**1363**	35	89
connect to 515 from outside	**747**	3	113
NMAP TCP ping!	722	166	46
[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.	525	57	59
Possible trojan server activity	378	46	44
EXPLOIT x86 stealth noop	348	10	11
[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC	332	3	2
Null scan!	311	34	23
External RPC call	290	4	181
Traffic from port 53 to port 123	128	1	1
ICMP SRC and DST outside network	120	34	117
TCP SRC and DST outside network	109	21	50
FTP passwd attempt	109	70	29
[UMBC NIDS] External MiMail alert	106	48	1
SMB C access	96	34	3
RFB - Possible WinVNC - 010708-1	82	21	19
EXPLOIT x86 setuid 0	35	30	26
FTP DoS ftpd globbing	29	7	1
EXPLOIT x86 setgid 0	25	22	23
TFTP - Internal UDP connection to external tftp server	17	6	6
IRC evil - running XDCC	11	1	1
External FTP to HelpDesk 130.85.70.49	11	7	1
TCP SMTP Source Port traffic	10	1	1
Tiny Fragments - Possible Hostile Activity	9	5	4
EXPLOIT NTPDX buffer overflow	9	5	5
External FTP to HelpDesk 130.85.70.50	9	6	1
Attempted Sun RPC high port access	9	6	6
External FTP to HelpDesk 130.85.53.29	7	6	1
[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC	6	2	1
DDOS mstream client to handler	6	2	2
NETBIOS NT NULL session	6	2	2
[UMBC NIDS IRC Alert] K\:line'd user detected, possible trojan.	6	3	3
NIMDA - Attempt to execute cmd from campus host	6	6	3
Probable NMAP fingerprint attempt	5	3	3
TFTP - External UDP connection to internal tftp server	3	2	2
PHF attempt	3	3	2
DDOS shaft client to handler	2	1	1
[UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot	2	2	2
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:24:26.839786 [**] 130.85.30.4 activity	1	1	1

130.85.30.4 activity [**] 67.21.63.1511/14-08:41:59.589747 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:44:28.873206 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:30:52.419890 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:18.506866 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:53:25.091300 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:43:40.920234 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:41:45.885391 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:28:52.670061 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:22:25.109429 [**] 130.85.30.4 activity	1	1	1
SMB Name Wildcard [**] 130.85.80.5111/11-15:17:53.278126 [**] SMB Name Wildcard	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:52:37.104082 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:43:43.357839 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:11:19.542619 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:47:04.391352 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:14:20.582558 [**] 130.85.30.4 activity	1	1	1
SMB Name Wildcard [**] 130.85.80.5111/11-14:47:40.127165 [**] SMB Name Wildcard	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:30:52.400953 [**] 130.85.30.4 activity [**] 67.21.63.1511/14-08:14:04.534460 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:42:04.050490 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:43:57.016716 [**] 130.85.30.4 activity	1	1	1
[UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected.	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:18.512840 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:52:52.261139 [**] 130.85.30.4 activity	1	1	1
SMB Name Wildcard [**] 130.85.80.5111/11-15:38:08.958248 [**] SMB Name Wildcard	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:52:25.140206 [**] 130.85.30.4 activity	1	1	1

130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:42:14.915560 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:53:26.605734 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:41:59.585257 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:30:52.407820 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:14:58.603784 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:28:49.450068 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:33:59.009310 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:42:27.162992 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:14:58.609461 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:06:53.173089 [**] [UMBC NIDS IRC Alert] XDCC client detected attempting to IRC	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:52.245119 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:29:06.872456 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:52.256076 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:45:22.670293 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:56:00.438755 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:33:35.769318 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:41:59.581257 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:25:33.739234 [**] 130.85.30.4 activity	1	1	1
Bugbear@MM virus in SMTP	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:53:46.099271 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:18.495025 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:31:01.949781 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:29:24.869948 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:12:17.990901 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:30:43.652384 [**] 130.85.30.4 activity	1	1	1

130.85.30.4 activity [**] 67.21.63.1511/14-08:30:55.089728 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:43:41.585871 [**] 130.85.30.4 activity	1	1	1
FTP .rhosts	1	1	1
SMB Name Wildcard [**] 130.85.80.5111/11-16:04:34.116144 [**] SMB Name Wildcard	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:52:24.686568 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:24:31.775011 [**] 130.85.30.4 activity	1	1	1
SMB Name Wildcard [**] 130.85.80.5111/11-15:56:59.363876 [**] SMB Name Wildcard	1	1	1
SMB Name Wildcard [**] 130.85.80.5111/11-14:12:23.283792 [**] SMB Name Wildcard	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:29:03.526208 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-09:27:02.599660 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:53:16.527813 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-15:55:20.447590 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:43:32.430602 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:45:22.676168 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:53:50.401510 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:18.483003 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:31:03.165520 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:24.552433 [**] 130.85.30.4 activity	1	1	1
SMB Name Wildcard [**] 130.85.80.5111/11-14:08:34.581330 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:12:35.063633 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:46:04.177878 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:07.981382 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:25:23.932639 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:22:16.153737 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:21:23.895049 [**] 130.85.30.4 activity	1	1	1

130.85.30.4 activity [**] 67.21.63.1511/14-08:56:00.574896 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:15.084312 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:18.488986 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:30:52.425664 [**] 130.85.30.4 activity	1	1	1
Incomplete Packet Fragments Discarded [**] 130.85.21.3711/14-01:20:57.861074 [**] Incomplete Packet Fragments Discarded	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:21:56.036502 [**] 130.85.30.4 activity	1	1	1
[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan. [**] 64.157.246.2411/14-08:16:02.015807 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:24:00.620640 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:28.554639 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:30:52.413541 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:14:06.552942 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:07:46.066160 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:23:43.534920 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:14:51.163079 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:56:00.445195 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:18.501182 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:28:59.154952 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:14:23.304493 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:24:24.134565 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:13:59.028429 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-07:58:30.051106 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:21:43.124564 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:21:47.173719 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:56:10.615010 [**] 130.85.30.4 activity	1	1	1

130.85.30.4 activity [**] 67.21.63.1511/14-08:21:34.636935 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:21:46.646165 [**] 130.85.30.4 activity	1	1	1
SMB Name Wildcard [**] 130.85.80.5111/11-14:17:22.050305 [**] SMB Name Wildcard	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:14:15.131615 [**] 130.85.30.4 activity [**] 67.21.63.1511/14-08:23:49.972632 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:12:31.591278 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:41:07.834524 [**] 130.85.30.4 activity	1	1	1
SMB Name Wildcard [**] 130.85.80.5111/11-15:15:36.541513 [**] SMB Name Wildcard	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:22:03.053802 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.15:1060 -> 130.85.30.411/14-08:12:47.405410 [**] 130.85.30.4 activity	1	1	1
130.85.30.4 activity [**] 67.21.63.1511/14-08:46:38.701178 [**] 130.85.30.4 activity	1	1	1

Top Ten Alerts by Volume

The top ten alerts section will review the top ten alerts by volume. The top ten alerts may not cover all perceived malicious activity within the environment. Table 3.3 shows the alerts that will be covered as a percentage of the total volume of alerts collected for the five day period.

Table 3.3 Top Ten shown as a Percent of Total

Alert	% of Total
130.85.30.4 activity	68.52%
SMB Name Wildcard	8.69%
Incomplete	5.32%
connect to 515 from inside	3.40%
TCP Red Worm	2.61%
UDP Red Worm	2.59%
Exploit x86 NOOP	2.44%
130.85.30.3 activity	2.43%
SUNRPC High port access	0.90%
Connect to port 515 from outside	0.47%
Percent of Top Ten Alerts to Total Alerts	97.37%

Alert #1:

Alert Signature: 130.85.30.4 activity
Alerts Generated: 104330
Source Addresses Involved: 287
Destination Addresses Involved: 1
Priority: Medium

Sample Alert Data for Alert #1

11/14-07:31:13.859453 [**] 130.85.30.4 activity [**] 67.21.63.15:1035 -> 130.85.30.4:524
11/14-07:31:13.887028 [**] 130.85.30.4 activity [**] 67.21.63.15:1035 -> 130.85.30.4:524
11/14-07:31:14.015465 [**] 130.85.30.4 activity [**] 67.21.63.15:1035 -> 130.85.30.4:524

The top three sources for "130.85.30.4 activity" are; 67.21.63.15, 68.81.2.19, and 68.55.205.18 with 92317, 3612, and 1193 alerts respectively. The following ARIN's search is shown to provide data for the validity of the IP addresses.

