Global Information Assurance Certification Paper

Intrusion Detection In Depth

GCIA Practical Assignment

Version 3.0

David Stewart

Assignment 1

Carnivore

Is Carnivore something that people should be scared of? Is their privacy in jeopardy? People are scared of it because of the mystery surrounding it. But when one pulls back the curtain, one will see something very different than what was expected by the public.

According to the FBI: "Carnivore is a computer-based system that is designed to allow the FBI, in cooperation with an Internet Service Provider (ISP), to comply with court orders requiring the collection of certain information about emails or other electronic communications to or from a specific user targeted in an investigation."

The FBI's website (http://www.fbi.gov/hq/lab/carnivore/carnivore2.htm) claims that Carnivore is nothing more than a "diagnostic tool".

> The Carnivore device works much like commercial "sniffers" and other network diagnostic tools used by ISPs every day, except that it provides the FBI with a unique ability to distinguish between communications, which may be lawfully intercepted, and those that may not.

We do know that Carnivore is a packet sniffer, a technology that is quite common and has been around for some time. A packet sniffer can see all the information going over a network that it is connected to. As the data goes back and forth through the network, the sniffer "sniffs" the packets that it sees, and filters out the packets it is interested in seeing.

Normally, a computer only sniffs out the packets that are addressed to it and ignores the rest of the traffic it sees on the network. When a packet sniffer is installed, it is set to promiscuous mode, which allows it to see all the traffic on the network.

There are two types of settings that the packet sniffer can be set to: Unfiltered and filtered. When it is set to "unfiltered", it captures all the packets that come through the network. This is not desired, especially on a network that is very busy with traffic. When the sniffer is set to "filtered", it only captures the packets that contain certain data signatures. This is the most desirable setting, considering you would not have to wade through irrelevant data that would hinder the investigation.

What we know about Carnivore mostly comes from declassified documents that were released during a lawsuit filed by the Electronic Privacy Information Center (EPIC). Taken from the EPIC website (http://www.epic.org/privacy/carnivore/), EPIC filed for a Freedom of Information Act (FOIA),

> …Request seeking the public release of all FBI records concerning Carnivore, including the source code, other technical details, and legal analyses addressing the potential privacy implications of the technology. On July 18, 2000, after Carnivore had become a major issue of public concern, EPIC asked the Justice Department to expedite the

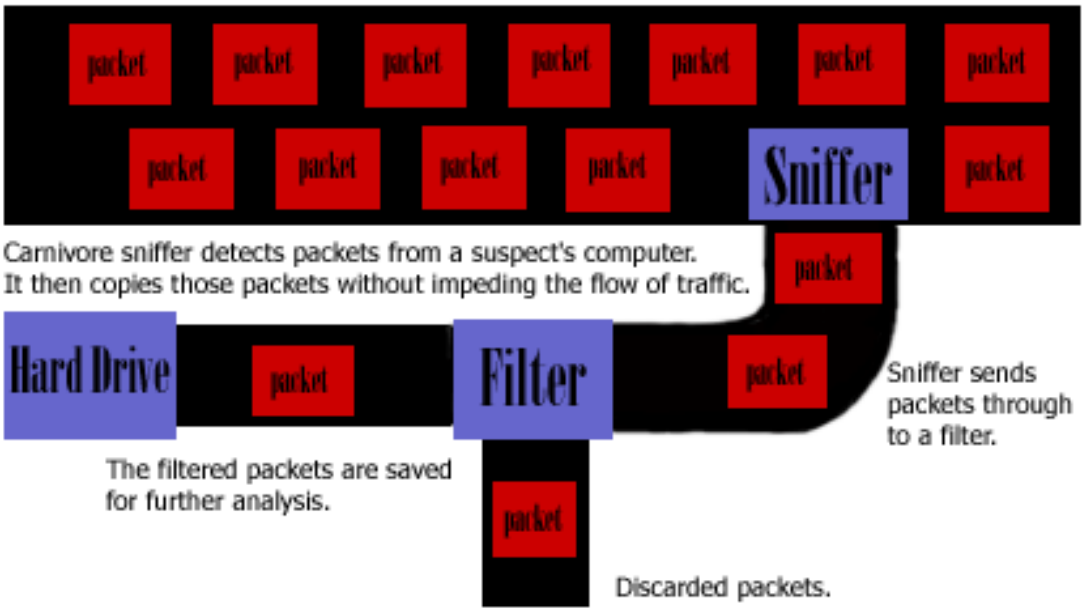
> processing of its request. When DOJ failed to respond within the statutory deadline, EPIC filed suit in U.S. District Court seeking the immediate release of all information concerning Carnivore.
>
> At an emergency hearing held on August 2, 2000, U.S. District Judge James Robertson ordered the FBI to report back to the court by August 16 and to identify the amount of material at issue and the Bureau's schedule for releasing it. The FBI subsequently reported that 3000 pages of responsive material were located, but it refused to commit to a date for the completion of processing.
>
> In late January 2001, the FBI completed its processing of EPIC's FOIA request. The Bureau revised its earlier estimate and reported that there were 1756 pages of responsive material; 1502 were released in part and 254 were withheld in their entirety.