Search results for: 67.21.63.15

```
Adelphia Cable Communications ADELPHIA-CABLE-5 (NET-67-20-0-0-1)
                                  67.20.0.0 - 67.23.255.255
Adelphia 6721480-Z5 (NET-67-21-48-0-1)
                                  67.21.48.0 - 67.21.63.255

nslookup for 67.21.63.15: md-wmnsmd-cuda2-c1d-15.chvlva.adelphia.net
```

Search results for: 68.81.2.19

```
Comcast Cable Communications, Inc. JUMPSTART-2 (NET-68-80-0-0-1)
                                  68.80.0.0 - 68.87.255.255
Comcast Cable Communications, Inc. PA-METRO-7 (NET-68-80-0-0-2)
                                  68.80.0.0 - 68.81.255.255

nslookup for 68.81.2.19: pcp228604pcs.catonv01.md.comcast.net
```

Search results for: 68.55.205.180

```
Comcast Cable Communications, Inc. JUMPSTART-1 (NET-68-32-0-0-1)
                                  68.32.0.0 - 68.63.255.255
Comcast Cable Communications, Inc. BALTIMORE-A-6 (NET-68-55-0-0-1)
                                  68.55.0.0 - 68.55.255.255

nslookup for 68.55.205.180: pcp228604pcs.catonv01.md.comcast.net
```

I am not sure of the Internet Service Provider (ISP) customer IP address fully qualified name designation for the above carriers. Based on the nslookup of the three addresses they could be customers or internal nodes of either Adelphia or

Comcast. Since these nodes are accessing the university remotely, my guess is that they are dhcp assigned addresses of customers to the ISP.

There is considerable activity around the 130.85.30.4 node. A good majority of the activity is requests on port 80, which would provide strong support to the fact that this node is a web server. Further there is some activity on port 51443 which is the SecureListen port on NetWare 6.0 and earlier running Apache Web Server (Balasubramaniam). In further support there is some RealPlayer and streaming media accessed from different queries on the server. The server is also enabled as an FTP server.

CVE indicates that an exploit with wu-ftpd 2.6.1 allows remote attackers to execute arbitrary code via a "~{" argument. Commands are able to be executed as they are not properly handled by the glob function (ftpglod) (CVE-2001-0550).

The volume of traffic from 67.21.63.15 can either be scan activity or time synchronization updates from a server that belongs to Adelphia (Novell appNotes). The later is more likely due to the consistent source and destination port numbers. Also the scan logs contain no entries for this IP address.

Search results for: 130.85.30.4

```
OrgName:    University of Maryland Baltimore County
OrgID:      UMBC
Address:    UMBC University Computing
City:       Baltimore
StateProv:  MD
PostalCode: 21250
Country:    US

NetRange:   130.85.0.0 - 130.85.255.255
CIDR:       130.85.0.0/16
NetName:    UMBCNET
NetHandle:  NET-130-85-0-0-1
Parent:     NET-130-0-0-0-0
NetType:    Direct Assignment
NameServer: UMBC5.UMBC.EDU
NameServer: UMBC4.UMBC.EDU
NameServer: UMBC3.UMBC.EDU
Comment:
RegDate:    1988-07-05
Updated:    2000-03-17

TechHandle: JJS41-ARIN
TechName:   Suess, John J.
TechPhone:  +1-410-455-2582
TechEmail:  jack@umbc.edu

# ARIN WHOIS database, last updated 2003-12-05 19:15
# Enter ? for additional hints on searching ARIN's WHOIS database.
```

Entering http://130.85.30.4/ in a web browser provides a webpage for Novell file share management.

Recommendation: Based on the shear volume of alert activity logged, it would appear that too many generalized alerts are being logged from this web server. The activity to this web server makes up 68.52% of the alert volume for the five days analyzed. Research should be performed around the exploits that the server is susceptible too and the IDS should be tuned to log only pertinent alerts that the system is vulnerable too. The reason for the medium priority is that IDS analysts and system resources could be quickly consumed with this one alert. With the amount of volume generated from this alert, critical alerts may be overlooked.

Alert #2:

Alert Signature: SMB Name Wildcard
Alerts Generated: 13232
Source Addresses Involved: 198
Destination Addresses Involved: 10037
Priority: Low

Sample Alert Data for Alert #2

11/11-14:04:59.523116 [**] SMB Name Wildcard [**] 130.85.80.51:1036 -> 13.5.61.55:137
11/11-14:04:59.973098 [**] SMB Name Wildcard [**] 130.85.80.51:1036 -> 13.5.61.58:137
11/11-14:05:00.123085 [**] SMB Name Wildcard [**] 130.85.80.51:1036 -> 13.5.61.59:137

The majority of SMB Name Wildcard traffic in this alert originates in the local network. The two destinations receiving the most activity are; 169.254.0.0 and 169.254.45.176. These addresses according to RFC 3330 are invalid IP addresses for the Internet. The 169.254.0.0/16 range of addresses are reserved for hosts attempting to locate a DHCP lease and unable to locate a DHCP server (Cheshire, 2003). This traffic would indicate that a host was probably rebooted and was looking for a lease and has assigned itself the temporary address of 169.254.xx.xx.

SMB traffic is considered to be normal NetBios traffic with many Windows machines and some Linux nodes if configured as a Samba server or client. All SMB alerts provided for the five day traffic analysis originate from within the home network. Some reasons for SMB traffic to outside sources may include negotiations from connections such as identd requests from a mail server or IRC server (arachnids database, IDS177).

Recommendation: The traffic should be analyzed for malicious activity from potential file sharing and the SMB traffic should be blocked at the network

perimeter. There is no good reason for NetBios traffic to enter or leave the local network. Look at firewall rules and/or router ACL's to block this egress traffic.

Alert #3:

Alert Signature: Incomplete Packet Fragments Discarded
Alerts Generated: 8103
Source Addresses Involved: 59
Destination Addresses Involved: 118
Priority: Medium

Sample Alert Data for Alert #3

11/14-01:13:41.989277 [**] Incomplete Packet Fragments Discarded [**] 130.85.21.37 -> 66.68.195.62
11/14-01:13:43.016875 [**] Incomplete Packet Fragments Discarded [**] 130.85.21.37 -> 66.68.195.62
11/14-01:13:44.974964 [**] Incomplete Packet Fragments Discarded [**] 130.85.21.37 -> 66.68.195.62

Cormier provides a link in his analysis for this type of fragmented traffic, which is a response from Marty Roesch (Cormier, 2003). Mr. Roesch states that the frag2 pre-processor was designed to eliminate the "Incomplete Packet Fragments Discarded" problem (Roesch, 2001).

Table 3.4 Top Ten Alerts for Fragmented Traffic

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
130.85.21.37	1416	1416	5	5
130.85.21.67	1298	1298	8	8
130.85.21.69	1090	1090	8	8
130.85.21.68	1078	1079	8	8
130.85.21.92	1031	1031	6	6
130.85.97.64	878	878	1	1
130.85.21.116	803	803	4	4
130.85.21.79	276	276	4	4
130.88.60.31	118	118	79	79
63.241.23.111	26	26	1	1

The analysis also indicates that there could be an issue on subnet 21 as shown in table 3.4. Of the 8103 total alerts in the "Incomplete Packet Fragments Discarded" collection, 86% of the alerts originate on subnet 21. This would provide data to further look at subnet 21 and see if there is a hardware problem from some network devise.

Recommendation: If the Snort IDS is currently running the defrag pre-processor, upgrade to the frag2 pre-processor. Also perform a packet capture on subnet 21 and look for hardware issues. Locate and repair any issues found on subnet 21. This item is rated medium due to the lack of a packet analysis. If no hardware problems are found on subnet 21, the priority could be reduced to low.

Alert #4:

Alert Signature: connect to 515 from inside
Alerts Generated: 5172
Source Addresses Involved: 2
Destination Addresses Involved: 2
Priority: Medium

Sample Alert Data for Alert #4

```
11/11-02:39:04.252257 [**] connect to 515 from inside [**] 130.85.162.41:721 -
> 128.183.110.242:515
11/11-02:39:32.295270 [**] connect to 515 from inside [**] 130.85.162.41:721 -
> 128.183.110.242:515
11/11-02:39:35.298677 [**] connect to 515 from inside [**] 130.85.162.41:721 -
> 128.183.110.242:515
```

Search results for: 128.183.110.242

```
OrgName:    National Aeronautics and Space Administration
OrgID:      NASA
Address:    AD33/Office of the Chief Information Officer
City:       MSFC
StateProv:  AL
PostalCode: 35812
Country:    US

NetRange:   128.183.0.0 - 128.183.255.255
CIDR:       128.183.0.0/16
NetName:    GSFC
NetHandle:  NET-128-183-0-0-1
Parent:     NET-128-0-0-0-0
NetType:    Direct Allocation
NameServer: NS.GSFC.NASA.GOV
NameServer: NS2.GSFC.NASA.GOV
Comment:
RegDate:    1993-04-01
Updated:    2003-02-05

TechHandle: ZN7-ARIN
TechName:   National Aeronautics and Space Administration
```

```
TechPhone:  +1-256-544-5623
TechEmail:  dns.support@nasa.gov

OrgAbuseHandle: NASAA-ARIN
OrgAbuseName:   NASA Abuse
OrgAbusePhone:  +1-800-762-7472
OrgAbuseEmail:  abuse@nasa.gov

OrgNOCHandle: NISN-ARIN
OrgNOCName:   NASA Information Services Network
OrgNOCPhone:  +1-256-961-4000
OrgNOCEmail:  noc@nisn.nasa.gov

OrgTechHandle: WEBBN-ARIN
OrgTechName:   Webb, Nancy
OrgTechPhone:  +1-256-544-3245
OrgTechEmail:  dns.support@nasa.gov

Nslookup for 128.183.110.242: tek924.gsfc.nasa.gov
```

According to RFC 1179, valid request ports for line printer daemons listening on port 515 are 721-731 inclusively (McLaughlin, 1990). This appears to be valid traffic. The question that needs to be asked is; why is the print traffic going to NASA? The top source is shown above and an ARIN search is provided for support.

Recommendation: The priority is medium due to the offsite printing. First investigate why the print traffic goes to NASA. If it is valid traffic, write a pass rule to pass or log the traffic. Also make sure it has the proper security around the traffic. If this is valid, make sure proper encryption is enabled, if sensitive data is being sent.

Alert #5: & Alert #6:

Alert #5 Signature: High port 65535 tcp – possible Red Worm - traffic
Alerts Generated: 3981
Source Addresses Involved: 105
Destination Addresses Involved: 125
Priority: Medium

Alert #6 Signature: High port 65535 udp – possible Red Worm - traffic
Alerts Generated: 3939
Source Addresses Involved: 189
Destination Addresses Involved: 85
Priority: Medium

Sample Alert Data for Alert #5

```
11/14-14:18:29.465743 [**] High port 65535 tcp - possible Red Worm - traffic [**]
129.165.254.6:65535 -> 130.85.162.56:49504
```

11/14-14:18:29.465891 [**] High port 65535 tcp - possible Red Worm - traffic [**] 129.165.254.6:65535 -> 130.85.162.56:49504
11/14-14:18:29.466325 [**] High port 65535 tcp - possible Red Worm - traffic [**] 129.165.254.6:65535 -> 130.85.162.56:49504

Sample Alert Data for Alert #6

11/15-02:05:31.635889 [**] High port 65535 udp - possible Red Worm - traffic [**] 219.1.220.74:65535 -> 130.85.70.176:6257
11/15-02:05:32.140049 [**] High port 65535 udp - possible Red Worm - traffic [**] 219.1.220.74:65535 -> 130.85.70.176:6257
11/15-02:05:32.704911 [**] High port 65535 udp - possible Red Worm - traffic [**] 219.1.220.74:65535 -> 130.85.70.176:6257

Search results for: 129.165.254.6

```
OrgName:    National Aeronautics and Space Administration
OrgID:      NASA
Address:    AD33/Office of the Chief Information Officer
City:       MSFC
StateProv:  AL
PostalCode: 35812
Country:    US

NetRange:   129.165.0.0 - 129.165.255.255
CIDR:       129.165.0.0/16
NetName:    NASA-GSFCSSE
NetHandle:  NET-129-165-0-0-1
Parent:     NET-129-0-0-0-0
NetType:    Direct Allocation
NameServer: NS.GSFC.NASA.GOV
NameServer: NS2.GSFC.NASA.GOV
Comment:
RegDate:    1988-01-04
Updated:    2002-09-05

AbuseHandle: ZN13-ARIN
AbuseName:   Network Engineering Branch
AbusePhone:  +1-301-286-6984
AbuseEmail:  ionet-pm@listserv.gsfc.nasa.gov

NOCHandle: ZN13-ARIN
NOCName:   Network Engineering Branch
NOCPhone:  +1-301-286-6984
NOCEmail:  ionet-pm@listserv.gsfc.nasa.gov

TechHandle: ZN13-ARIN
TechName:   Network Engineering Branch
TechPhone:  +1-301-286-6984
TechEmail:  ionet-pm@listserv.gsfc.nasa.gov
```

```
OrgAbuseHandle: NASAA-ARIN
OrgAbuseName:   NASA Abuse
OrgAbusePhone:  +1-800-762-7472
OrgAbuseEmail:  abuse@nasa.gov

OrgNOCHandle: NISN-ARIN
OrgNOCName:   NASA Information Services Network
OrgNOCPhone:  +1-256-961-4000
OrgNOCEmail:  noc@nisn.nasa.gov

OrgTechHandle: WEBBN-ARIN
OrgTechName:   Webb, Nancy
OrgTechPhone:  +1-256-544-3245
OrgTechEmail:  dns.support@nasa.gov

Nslookup for 129.165.254.6: g0acg01u.ecs.nasa.gov
```

Search results for: 203.181.25.21

```
OrgName:    Asia Pacific Network Information Centre
OrgID:      APNIC
Address:    PO Box 2131
City:       Milton
StateProv:  QLD
PostalCode: 4064
Country:    AU

ReferralServer: whois://whois.apnic.net

NetRange:   202.0.0.0 - 203.255.255.255
CIDR:       202.0.0.0/7
NetName:    APNIC-CIDR-BLK
NetHandle:  NET-202-0-0-0-1
Parent:
NetType:    Allocated to APNIC
NameServer: NS1.APNIC.NET
NameServer: NS3.APNIC.NET
NameServer: NS.RIPE.NET
NameServer: RS2.ARIN.NET
NameServer: DNS1.TELSTRA.NET
Comment:    This IP address range is not registered in the ARIN
database.
Comment:    For details, refer to the APNIC Whois Database via
Comment:    WHOIS.APNIC.NET or http://www.apnic.net/apnic-bin/whois2.pl
Comment:    ** IMPORTANT NOTE: APNIC is the Regional Internet Registry
Comment:    for the Asia Pacific region. APNIC does not operate
networks
Comment:    using this IP address range and is not able to investigate
Comment:    spam or abuse reports relating to these addresses. For more
Comment:    help, refer to http://www.apnic.net/info/faq/abuse
Comment:
RegDate:    1994-04-05
Updated:    2002-09-11
```

```
OrgTechHandle: AWC12-ARIN
OrgTechName:   APNIC Whois Contact
OrgTechPhone:  +61 7 3858 3100
OrgTechEmail:  search-apnic-not-arin@apnic.net

Nslookup for 203.181.25.21: no resolve
```

There seems to be a considerable amount of traffic in and out of the local network. An address from a node at NASA as shown above is contacting a node within the local network on port 65535. There is also considerable traffic from the Asia Pacific region as indicated by the UDP traffic and the ARIN search. This has the potential to be malicious traffic. Without packets to analyze, the assumption is that this is Red Worm traffic as flagged by the IDS.

Port 65535 is used by a node infected with the Linux or UNIX.Red.Worm or Adore. As shown in both Alert #5 and #6, at least one node within the local network and others outside the local network are communicating. Many of the addresses outside the local network originate in the Asia Pacific range of IP addresses as shown in Alert #6.

According to Maher, the UDP alerts are not triggered by Red Worm activity (Maher, 2003). A closer look at the UDP alerts show that all alerts have a destination port of 6257, which may indicate a response from a source running a file share program like WinMX (WinMX).

The scan logs were consulted to verify rather data could be extracted to support rather a system is infected with “Red Worm”. The scan logs indicate that most traffic seems to be valid traffic that randomly chooses the ephemeral port 65535. The scan logs support that conclusion as there are not a large amount of scans for services that “Red Worm” would be expected to search for. For a more in depth analysis of the traffic, sample data should be collected by an analyzer or packet sniffer.