As recently as August 9, 2001, EPIC filed a motion on the grounds that the FBI has failed to conduct an adequate search for responsive documents. This came after an August 1, 2001 move by the FBI, stating that it fulfilled its obligation under FOIA.

What does Carnivore intercept? According to http://www.robertgraham.com/pubs/carnivore-faq.html, it uses two types of methodologies: Content wiretap and trap and trace/pen register. Content wiretap captures all email messages coming to and from a given account, or it can capture all network traffic both inbound and outbound to and from a specific IP address or account. The other method is a less intrusive. Trap and trace (inbound) and pen register (outbound traffic), simply refer to the monitoring and recording of traffic to and from a site. It captures email headers, including addresses, but does not include the content wherein. The same goes for FTP traffic. It sees what websites addresses but not the content.

The reason for the two methods of interception is because for a full content wiretap, it has to be authorized by a Federal District judge. Whereas the trace/pen register and trap can be granted by a judge in a lower court. Because of this, it is harder to get a full content wiretap. This is useful, however, when they want to gather evidence for a prosecution. In contrast, a trace/pen register can be used to get background information on a suspect, but this evidence most likely cannot be used in a court of law.

Below is a diagram of how Carnivore "sniffs" packets from a network. This picture was inspired by a picture from a website called http://www.howstuffworks.com. I just reproduced the picture in Photoshop with my own look and explanation of how Carnivore "sniffs" packets.



What does the Carnivore box consist of? Carnivore is literally a "COTS" (Commercial Off The Shelf) Windows NT box, Pentium III or IV, 128-megabytes of RAM (most likely more), with a 2 GB Jazz drive to store information. There is no TCP/IP stack, so it cannot be hacked into through the net. Robert Graham, the author of http://www.robertgraham.com/pubs/carnivore-faq.html, guesses that EtherPeek is used by Carnivore to capture IP address traffic. EtherPeek is available to anyone; in addition, it is also mentioned in the FBI's declassified documents.

The Carnivore box is taken to an ISP along with a court order search warrant and information of who they need to eavesdrop on. Is all of this necessary? Not if the ISP can provide the information to the FBI through other means. The ISP can simply copy your emails and send them to the FBI. That way Carnivore never gets used. If it were used, it would probably be there no longer than a month, since a court order must be reissued every month.

Carnivore can also be circumvented rather easily. If someone feels like they might be a suspect, or if someone knows they are doing illegal activity in the first place, you can get a practically anonymous email address from Yahoo!, or some other email provider. Just make up the personal information. It is very easy to say that you did not send that email, especially if you say a Trojan has been placed on your system. Or, someone came to your terminal and was able to use your email account to send the offending emails.

Carnivore works off email. Even with all the Fourth Amendment arguments taking place, those who are paranoid should not worry. As previously mentioned, Carnivore is nothing more than a packet sniffer. It can do the same thing that any ISP can do. This is what I do on my job everyday, and I do not work for the FBI.

Since Carnivore does work off email, it is not the only mail sniffer out there, an email sniffer called mailsnarf. Mailsnarf comes in a package called dsniff. Created by Dug Song, dsniff is a

plethora of tools used for network auditing and penetration testing, that passively monitor a network for interesting data (http://monkey.org/~dugsong/dsniff/). Mailsnarf can sniff SMTP-related packets off the wire and reassemble entire email messages into a common format that popular mail clients can read in real time (http://www.ntsecurity.net/Articles/Index.cfm?ArticleID=8878). This is different from Carnivore because it can retrieve entire emails rather than just headers.

One concern that many people have against Carnivore is its use at sniffing out the emails. Does Carnivore pick out innocent emails from people who are not included in the search warrant? Does the use of Carnivore infringe on Americans Fourth Amendment Rights?

Another feature of Carnivore that has been revealed is called 'Magic Lantern'. Magic Lantern installs "keylogging" software on a suspect's machine that is capable of capturing keystrokes typed on a computer. By tracking exactly what a suspect types, critical encryption key information can be gathered, then transmitted back to the FBI. (http://www.msnbc.com/news/660096.asp?cp1=1). Most likely this would be sent to the suspect via email by a trusted source.

When the FBI released the classified documents on Carnivore last year in response to the Freedom of Information Act request filed by EPIC, there was a section called the "Enhanced Carnivore Project Plan." This is very similar to Trojan horse technology. It seems to represent itself as a type of SubSeven trojan. I am not sure why the FBI does not use SubSeven itself instead of spending whatever amount of money they did to develop Carnivore. In my opinion they just took Mailsnarf and SubSeven and turned it into what they wanted.