Recommendation: It does not appear that infections exist in the environment based on the analysis of alert and scan data only. To be safe, a packet analysis and search of system logs should be conducted to look for the existence of malware. Clean or rebuild affected systems and apply fixes if necessary. My assumptions, not having any further data on the type of systems that have the potential to be infected are that they are one of the major Linux varieties. According to F-Secure, Debian, Mandrake, SuSE, and Redhat all provide fixes for this vulnerability. The priority is listed as medium due to the fact that I am not sure what the Linux systems are used for and I do not have system log files. If they are mission critical/important then this issue should be addressed as there is a small possibility that nodes are infected.

Alert #7:

Alert Signature: Exploit x86 NOOP
Alerts Generated: 3711
Source Addresses Involved: 275
Destination Addresses Involved: 112
Priority: Low

Sample Alert Data for Alert #7

11/13-19:19:02.648272 [**] EXPLOIT x86 NOOP [**] 68.107.188.244:3332 -> 130.85.5.20:80
11/13-19:19:02.690824 [**] EXPLOIT x86 NOOP [**] 68.107.188.244:3332 -> 130.85.5.20:80
11/13-19:19:02.925099 [**] EXPLOIT x86 NOOP [**] 68.107.188.244:3332 -> 130.85.5.20:80

Search results for: 68.107.188.244

```
Cox Communications Inc. NETBLK-RI-RDC-68-107-188-0 (NET-68-107-188-0-1)
                                  68.107.188.0 - 68.107.191.255
Cox Communications Inc. COX-ATLANTA-2 (NET-68-96-0-0-1)
                                  68.96.0.0 - 68.111.255.255
```

All of the “EXPLOIT x86 NOOP” traffic is related to port 80. This is normal Internet related traffic. The reason for triggering the alert is that some Internet traffic contains NOOP padding that looks to an IDS to be a NOOP sled. The x86 NOOP triggers primarily on | 43 43 43 43 |, | 61 61 61 61 |, or | 90 90 90 90 | patterns within the packet. Bassett states that Internet downloads containing binaries or image files contain the 0x90 0x90 0x90 0x90 padding and cause many false positives (Bassett, 2003).

Recommendation: Write the appropriate pass rules for the IDS so the traffic does not alert. This can be a benefit if paging is enabled as the NOOP is generally a priority one so the analyst receives a page for every occurrence. Be particularly careful to not write a generalized pass rule. Be as specific as possible about the traffic to be passed. The IDS still needs to alert appropriate personnel if malicious shellcode activity is seen.

Alert #8:

Alert Signature: 130.85.30.3 activity
Alerts Generated: 3695
Source Addresses Involved: 78
Destination Addresses Involved: 1
Priority: Low

Sample Alert Data for Alert #8

11/11-00:19:10.683702 [**] 130.85.30.3 activity [**] 68.55.233.51:63637 -> 130.85.30.3:524
11/11-00:19:10.919782 [**] 130.85.30.3 activity [**] 68.55.233.51:63637 -> 130.85.30.3:524
11/11-00:20:12.418124 [**] 130.85.30.3 activity [**] 68.55.233.51:63637 -> 130.85.30.3:524

130.85.30.3 is a web server that appears to be running NetWare. Upon connecting to the web server Novell services are offered. The many connections to port 524 are for the most part accessing NetWare Core Protocol (NCP), which handles client server requests. This would help explain the many connections from a considerable number of outside nodes.

Recommendation: Tighten up the rule in Snort as this traffic would be considered normal. The alert was given a low rating due to the fact that traffic is web related.

Alert #9:

Alert Signature: SUNRPC Highport Access!
Alerts Generated: 1363
Source Addresses Involved: 35
Destination Addresses Involved: 89
Priority: Medium

Sample Alert Data for Alert #9

11/15-13:31:19.408190 [**] SUNRPC highport access! [**] 24.103.156.16:3246 -> 130.85.5.13:32771
11/15-13:31:19.942883 [**] SUNRPC highport access! [**] 24.103.156.16:3246 -> 130.85.5.13:32771
11/15-13:31:20.447131 [**] SUNRPC highport access! [**] 24.103.156.16:3246 -> 130.85.5.13:32771
11/15-13:31:53.326980 [**] External RPC call [**] 24.103.156.16:3262 -> 130.85.6.15:111
11/15-13:31:53.392500 [**] External RPC call [**] 24.103.156.16:3262 -> 130.85.6.15:111
11/15-13:31:53.404598 [**] External RPC call [**] 24.103.156.16:3262 -> 130.85.6.15:111
11/15-13:31:55.480737 [**] SUNRPC highport access! [**] 24.103.156.16:3283 -> 130.85.6.15:32771
11/15-13:31:55.564851 [**] SUNRPC highport access! [**] 24.103.156.16:3283 -> 130.85.6.15:32771
11/15-13:42:32.094242 [**] SUNRPC highport access! [**] 24.103.156.16:3376 -> 130.85.24.20:32771

A search using ARIN, as shown below, indicates that the top activity node of 24.103.156.16 is not spoofed. However it may be used as a drone under an attacker's control. Since the scan activity and the exploit attempts come from the same address, the actual address would appear to be the attack node and not be performed from a drone.

Search results for: 24.103.156.16

```
Rogers Cable Inc. ROGERS-CAB-6 (NET-24-100-0-0-1)
                                  24.100.0.0 - 24.103.255.255
Rogers Cable Inc. ROGERS-DOC-2 (NET-24-103-140-0-2)
                                  24.103.140.0 - 24.103.161.255
Rogers Cable Inc. Lndn ON-ROG-18-LNDN-12 (NET-24-103-156-0-1)
                                  24.103.156.0 - 24.103.159.255
```

```
11/13-03:13:54.748364 [**] High port 65535 tcp - possible Red Worm - traffic [**]
66.93.54.236:50449 -> 130.85.6.63:65535
11/13-03:13:54.758299 [**] High port 65535 tcp - possible Red Worm - traffic [**]
66.93.54.236:50449 -> 130.85.6.63:65535
11/13-03:13:56.029993 [**] SUNRPC highport access! [**] 66.93.54.236:50453
-> 130.85.6.63:32771
11/13-03:13:56.053005 [**] SUNRPC highport access! [**] 66.93.54.236:50453
-> 130.85.6.63:32771
11/13-03:14:07.004886 [**] FTP passwd attempt [**] 66.93.54.236:49955 ->
130.85.6.63:21
```

Search results for: 66.93.54.236

```
Speakeasy Network SPEAKEASY-5 (NET-66-92-0-0-1)
                                  66.92.0.0 - 66.93.255.255
BLT BRIDGED CIRCUITS SPEK-BLT-BR-1 (NET-66-93-54-1-1)
                                  66.93.54.1 - 66.93.54.255
```

The vast majority of activity with the SUNRPC activity alert is normal web server traffic on port 80. The majority of secondary traffic is mail server related. The two sets of interesting alerts come from the alert captures shown above. First, 24.103.156.16 has many triggers; 101 alerts from 13:31:19 to 15:23:18 on 11/15/2003. Throughout the 101 alerts, the source port iterates steadily up as if multiple attempts are being made. Then as shown in the alert logs above, an RPC attempt on port 111 is tried. This could be valid traffic, but it is difficult to tell without a packet capture. Further it is difficult to tell if the attempts were successful.

The second set of alerts show 66.93.54.236 with possible Red Worm infection attempting to contact 130.85.6.63. It is followed by an attempt on the SUNRPC high port, which is then followed by an FTP attempt. Again, it is difficult to know whether these attempts were successful without server log files.

The attack is implemented by attempting to access rpcbind or rpcportmapper on a standard udp port 111 or tcp port 111. If this fails, then the attacker can move to udp ports greater than 32770 in hopes of finding the rpcbind process on a higher port (Friedrichs, 1997).

The scan data below indicates that there was significant reconnaissance activity happening just prior to the attempts. So it can be stated with a high level of confidence that the activity from source address 24.103.156.16 was attempting to locate vulnerabilities and then exploit them. No scan data was located for 66.93.54.236. Table 3.5 is shortened from 1213 scans to 34 scans for brevity. The table shows the start and end times of the scan activity for 24.103.156.16

Table 3.5 Scan data for 24.103.156.16

Month	Day	Time	Source IP	Destination IP	Type of Scan	Flags
Nov	15	14:09:29	24.103.156.16:3532	130.85.70.1:22	SYN	******S*
Nov	15	14:09:29	24.103.156.16:3541	130.85.70.1:80	SYN	******S*
Nov	15	14:09:29	24.103.156.16:3533	130.85.70.1:23	SYN	******S*
Nov	15	14:09:30	24.103.156.16:3545	130.85.70.1:25	SYN	******S*
Nov	15	14:09:29	24.103.156.16:3548	130.85.70.1:110	SYN	******S*
Nov	15	14:09:31	24.103.156.16:3558	130.85.70.1:143	SYN	******S*
Nov	15	14:09:30	24.103.156.16:3543	130.85.70.1:109	SYN	******S*
Nov	15	14:09:31	24.103.156.16:3590	130.85.70.1:1080	SYN	******S*
Nov	15	14:09:31	24.103.156.16:3598	130.85.70.1:2000	SYN	******S*
Nov	15	14:09:31	24.103.156.16:3602	130.85.70.1:3306	SYN	******S*
Nov	15	14:09:31	24.103.156.16:3613	130.85.70.1:5000	SYN	******S*
Nov	15	14:09:31	24.103.156.16:3635	130.85.70.1:8080	SYN	******S*
Nov	15	14:09:30	24.103.156.16:3544	130.85.70.5:25	SYN	******S*
Nov	15	14:09:30	24.103.156.16:3546	130.85.70.5:42	SYN	******S*
Nov	15	14:09:30	24.103.156.16:3549	130.85.70.5:53	SYN	******S*
Nov	15	15:23:15	24.103.156.16:2926	130.85.190.202:25	SYN	******S*
Nov	15	15:23:15	24.103.156.16:2938	130.85.190.202:109	SYN	******S*
Nov	15	15:23:15	24.103.156.16:2923	130.85.190.202:22	SYN	******S*
Nov	15	15:23:15	24.103.156.16:2924	130.85.190.202:23	SYN	******S*
Nov	15	15:23:15	24.103.156.16:2949	130.85.190.202:143	SYN	******S*
Nov	15	15:23:15	24.103.156.16:2942	130.85.190.202:110	SYN	******S*
Nov	15	15:23:15	24.103.156.16:2943	130.85.190.202:111	SYN	******S*
Nov	15	15:23:16	24.103.156.16:2955	130.85.190.202:443	SYN	******S*
Nov	15	15:23:15	24.103.156.16:2964	130.85.190.202:514	SYN	******S*
Nov	15	15:23:16	24.103.156.16:2968	130.85.190.202:1080	SYN	******S*
Nov	15	15:23:16	24.103.156.16:2978	130.85.190.202:3389	SYN	******S*
Nov	15	15:23:16	24.103.156.16:2979	130.85.190.202:5000	SYN	******S*
Nov	15	15:23:16	24.103.156.16:2972	130.85.190.202:2000	SYN	******S*
Nov	15	15:23:14	24.103.156.16:2931	130.85.190.203:22	SYN	******S*

Nov	15	15:23:15	24.103.156.16:2937	130.85.190.203:25	SYN	******S*
Nov	15	15:23:18	24.103.156.16:2994	130.85.190.203:32771	SYN	******S*
Nov	15	15:23:17	24.103.156.16:2991	130.85.190.203:5800	SYN	******S*
Nov	15	15:23:17	24.103.156.16:2992	130.85.190.203:6000	SYN	******S*
Nov	15	15:23:17	24.103.156.16:2983	130.85.190.203:3306	SYN	******S*

Recommendation: Unless required for business purposes, block all RPC attempts from outside the local network. Make sure that current security patches are applied to all UNIX equipment. This issue was given a medium priority due to the fact that evidence of actual ftp and RPC attempts were shown in the alert logs. While an attempt is not in and of itself evidence that damage was done, it indicates that some perceived vulnerability must exist in order for the attacker to try. This box should also be checked for the any signs of Adore/Red Worm as rpcbind is one element the virus exploits. The node is also accepting traffic on port 65535, which is a listening port for Adore.

Alert #10:

Alert Signature: connect to 515 from outside
Alerts Generated: 747
Source Addresses Involved: 3
Destination Addresses Involved: 113
Priority: Low

1. Sample Alert Data for Alert #10

```
11/11-23:23:24.173500 [**] connect to 515 from outside [**]
68.32.127.158:49357 -> 130.85.24.15:515
```

```
11/11-23:23:24.180098 [**] connect to 515 from outside [**]
68.32.127.158:49357 -> 130.85.24.15:515
```

```
11/11-23:23:24.191450 [**] connect to 515 from outside [**]
68.32.127.158:49357 -> 130.85.24.15:515
```

2. Sample Alert Data for Alert #10

```
11/13-09:27:57.849174 [**] connect to 515 from outside [**]
195.227.113.150:1195 -> 130.85.5.5:515
```

```
11/13-09:28:28.866914 [**] connect to 515 from outside [**]
195.227.113.150:1193 -> 130.85.16.90:515
```

```
11/13-09:28:50.382636 [**] connect to 515 from outside [**]
195.227.113.150:4218 -> 130.85.24.15:515
```

3. Sample Alert Data for Alert #10

```
11/15-15:23:17.452085 [**] connect to 515 from outside [**] 24.103.156.16:2975
-> 130.85.190.203:515
```

Three sources exist in the alert logs for outside connection to port 515. Port 515 is the print spooler port and has the potential to be exploited. One exploit is to send a large amount of trash characters to an HP LPR print spooler on tcp port 515. The overflow then causes the printer to enter a DoS, which disallows new print jobs (Bock, 2000). A second exploit is to locate a Linux server generally 3.6.x and older, where the LPR daemon can be exploited via a buffer overflow in use_syslog(). This will yield root access to the node either locally or through a remote attack (CVE-2000-0917).

The first set of alerts above from 68.32.127.158 seem like normal connections to the printer pool. The second set of connections from 195.227.113.150 iterate through an increasing range of source ports as if it were making successive connections back to back. A search of the scan logs on November 13th proves that 195.227.113.150 was in deed scanning the subnet 190 range of addresses as shown in table 3.5. Not all 190 range addresses resolve to a fully qualified name, so it appears that the scan was scripted to scan for all addresses in the 190 range. No logs are available to correlate issues that came from this scan. Some addresses resolve to names like pooled190-XXXXX, which could be printer pools by the naming convention. A search of RIPE below shows that the address is a valid address in Europe. Using nslookup, the address does not resolve to a known good fully qualified address.

Table 3.5 Scan logs for 195.227.113.150

Month	Day	Time	Source IP	Destination IP	Type of Scan	Flags
Nov	13	9:36:24	195.227.113.150:2883	130.85.190.5:515	SYN	******S*
Nov	13	9:36:24	195.227.113.150:2886	130.85.190.8:515	SYN	******S*
Nov	13	9:36:24	195.227.113.150:2888	130.85.190.10:515	SYN	******S*
Nov	13	9:36:24	195.227.113.150:2890	130.85.190.12:515	SYN	******S*
Nov	13	9:36:24	195.227.113.150:2892	130.85.190.14:515	SYN	******S*
Nov	13	9:36:24	195.227.113.150:2894	130.85.190.16:515	SYN	******S*
Nov	13	9:36:24	195.227.113.150:2895	130.85.190.17:515	SYN	******S*
Nov	13	9:36:24	195.227.113.150:2896	130.85.190.18:515	SYN	******S*
Nov	13	9:36:24	195.227.113.150:2898	130.85.190.20:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2925	130.85.190.47:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2926	130.85.190.48:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2928	130.85.190.50:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2930	130.85.190.52:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2931	130.85.190.53:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2932	130.85.190.54:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2879	130.85.190.1:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2933	130.85.190.55:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2934	130.85.190.56:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2935	130.85.190.57:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2937	130.85.190.59:515	SYN	******S*
Nov	13	9:36:25	195.227.113.150:2938	130.85.190.60:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3026	130.85.190.147:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3022	130.85.190.143:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3023	130.85.190.144:515	SYN	******S*