The goal of the Magic Lantern is to allow law enforcement to obtain passwords needed to unlock the encryption programs that the suspects might use. Rather than try and crack the encryption, the FBI will now be able to detect the passwords from the keystroke logs.

Ever since the dreadful attacks on September 11, 2001, privacy issues concerning email and personal privacy have been in the minds of everyone, including the FBI. According to a report from Wired News (http://www.wired.com/news/politics/0,1283,46747,00.html):

> "Federal police are reportedly increasing Internet surveillance after Tuesday's deadly attacks on the World Trade Center and the Pentagon."

According to this article, the FBI was going to different email companies and ISP's like Hotmail, and setting up their boxes. What is not said is whether or not the FBI had a court order to do so. An engineer said in this article:

> "… A lot of people" at other firms were quietly going along with the FBI's request. "I know that they are getting a lot of 'OKs' because they made it a point to mention that they would only be covering our core for a few days, while their 'main boxes were being set up at the Tier 1 carriers' -- scary,"

I can understand that why authorities would want to bug a phone, read suspect's email and where they surf on the internet, but it crosses the line when law enforcement alters ones PC without their knowledge. Large corporations are already using keystroke technologies on their networks

as a legal safeguard and to keep an eye on problem employees. Usually that is known to the employee that they are subject to monitoring.

As this world crisis continues, we are unsure as to what the FBI has in store for us concerning Carnivore, whose name has been changed to DCS1000. I guess that is to make it sound less threatening.

There are many websites out there that are completely against Carnivore. One of them is http://www.stopcarnivore.org/. As the name suggests, this site is dedicated to the stopping of Carnivore. It defines what Carnivore is, why it is bad for you, and what you can do to try and stop it.

In conclusion, Carnivore is a "diagnostic" tool used by the FBI to sniff out suspecting emails from a cooperating ISP. The FBI does need a court order in order to use it, but in times of terrorist attacks it seems that those procedures are thrown out the window.

There is room to be concerned about Carnivore, but it should be restated that any ISP could retrieve the same kind of information by filtering it out themselves without the help of Carnivore.

Works Cited

1) http://www.wired.com/news/technology/0,1282,37915,00.html
2) http://stopcarnivore.org/
3) http://www.infowarrior.org/articles/carnivore.html
4) http://www.sans.org/infosecFAQ/legal/carnivore.htm
5) http://www.robertgraham.com/pubs/carnivore-faq.html
6) http://www.usdoj.gov/jmd/publications/carnivore_draft_1.pdf
7) http://www.epic.org/privacy/carnivore/foia_documents.html
8) http://www.wired.com/news/business/0,1367,38618,00.html
9) http://www.wired.com/news/politics/0,1283,46747,00.html
10) http://www.computerworld.com/resources/carnivore/
11) http://www.fbi.gov/hq/lab/carnivore/carnivore2.htm
12) http://www.cdt.org/security/carnivore/000724fbi.shtml
13) http://www.cdt.org/security/carnivore/
14) http://monkey.org/~dugsong/dsniff/
15) http://www.ntsecurity.net/Articles/Index.cfm?ArticleID=8878

Assignment 2

Network Detects

Detect #1

Incoming Traffic:
17:17:38.176250 bad.guy.net.6667 > good.guy.net.50796: S 514319330:514319330(0) ack 1 win 8040 <mss 536> (DF)
17:18:11.820461 bad.guy.net.6667 > good.guy.net.50605: S 1026500771:1026500771(0) ack 1 win 8040 <mss 536> (DF)
17:19:58.617718 bad.guy.net.6667 > good.guy.net.23360: S 3916551146:3916551146(0) ack 1 win 8040 <mss 536> (DF)
17:20:36.434330 bad.guy.net.6667 > good.guy.net.8690: S 2602548537:2602548537(0) ack 1 win 8040 <mss 536> (DF)
17:21:28.192992 bad.guy.net.6667 > good.guy.net.26026: S 3146359024:3146359024(0) ack 1 win 8040 <mss 536> (DF)
17:22:11.628526 bad.guy.net.6667 > good.guy.net.62604: S 1761663679:1761663679(0) ack 1 win 8040 <mss 536> (DF)
17:22:55.150658 bad.guy.net.6667 > good.guy.net.25835: S 1153917478:1153917478(0) ack 1 win 8040 <mss 536> (DF)
17:23:38.032835 bad.guy.net.6667 > good.guy.net.28692: S 688789936:688789936(0) ack 1 win 8040 <mss 536> (DF)
17:24:09.713497 bad.guy.net.6667 > good.guy.net.24691: S 670236619:670236619(0) ack 1 win 8040 <mss 536> (DF)
17:24:50.314089 bad.guy.net.6667 > good.guy.net.3719: S 1308119622:1308119622(0) ack 1 win 8040 <mss 536> (DF)
17:25:32.173757 bad.guy.net.6667 > good.guy.net.59364: S 31786951:31786951(0) ack 1 win 8040 <mss 536> (DF)
-Sample

Outbound Traffic:
17:17:38.176362 good.guy.net.50796 > bad.guy.net.6667: R 1:1(0) win 0 (DF)
17:18:11.903162 good.guy.net.50605 > bad.guy.net.6667: R 1:1(0) win 0 (DF)
17:19:58.618945 good.guy.net.23360 > bad.guy.net.6667: R 1:1(0) win 0 (DF)
17:20:36.426784 good.guy.net.8690 > bad.guy.net.6667: R 1:1(0) win 0 (DF)
17:21:28.193442 good.guy.net.26026 > bad.guy.net.6667: R 1:1(0) win 0 (DF)
17:22:11.632090 good.guy.net.62604 > bad.guy.net.6667: R 1:1(0) win 0 (DF)
17:22:55.151198 good.guy.net.25835 > bad.guy.net.6667: R 1:1(0) win 0 (DF)
17:23:38.034713 good.guy.net.28692 > bad.guy.net.6667: R 1:1(0) win 0 (DF)
17:24:09.714007 good.guy.net.24691 > bad.guy.net.6667: R 1:1(0) win 0 (DF)
17:24:50.314584 good.guy.net.3719 > bad.guy.net.6667: R 1:1(0) win 0 (DF)
17:25:32.174264 good.guy.net.59364 > bad.guy.net.6667: R 1:1(0) win 0 (DF)
-Sample

1. **Source of Trace:**

   Traffic originated from an ISP in Korea.

2. **Detect was Generated by:**

   Detect was generated using TCPDump and Shadow.

3. **Probability Source Address was Spoofed:**

   Most likely the source address was spoofed. It is possible that our own address was spoofed, which would cause us to see the reflection.

4. **Description of Attack:**

   The incoming packets were SYN-ACK's coming from a single IP address. Our machines responded to this attack with a RESET to the incoming SYN-ACK. This kind of activity is not normal. There were no initial SYN's detected. The missing SYN's in question most likely are from another machine. If you look at the time stamps, they look to be slower than

a typical scan. Most likely other machines are being spoofed. Incoming SYN-ACK's are a good sign that a machine is being spoofed.

When conducting a search for port 6667 at: http://www.treachery.net/security_tools/ports/lookup.cgi, I found the following services that run on this particular port:

IRC, Kaitex Trojan, ScheduleAgent Trojan and Internet relay chat.

5. Attack Mechanism:

Most likely there is a Denial of Service attack against the remote host. This bad guy is not directly attacking our network.

6. Correlations:

Through the time that I have worked here, I have found this type of traffic within our logs only a few times. Most spoofs have different originating ports, whereas this spoof all came from port 6667.

7. Evidence of Active Targeting:

Because it was a spoof, there was no evidence of active targeting on our network. As stated above, most likely there was a DOS attack on the spoofed IP address.

8. Severity:

(Criticality + Lethality) - (System Countermeasures + Network Countermeasures) = Severity

Criticality:	3	My employer deems all machines critical machines.
Lethality:	1	This traffic is only response to the spoof.
Sys Counters:	5	There is nothing that can be defended against from a spoof.
Net Counters:	5	Firewall drops most of these packets.

Severity	-6	

9. Defensive Recommendations:

Traffic of this nature is not a problem, unless it becomes constant and causes slow downs on the network. Blocking connections to IRC or port 6667 should be done, but this is authorized activity. If it were to be blocked then a default deny policy on the firewall would mitigate this.

10. Multiple Choice Test Question:

What Trojans can be found on port 6667.

a) IRC
b) ICQ
c) ScheduleAgent
d) SubSeven

Answer: C

Detect #2:

Incoming Traffic:
06:00:22.983324 bad.guy.4269 > good.1.net.27374: S 150945583:150945583(0) win 16384 (DF) (ttl 114, id 9632)
06:00:22.983872 bad.guy.4270 > good.1.net.12345: S 150985516:150985516(0) win 16384 (DF) (ttl 114, id 9633)
06:00:22.984958 bad.guy.4271 > good.1.net.139: S 151049163:151049163(0) win 16384 (DF) (ttl 114, id 9634)
06:00:31.297420 bad.guy.4271 > good.1.net.139: S 151049163:151049163(0) win 16384 (DF) (ttl 114, id 11209)
06:00:40.727546 bad.guy.2177 > good.2.net.12345: S 241670575:241670575(0) win 16384 (DF) (ttl 114, id 14593)
06:00:40.729961 bad.guy.2176 > good.2.net.27374: S 241632144:241632144(0) win 16384 (DF) (ttl 114, id 14592)
06:00:40.731560 bad.guy.2178 > good.2.net.139: S 241707087:241707087(0) win 16384 (DF) (ttl 114, id 14594)
06:00:41.765678 bad.guy.4269 > good.1.net.27374: S 150945583:150945583(0) win 16384 (DF) (ttl 114, id 14907)
06:00:41.825911 bad.guy.4271 > good.1.net.139: S 151049163:151049163(0) win 16384 (DF) (ttl 114, id 14962)
06:00:44.539022 bad.guy.2522 > good.3.net.27374: S 258632843:258632843(0) win 16384 (DF) (ttl 114, id 15746)
06:00:44.539027 bad.guy.2523 > good.3.net.12345: S 258683969:258683969(0) win 16384 (DF) (ttl 114, id 15747)
06:00:44.546646 bad.guy.2524 > good.3.net.139: S 258747880:258747880(0) win 16384 (DF) (ttl 114, id 15748)
06:00:44.849729 bad.guy.2524 > good.3.net.139: . ack 1765907744 win 17520 (DF) (ttl 114, id 15970)
06:00:44.850398 bad.guy.2524 > good.3.net.139: F 0:0(0) ack 1 win 17520 (DF) (ttl 114, id 15971)
06:00:44.884418 bad.guy.39 > good.3.net: icmp: echo reply (DF) (ttl 114, id 15972)
06:00:45.075316 bad.guy.2524 > good.3.net.139: . ack 2 win 17520 (DF) (ttl 114, id 15992)
06:00:46.173782 bad.guy.2523 > good.3.net.12345: S 258683969:258683969(0) win 16384 (DF) (ttl 114, id 16065)
06:00:46.188360 bad.guy.2522 > good.3.net.27374: S 258632843:258632843(0) win 16384 (DF) (ttl 114, id 16072)
06:00:46.443164 bad.guy.2178 > good.2.net.139: S 241707087:241707087(0) win 16384 (DF) (ttl 114, id 16245)
06:00:46.449605 bad.guy.2176 > good.2.net.27374: S 241632144:241632144(0) win 16384 (DF) (ttl 114, id 16251)
06:00:46.518128 bad.guy.2177 > good.2.net.12345: S 241670575:241670575(0) win 16384 (DF) (ttl 114, id 16295)
06:00:46.955121 bad.guy.2523 > good.3.net.12345: S 258683969:258683969(0) win 16384 (DF) (ttl 114, id 16592)
06:00:47.861664 bad.guy.2522 > good.3.net.27374: S 258632843:258632843(0) win 16384 (DF) (ttl 114, id 16618)
06:00:51.587097 bad.guy.3004 > good.4.net.27374: S 295032343:295032343(0) win 16384 (DF) (ttl 114, id 18378)
06:00:51.587332 bad.guy.3005 > good.4.net.12345: S 295097126:295097126(0) win 16384 (DF) (ttl 114, id 18379)
06:00:51.588077 bad.guy.3006 > good.4.net.139: S 295159183:295159183(0) win 16384 (DF) (ttl 114, id 18380)
06:00:52.770569 bad.guy.3004 > good.4.net.27374: S 295032343:295032343(0) win 16384 (DF) (ttl 114, id 18816)
06:00:52.981678 bad.guy.3005 > good.4.net.12345: S 295097126:295097126(0) win 16384 (DF) (ttl 114, id 18871)
06:00:53.934247 bad.guy.3004 > good.4.net.27374: S 295032343:295032343(0) win 16384 (DF) (ttl 114, id 19290)
06:00:54.169858 bad.guy.3005 > good.4.net.12345: S 295097126:295097126(0) win 16384 (DF) (ttl 114, id 19391)
06:00:54.948553 bad.guy.2178 > good.2.net.139: S 241707087:241707087(0) win 16384 (DF) (ttl 114, id 19641)

1. **Source of Traffic**

   Traffic originated from an ISP in Seoul, Korea.

2. **Detect was generated by:**

   Using tcpdump log format (as used by Shadow IDS).

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

   There is a good chance that this address was not spoofed. Most likely it was reconnaissance so the scanner would get back information from machines scanned.

   But there could be a middleman involved here as well. There is a FIN-ACK that is incoming from the attacker, preceded by an ACK. Either it is spoofed, or a connection was made with one of the machines. If a connection was made with one of the machines, it is not evident from the traffic.

There is also an echo reply hidden within this scan. Not sure as to what its purpose is. An incoming echo reply would indicate a there was an echo request somewhere along the line. Since there was not echo request found within the traffic, it is possible that this traffic could have been spoofed.