Nov	13	9:36:29	195.227.113.150:3024	130.85.190.145:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3034	130.85.190.155:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3031	130.85.190.152:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3036	130.85.190.157:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3032	130.85.190.153:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3033	130.85.190.154:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3042	130.85.190.163:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3084	130.85.190.204:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3080	130.85.190.200:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3081	130.85.190.201:515	SYN	******S*
Nov	13	9:36:29	195.227.113.150:3077	130.85.190.197:515	SYN	******S*
Nov	13	9:36:30	195.227.113.150:3119	130.85.190.239:515	SYN	******S*
Nov	13	9:36:30	195.227.113.150:3115	130.85.190.235:515	SYN	******S*
Nov	13	9:36:30	195.227.113.150:3120	130.85.190.240:515	SYN	******S*
Nov	13	9:36:30	195.227.113.150:3116	130.85.190.236:515	SYN	******S*
Nov	13	9:36:30	195.227.113.150:3121	130.85.190.241:515	SYN	******S*
Nov	13	9:36:30	195.227.113.150:3117	130.85.190.237:515	SYN	******S*
Nov	13	9:36:30	195.227.113.150:3118	130.85.190.238:515	SYN	******S*
Nov	13	9:36:30	195.227.113.150:3122	130.85.190.242:515	SYN	******S*

Search on 195.227.113.150

inetnum: 195.227.113.128 - 195.227.113.191
netname: WEKO-BUEROMOEBEL
descr: Weko Bueromoebelfabrik Wessel GmbH
descr: Koeln, Germany
country: DE
admin-c: MA1713-RIPE
tech-c: TC1997-RIPE
status: ASSIGNED PA
mnt-by: NDH-MNT
changed: tc@ndh.net 20001030
source: RIPE
route: 195.227.0.0/16
descr: DE-PIRONET-971119
origin: AS8469
mnt-by: PIRONET-MNT
changed: nw@pironet-ndh.com 20030522
source: RIPE
person: Markus Aschke
address: Weko Bueromoebelfabrik Wessel GmbH
address: Melatengurtel 131
address: D-50825 Koeln
address: GERMANY
phone: +49-221-9544900
fax-no: +49-221-95449038
e-mail: nw@pironet-ndh.com
nic-hdl: MA1713-RIPE
changed: ebastuz@netcologne.de 19990106
changed: ebastuz@netcologne.de 20030829
source: RIPE
person: Thorsten Clever

```
address:        NDH IT Service AG
address:        Theodor-Heuss-Strasse 92-100
address:        D-51149 Koeln
address:        DE
phone:          +49 2203 93530 0
fax-no:         +49 2203 93530 99
nic-hdl:        TC1997-RIPE
mnt-by:         NDH-MNT
changed:        kc@ndh.net 20001026
source:         RIPE
```

The third address of 24.103.156.16 is from the scan sweep performed and discussed in Alert #9.

Recommendation: Unless there are valid business reasons to allow printing from outside the local network, port 515 should be filtered on the firewall or ingress ACL on the boundary router. All Linux and UNIX systems should have current and up-to-date patch sets. All HP printers should have current and up-to-date firmware.

Review of Interesting Alerts

This section will review alerts that may be an indication of issues, but did not make the top ten alerts section, which makes up greater than 97% of the activity.

IRC Activity

There is a considerable amount of IRC related activity on campus and these systems should be reviewed. The affected nodes could be infected and managed from outside the home network or could be managed from within the home network. Most activity is seen from nodes on the home network accessing nodes on the external network. Much of the activity centers on potentially compromised nodes acting as XDCC bots. These systems act as file share "bots", advertising their files on an IRC server (Martin, 2003). Table 3.6 shows source nodes participating in IRC activity.

Tablet 3.6 Nodes Acting as Potential Bots

130.85.82.79	130.85.42.8	130.85.97.42	130.85.97.16	130.85.97.78
130.85.81.18	130.85.42.9	130.85.97.80	130.85.42.1	130.85.82.79
130.85.15.198	130.85.80.16			

The sources and destinations in tables 3.7 and 3.8 are the top five participants that triggered an alert for “IRC user/kill detected, possible Trojan”. According to Martin, the “’ERROR:Closing Link’” is responsible for closing or killing IRC connections, which fires the kill detected alert (et al.). The nodes participating in this activity should be reviewed for signs of compromise.

Table 3.7 Top Five Sources of IRC Kill Detected

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
64.157.246.22	177	177	1	1
64.157.246.24	133	133	1	1
65.57.234.3	62	62	1	1
69.36.232.118	18	18	4	4
193.201.71.39	13	13	1	1

Table 3.8 Top Destinations of IRC Kill Detected

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
130.85.15.198	177	177	1	1
130.85.81.18	133	134	1	2
130.85.60.39	67	68	5	6
130.85.60.40	22	22	5	5
130.85.71.243	17	17	4	4

Possible Trojan Activity

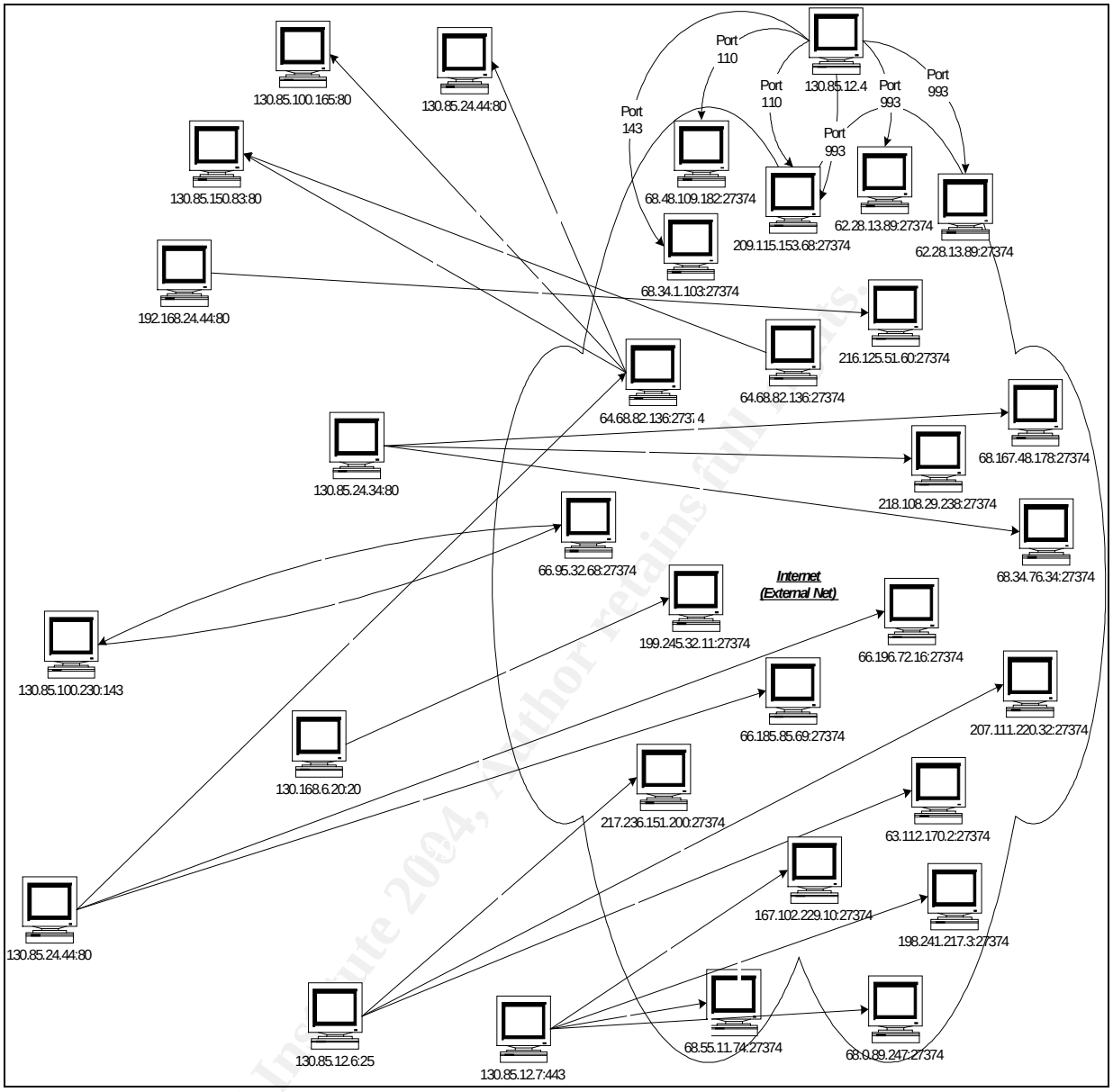
Alert Signature: Possible Trojan Activity
Alerts Generated: 378
Priority: Medium

Some of the source and destination nodes participating in the Trojan activity are the same as the Red Worm activity discussed earlier. This section will discuss additional findings captured under this alert.