4. Attack Description:

The tcpdump logs initially showed SYN packets originating from an IP address in Korea. All incoming packets originated from the same IP address. The IP scanned ports 139, 12345, and 27374. A further analysis on these ports:

Port #	Protocol	Keyword	Description
139	TCP or UDP	netbios-ssn	NETBIOS Session Service
12345	TCP	NetBus	[Trojan] NetBus backdoor Trojan
27374	TCP	SubSeven	[Trojan] SubSeven

5. Attack Mechanism:

It seems that the attack was generated by a scanning utility. It was scanning for open tcp ports (zombies/backdoors) for Netbios, SubSeven and NetBus, the later two both well-known Trojans.

Some of the abilities of SubSeven can be found on this website from Symantec: http://www.symantec.com/avcenter/venc/data/backdoor.subseven.22.a.html. It can search, retrieve, and send files. It can steal passwords, change resolution, etc. If this program is on your machine, good luck trying to get it off. It is a very powerful Trojan and is a security risk.

On NetBus: http://www.symantec.com/avcenter/venc/data/backdoor.netbus.444051.html

The attacker is most likely using a tool to scan for these three different ports on the same IP. Once the IP addresses were scanned for those three ports, the scan went to another subnet.

Even though these ports are common for the Trojans, in SubSeven, the port number can be configured to be anything the sender wants. The scan simply requires sending tcp SYN packet to the pre-configured port. If the attacker gets a SYN-ACK in response, then you are had. Most likely it was a script kiddie using a tool to scan for these ports.

6. Correlations:

These are the most popular Trojans out there today. It is always good to keep up the latest ports. A good search engine for ports is at http://www.ec11.dial.pipex.com/port-num.htm.

7. Evidence of Active Targeting:

Since these packets were to commonly used ports of SubSeven and NetBus, then it would mean that it was not active targeting. It is the work of script kiddies, especially since the subnets seem to randomize itself through various subnets. However, if the traffic were going to ports other than the normal ports, then it would be considered active targeting.

8. Severity:

(Criticality + Lethality) - (System Countermeasures + Network Countermeasures) = Severity

Criticality:	3	There are few boxes that are not considered critical.
Lethality:	1	The attack was not lethal.
Sys Counters:	4	Firewalls in place counteracted the attack.
Net Counters:	4	The response was only 'unreachable'. These ports were not open.

Severity	-4	

9. Defensive Recommendations:

The firewall needs to filter out packets destined for these ports mentioned above. Be sure to check all the machines within network on a regular basis for any vulnerability that they may have, especially if there is heavy scanning of this nature.

10. Multiple Choice Test Question:

What is a Trojan horse?

a) A program that neither replicates nor copies itself but does damage or compromises a computer.
b) A large wooden horse that hides soldiers inside. ☺
c) A necessary security program that needs to be run on every machine you own.
d) A virus that replicates on its own.

Answer: A

Detect #3

Incoming Traffic:
16:00:04.041838 bad.guy.44390 > good.guy1.111: R 1289085467:1289085467(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.041943 bad.guy.44392 > good.guy2.111: R 1289318608:1289318608(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.042466 bad.guy.44394 > good.guy3.111: R 1289459629:1289459629(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.043216 bad.guy.44396 > good.guy4.111: R 1289531254:1289531254(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.044040 bad.guy.44398 > good.guy5.111: R 1289567185:1289567185(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.044575 bad.guy.44391 > good.guy6.111: R 1289194260:1289194260(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.044579 bad.guy.44400 > good.guy7.111: R 1289779540:1289779540(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.044583 bad.guy.44393 > good.guy8.111: R 1289431466:1289431466(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.044587 bad.guy.44402 > good.guy9.111: R 1289896939:1289896939(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.045940 bad.guy.44395 > good.guy10.111: R 1289471633:1289471633(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.046002 bad.guy.44404 > good.guy11.111: R 1290021016:1290021016(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.046262 bad.guy.44406 > good.guy12.111: R 1290124342:1290124342(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.046304 bad.guy.44397 > good.guy13.111: R 1289556088:1289556088(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.046344 bad.guy.44399 > good.guy14.111: R 1289690962:1289690962(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.046350 bad.guy.44408 > good.guy15.111: R 1290145425:1290145425(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.047163 bad.guy.44401 > good.guy16.111: R 1289808676:1289808676(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.047446 bad.guy.44410 > good.guy17.111: R 1290247902:1290247902(0) win 8760 (DF) (ttl 239, id 33284)
16:00:04.047977 bad.guy.44412 > good.guy18.111: R 1290267355:1290267355(0) win 8760 (DF) (ttl 239, id 33294)

1. **Source of Trace:**

   This came from an Internet Service Provider in Beijing, China.

2. **Detect Generated by:**

   This detect was generated by Shadow and TCP Dump.

3. **Probability the Source Address was Spoofed:**

   This source was probably not spoofed; it is most likely reconnaissance so the scanner would come back to this computer after initial information received.

4. **Description of Attack:**

   The kind of attack presented here is a reset scan, can also be known as inverse scanning. The attacker sent a packet addressed to a machine located in our network that was protected by a firewall. The port that was targeted was port 111, also known as the SUN Remote Procedure Call (sunrpc). A favorite target of scanners is port 111, portmapper. By connecting to port 111, a user can find out what highport RPC services are running. There are numerous exploits for portmapper.

5. **Attack Mechanism:**

   Resets are normally seen when there are errors. Since much of this traffic is going to non-existing hosts, it is suspected that this attacker is conducting a reset scan.

   If the packet reached the target machine or was dropped by the firewall, then it is assumed that a machine may exist. But if an ICMP 'host unreachable' message is returned, it is then

assumed that a machine does not exist there, and the attacker can then concentrate on the machines that possibly exist. This attacker knows what does not exist, not the ones that do.

While he is conducting this scan, he is trying to avoid detection by the IDS, but this attacker can be caught. Taken directly from http://www.sans.org/infosecFAQ/audit/inverse_map.htm

> The problem for the attacker is that in order to gain information from the scan, he has to provide at least one genuine source address where he can study the returned packets. Many scanning tools, e.g. nmap[2], provide a way of sending decoy packets[5] from several forged source addresses to confuse the IDS systems. But always there is one genuine address among the rest, making the tracing of the attacker possible if not probable.

6. Correlations:

As seen above, Minna Kangasluoma has a very good over of inverse scanning that can be found at: http://www.sans.org/infosecFAQ/audit/inverse_map.htm

7. Active Targeting:

This scan is actively targeting us on port 111 using an inverse or reset scan. From this scan, the attacker was able to determine what hosts did not exist, not ones that did. This information could be used to determine what does exist, though.

8. Severity:

(Criticality + Lethality) - (System Countermeasures + Network Countermeasures) = Severity

Criticality:	3	Most machines at place of work are considered critical.
Lethality:	2	A scan like this can be lethal, considering what could be returned.
Sys. Counter:	4	It was blocked, but 'host unreachables' were sent out.
Net Counter:	4	'host unreachable' sent out. Port 111 was closed.

Severity	-3	

9. Defensive Recommendations:

One possible solution is to blocking reset scans at the firewall. A problem with this method is the possibility of sacrificing resources to connections, which the remote host is really trying to reset. Also, blocking all outgoing ICMP 'host unreachable' packets originating from the network. This would keep an attacker from having any knowledge of what is in the local network.

10. Multiple Choice:

What is one way of performing an inverse scan?

a) Send ICMP host unreachables

b) Attempt SYN scan
c) Sending RST packets
d) Going from last to first on the IP list.

Answer: c

Detect #4

Subject: NID PortScan Alert

Port Scan detected.
Origination MAC address: x:x:x:8d:f9:c4
Destination MAC address: x:x:x:12:57:dd
Origination IP address : bad.home.addr.64.63
Destination IP address : scanned.home.addr.153