- Traffic related to 130.85.12.4, 130.85.100.230, and 130.85.83.109 is mail traffic – note ports 25, 110, 143, and 993, 3264 which are SMTP, mailbox protocol, IMAP, IMAP4, and cc:Mail/Lotus respectively.
- Traffic related to 130.85.30.4, 130.85.100.165, 130.85.150.83, 130.85.24.44, 130.85.12.7, 130.85.10.83, 130.85.24.34, and 130.85.24.44 and on ports, 80, 443 HTTP, HTTP over TLS/SSL is web server traffic.
- Traffic on 130.85.6.20:20 is File Transfer [Default Data] related.
- 130.85.112.222:9591 -> 66.90.79.36:27374 could be legitimate Trojan traffic as well as 24.206.144.65:27374 -> 130.85.97.68:6346, which could be a gnutella-svc related connection.

Of all Trojan activity, the two connections in the last bullet are the only alerts that look suspicious due to a high ephemeral port connecting to another high ephemeral port. To help depict the web traffic, the link diagram in figure 3.1 shows the relationship of the top ten source IP's and their connection partners.

Figure 3.1 Potential Trojan Activity From Top Ten Source IP's



Most activity with the exception of the two nodes noted above are related to mail or web traffic and the random connection on a port that triggers the Trojan alert. As can be seen from figure 3.1, most activity originates in the local network and connects to hosts on the Internet. The traffic could be response activity based on a stimulus that originated on remote sources. This is likely the case since most activity is web and mail related. Users are probably accessing data remotely.

Recommendation: since I had no packet captures or system logs to verify actual traffic or connections, the nodes should be checked for any malware. A good idea would be to maintain current Anti-virus (AV) software and monitor it regularly.

External RPC Traffic

Alert Signature: External RPC Call
Alerts Generated: 254
Priority: Medium

A considerable amount of scan activity is seen for port 111, which is a SUN rpc port. The greatest volume is seen against subnet 190, but subnet 6 and 16 is also being scanned. On 11/15/2003, 199.186.199.35 scanned subnet 190 for port 111, which alerted on 157 scans. 199.186.199.35 is the number one source for port 111 scan activity and according to ARIN it belongs to AT&T Bell Laboratories. According to Reese, there are numerous vulnerabilities associated with rpc on UNIX systems (Reese, 2000).

Recommendation: Unless required for business purposes, block all RPC attempts from outside the local network. Make sure that current security patches are applied to all UNIX equipment. This issue was given a medium priority due to the fact that system logs are not available to check for successful connection attempts.

FTP Password Attempt

Alert Signature: FTP Password Attempt
Alerts Generated: 109
Priority: Low

A considerable amount of scan activity is seen for port 21, with 10722 scans from 172.177.207.191, which is an America on Line address according to ARIN. 172.177.207.191 is the largest contributor to the FTP activity for this alert. If FTP services are to be offered, this type of activity can be expected. While undesirable, outsiders are looking for an easy victim. Make sure port 21 is only open on nodes running FTP services.

Recommendation: Consider creating a demilitarized zone (DMZ) if one does not all ready exist in the environment. Place all services that can be accessed from external networks in the DMZ. This can include FTP services, web services, and anything else your external users require. This will limit the exposure of internal systems by allowing the external user no deeper than the DMZ.

Scan Data

The scan data shown in table 3.8 was developed by sorting scan logs from the 11th through the 15th. It was sorted based on the type of scan and the volume of scans. The data is presented in descending order.

Table 3.8 Scan Types

# Scans	Type	Name
10821332	******S*	SYN
2931237		UDP
6114	12****S*	SYN
799	*******F	FIN
470	***A*R*F	INVALIDACK
217	********	NULL
137	*2*A**S*	UNKNOWN
95	1**A*R**	UNKNOWN
81	1****R**	UNKNOWN
43	*2*A****	UNKNOWN
38	***APR*F	INVALIDACK
27	*2***R**	UNKNOWN
20	12***R**	UNKNOWN
18	**U**RS*	NOACK
16	*****RS*	NOACK
8	12UA**SF	INVALIDACK
8	**U*P*S*	NOACK
8	**U**RSF	NOACK
7	**U*P*SF	NMAPID
7	***A**SF	INVALIDACK
6	**U*P***	VECNA
5	**U*P**F	XMAS
4	12**P***	VECNA
4	*2UA**SF	INVALIDACK
4	**U*PRS*	NOACK
4	*****R*F	NOACK

The scan data was used throughout the earlier analysis to provide support for conclusions that were drawn from the alert files.

Top Talkers

The top talkers are included in tables 3.9 and 3.10 and were developed by running SnortSnarf. Included are the top 10 source IP addresses and the top 10 destination addresses. The break down includes the number of total alerts that the address was involved in, the total number of signatures the node triggered, and the number of destinations or sources involved.

Table 3.9 Top 10 Source IP's

Rank	Total # Alerts	**Source IP**	# Signatures triggered	Destinations involved
rank #1	92838 alerts	**67.21.63.15**	104 signatures	(3 destination IPs)
rank #2	9718 alerts	**130.85.80.51**	9 signatures	(9718 destination IPs)
rank #3	5171 alerts	**130.85.162.41**	1 signatures	128.183.110.242
rank #4	3617 alerts	**68.81.2.19**	2 signatures	130.85.30.3, 130.85.30.4
rank #5	1416 alerts	**130.85.21.37**	1 signatures	(5 destination IPs)
rank #6	1322 alerts	**129.165.254.6**	2 signatures	130.85.162.56, 130.85.72.178
rank #7	1298 alerts	**130.85.21.67**	1 signatures	(8 destination IPs)
rank #8	1193 alerts	**68.55.205.180**	1 signatures	130.85.30.4
rank #9	1190 alerts	**68.57.90.146**	2 signatures	130.85.30.3, 130.85.30.4
rank #10	1090 alerts	**130.85.21.69**	1 signatures	(8 destination IPs)

Table 3.10 Top 10 Destination IP's

Rank	Total # Alerts	**Destination IP**	# Signatures triggered	Originating sources
rank #1	104431 alerts	**130.85.30.4**	102 signatures	(287 source IPs)
rank #2	5171 alerts	**128.183.110.242**	1 signatures	130.85.162.41
rank #3	3722 alerts	**130.85.70.176**	6 signatures	(118 source IPs)
rank #4	3695 alerts	**130.85.30.3**	1 signatures	(78 source IPs)
rank #5	2568 alerts	**66.68.195.62**	2 signatures	(6 source IPs)
rank #6	2000 alerts	**66.65.253.92**	1 signatures	(6 source IPs)
rank #7	1321 alerts	**130.85.162.56**	2 signatures	129.165.254.6
rank #8	991 alerts	**169.254.0.0**	1 signatures	(3 source IPs)
rank #9	878 alerts	**81.15.213.23**	1 signatures	130.85.97.64
rank #10	798 alerts	**130.85.15.71**	4 signatures	(4 source IPs)

Table 3.11 and Chart 3.1 provide an overview of the top ten scanned destination IP addresses. Note that all scan destinations are to nodes within the universities local network. Table 3.12 shows the top ten scanned destination ports with a description of the ports primary use

Table 3.11 Top Ten Scanned Destination IP's

Destination IP	# Scans	Service	Name
130.85.1.3	2792448	Web Server	umdc3.umbc.edu
130.85.70.129	2606265	Mail Server	ecs128pc01.umbc.edu
130.85.111.72	1750842	Web Server	cuereims.umbc.edu
130.85.163.107	1524789		physics105pc-01.umbc.edu
130.85.162.92	1522724		oneill-1.umbc.edu
130.85.84.194	1437878		engr-84-194.pooled.umbc.edu
130.85.1.4	335185		umdc4.umbc.edu
130.85.153.37	210998		refweb08.libpub.umbc.edu
130.85.53.41	160303	Mail Server	ecs021pc11.ucslab.umbc.edu

Chart 3.1 Top Ten Scan Destinations By IP

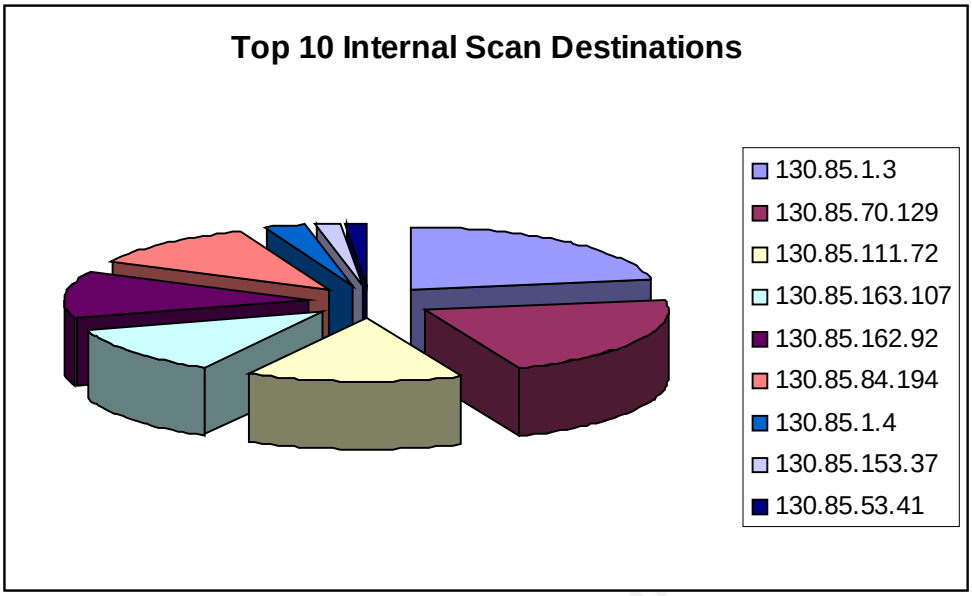


Table 3.12 Top Ten Destination Ports scanned

Dst Port #	# of Scans	Port usage
135	8646202	Microsoft Windows RPC default port - DCE endpoint resolution
53	3111642	DNS
80	661730	HTTP
21	147303	FTP
4000	104424	Asp module for Apache servers - Used to transfer file when using iMesh
6346	88020	Telnet (binary mode) mirror of 2001-2999 range - P2P file-sharing
25	63545	Simple Mail Transfer
554	32783	Real Time Streaming Protocol
1257	29715	Shockwave 2

Final Recommendation

Probably the single most important recommendation is to tighten Snort rules to alert on actual threats. In some cases this is being done, but it is washed out by the shear number of false positives. A team of analysts would have great difficulty sorting through this data and staying current with it. The “cry wolf” scenario will make the analyst numb to a real alert.

An AV program must be implemented, if it is not currently running, to maintain and keep watch for known viruses. The program must include updates to AV engines and dat files on a regular basis. Centralized monitoring of all critical

system AV scans should be watched daily. This will provide feedback and trending on virus detection on the campus.

Perimeter measures must be installed such as firewalls and ACLs on boundary routers. This will help lock down the perimeter. At this point the IDS should be used to monitor what gets through the ACLs and the firewall. CIP methodology must then be applied to the ACLs and firewall rules to maintain the defense in depth model. The IDS should be used as the last line of defense in a layered security model and should validate that nothing gets through the upper layers of the defense.

Finally, review the IDS location for optimal enterprise monitoring. For most applications, the IDS can be located at choke points behind the boundary router or firewall. A second location is to monitor the activity of critical servers. It would appear due to the massive alerts captured that a tuning problem exists, but a review of the IDS location may help.

Process Used for Analysis

- Three sets of five log files were downloaded from incidents.org/logs. The entire set contained:
    - Five Alert files ranging from November 11, 2003 – November 15, 2003, which contain data collected from the University Snort IDS
    - Five Scan files ranging from November 11, 2003 – November 15, 2003, which contain data collected from the Snort portscan pre-processor
    - Five OOS files ranging from November 11, 2003 – November 15, 2003, which contain data of packets that do not meet RFC standards

- Server used for data crunching:
    - Home built server including 1.8 ghz Intel processor, with 1 gig of RAM, and running Linux 9
    - SnortSnarf was installed and its operation was verified with at test sample of the alert files

- The alert files were prepared for SnortSnarf by removing the portscan data as it is redundant to the Scan logs that were downloaded
    - Simple command used: grep –iv portscan alert.03111* >> *<newfile>*
    - All five files minus the portscan data was now located in one consolidated file that SnortSnarf could run against.
    - A second preparation step was to modify the files by changing “MY.NET” to “192.168”. Some earlier papers stated that SnortSnarf was unable to deal with “MY.NET”. I did not try, going straight for the modification. I did not review the scan logs prior to modify the

alert logs with “192.168” or I would have gone straight to the known University address of “130.85”. The following command was used on the consolidated alert file to globally change “MY.NET” to “192.168”
    - perl –p –i –e s/MY.NET/192.168/g <*filename*>

- The modified logs were then fed to SnortSnarf using the following command:
    - ./snortsnarf.pl –rulesfile /etc/snort –d <*www output*> <*consolidated alert file*>

  Note: 152471 alerts took less than 20 hours as I was not there when it stopped

  Note: Your web server of choice must be running to serve the output. I used Apache on Linux
    - SnortSnarf consolidates all alert files to a single webpage to include a roll up of the alert data, a top 20 source list, and a top 20 destination list. It also provides links to perform port and IP address look up directly from the web page.

- The scan file analysis was done using custom csh scripts containing grep, awk, sed, perl, and sort commands. I would have preferred to put this data in a database, but my shell scripting ability using flat files was there and my database skills are long gone. I found shell scripting the data to be effective but cumbersome at times. It was also somewhat taxing on the hardware resources at times.

References for Section 3:

arachnids Database. “IDS177 "*NETBIOS-NAME-QUERY*"”. URL: http://whitehats.com/info/IDS177. (06 December 2003).

Balasubramaniam Manjunath. **“***IP Address Management Framework: Managing Application IP Address/ Port Configurations in NetWare 6.5.”* URL: http://216.239.57.104/search?q=cache:c-vhGxhhSFUJ:developer.novell.com/research/appnotes/2003/septembe/02/a030902.pdf. (23 November 2003).

Bassett, Greg. “Intrusion Detection: An Inside Look”. 21 September 2003. URL: http://www.giac.org/practical/GCIA/Greg_Bassett_GCIA.pdf. (14 December 2003).

Bock, John. “HP printers vulnerable to remote DoS (spooler port)”. 26 April 2000. URL: http://www.securiteam.com/exploits/5DQ0G000JA.html. (08 December 2003).

Common Vulnerabilities and Exposures. “CAN-2000-0917.” URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0917. (08 December 2003).

Common Vulnerabilities and Exposures. “CAN-2001-0550.” URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name-CAN-2001-0550. (23 November 2003).

Cormier, Andre. “Intrusion Prevention and the Quest for the Holy Grail”. (05 February 2003). http://www.giac.org/practical/GCIA/Andre_Cormier_GCIA.pdf. (08 December 2003).

Cheshire, Stuart. “Dynamic Configuration of Link-Local IPv4 Addresses.” 29 September 2003. URL: http://files.zeroconf.org/draft-ietf-zeroconf-ipv4-linklocal.txt. (23 November 2003).

Friedrichs, Oliver. “Solaris rpcbind listens on undocumented high UDP port”. 4 June 1997. URL: http://www.insecure.org/sploits/solaris.rcpbind.high-udp-port-listen.html. (07 December 2003).

Maher, James. “Intrusion Detection In Depth”. 16 July 2003. URL: http://www.giac.org/practical/GCIA/James_Maher_GCIA.pdf. (07 December 2003).

Martin, Ian. “SANS GCIA Practical Version 3.3”. 17 July 2003. URL: http://www.giac.org/practical/GCIA/Ian_Martin_GCIA.pdf (07 December 2003).

McLaughlin III L. “Line Printer Daemon Protocol”. August 1990. URL: http://www.ietf.org/rfc/rfc1179.txt. (23 November 2003).

Novell AppNotes. “Answers to your technical questions”. December 2001. URL: http://developer.novell.com/research/sections/netsupport/abend/2001/december/x011201.htm. (23 November 2003).

Reese, David**. “**Is blocking port 111 sufficient to protect your systems from RPC attacks?”. 26 February 2000. URL: http://www.sans.org/resources/idfaq/blocking.php. (17 December 2003).

RFC 3330. “RFC 3330 – Special-Use IPv4 Addresses.” URL: http://www.faqs.org/rfcs/rfc3330.html. (23 November 2003).

Roesch, Marty. “Re: [Snort-users] Incomplete Packet Fragments Discarded” November 2001. URL: http://marc.theaimsgroup.com/?l=snort-users&m=100681596629407&w=2. (06 December 2003).

WinMX. “Working Around ISP Port Blocks”. URL: http://winmx.2038.net/winmx/fr-blocked.html. (06 December 2003).