TCPDump Data:
09:04:14.415418 bad.home.addr.64.63.20 > scanned.home.addr.153.1: S 1234:1234(0) win 4096
09:04:14.415542 bad.home.addr.64.63.20 > scanned.home.addr.153.1: S 1234:1234(0) win 4096
09:04:14.472771 bad.home.addr.64.63.20 > scanned.home.addr.153.7: S 1234:1234(0) win 4096
09:04:14.472897 bad.home.addr.64.63.20 > scanned.home.addr.153.7: S 1234:1234(0) win 4096
09:04:14.535312 bad.home.addr.64.63.20 > scanned.home.addr.153.9: S 1234:1234(0) win 4096
09:04:14.535488 bad.home.addr.64.63.20 > scanned.home.addr.153.9: S 1234:1234(0) win 4096
09:04:14.603330 bad.home.addr.64.63.20 > scanned.home.addr.153.11: S 1234:1234(0) win 4096
09:04:14.603488 bad.home.addr.64.63.20 > scanned.home.addr.153.11: S 1234:1234(0) win 4096
09:04:14.663929 bad.home.addr.64.63.20 > scanned.home.addr.153.13: S 1234:1234(0) win 4096
09:04:14.664117 bad.home.addr.64.63.20 > scanned.home.addr.153.13: S 1234:1234(0) win 4096
09:04:14.726225 bad.home.addr.64.63.20 > scanned.home.addr.153.15: S 1234:1234(0) win 4096
09:04:14.726386 bad.home.addr.64.63.20 > scanned.home.addr.153.15: S 1234:1234(0) win 4096
09:04:14.800974 bad.home.addr.64.63.20 > scanned.home.addr.153.19: S 1234:1234(0) win 4096
09:04:14.801117 bad.home.addr.64.63.20 > scanned.home.addr.153.19: S 1234:1234(0) win 4096
09:04:14.864544 bad.home.addr.64.63.20 > scanned.home.addr.153.20: S 1234:1234(0) win 4096
09:04:14.864687 bad.home.addr.64.63.20 > scanned.home.addr.153.20: S 1234:1234(0) win 4096
09:04:14.924490 bad.home.addr.64.63.20 > scanned.home.addr.153.21: S 1234:1234(0) win 4096
09:04:14.924662 bad.home.addr.64.63.20 > scanned.home.addr.153.21: S 1234:1234(0) win 4096
09:04:14.987991 bad.home.addr.64.63.20 > scanned.home.addr.153.22: S 1234:1234(0) win 4096
09:04:14.988100 bad.home.addr.64.63.20 > scanned.home.addr.153.22: S 1234:1234(0) win 4096
09:04:15.054482 bad.home.addr.64.63.20 > scanned.home.addr.153.23: S 1234:1234(0) win 4096
09:04:15.054600 bad.home.addr.64.63.20 > scanned.home.addr.153.23: S 1234:1234(0) win 4096
09:04:15.113005 bad.home.addr.64.63.20 > scanned.home.addr.153.25: S 1234:1234(0) win 4096
09:04:15.113103 bad.home.addr.64.63.20 > scanned.home.addr.153.25: S 1234:1234(0) win 4096
09:04:15.198699 bad.home.addr.64.63.20 > scanned.home.addr.153.37: S 1234:1234(0) win 4096
09:04:15.198801 bad.home.addr.64.63.20 > scanned.home.addr.153.37: S 1234:1234(0) win 4096
09:04:15.254265 bad.home.addr.64.63.20 > scanned.home.addr.153.43: S 1234:1234(0) win 4096
09:04:15.254353 bad.home.addr.64.63.20 > scanned.home.addr.153.43: S 1234:1234(0) win 4096
09:04:15.319680 bad.home.addr.64.63.20 > scanned.home.addr.153.53: S 1234:1234(0) win 4096
09:04:15.319775 bad.home.addr.64.63.20 > scanned.home.addr.153.53: S 1234:1234(0) win 4096
09:04:15.394302 bad.home.addr.64.63.20 > scanned.home.addr.153.57: S 1234:1234(0) win 4096
09:04:15.394454 bad.home.addr.64.63.20 > scanned.home.addr.153.57: S 1234:1234(0) win 4096
09:04:15.459143 bad.home.addr.64.63.20 > scanned.home.addr.153.70: S 1234:1234(0) win 4096
09:04:15.459343 bad.home.addr.64.63.20 > scanned.home.addr.153.70: S 1234:1234(0) win 4096
09:04:15.529355 bad.home.addr.64.63.20 > scanned.home.addr.153.77: S 1234:1234(0) win 4096
09:04:15.529476 bad.home.addr.64.63.20 > scanned.home.addr.153.77: S 1234:1234(0) win 4096
09:04:15.644876 bad.home.addr.64.63.20 > scanned.home.addr.153.80: S 1234:1234(0) win 4096
09:04:15.644990 bad.home.addr.64.63.20 > scanned.home.addr.153.80: S 1234:1234(0) win 4096
09:04:15.716672 bad.home.addr.64.63.20 > scanned.home.addr.153.87: S 1234:1234(0) win 4096
09:04:15.716770 bad.home.addr.64.63.20 > scanned.home.addr.153.87: S 1234:1234(0) win 4096
09:04:15.769629 bad.home.addr.64.63.20 > scanned.home.addr.153.88: S 1234:1234(0) win 4096
09:04:15.769735 bad.home.addr.64.63.20 > scanned.home.addr.153.88: S 1234:1234(0) win 4096
09:04:15.863185 bad.home.addr.64.63.20 > scanned.home.addr.153.95: S 1234:1234(0) win 4096
09:04:15.863302 bad.home.addr.64.63.20 > scanned.home.addr.153.95: S 1234:1234(0) win 4096

09:04:15.945075 bad.home.addr.64.63.20 > scanned.home.addr.153.101: S 1234:1234(0) win 4096
09:04:15.945178 bad.home.addr.64.63.20 > scanned.home.addr.153.101: S 1234:1234(0) win 4096
09:04:16.008283 bad.home.addr.64.63.20 > scanned.home.addr.153.102: S 1234:1234(0) win 4096
09:04:16.008444 bad.home.addr.64.63.20 > scanned.home.addr.153.102: S 1234:1234(0) win 4096
09:04:16.069456 bad.home.addr.64.63.20 > scanned.home.addr.153.103: S 1234:1234(0) win 4096

1. **Source of Trace**

   Trace came within our own network.

2. **Detect Generated by:**

   Data was found using NID and TCPDump.

3. **Was Source Spoofed:**

   The source was not spoofed (explanation below).

4. **Description of Attack:**

   A machine inside our network was seen port scanning other machines within the network, all within the firewall. The initial machine was scanning one IP address for it's entire range of open ports. The scan was coming from port 20. All sequence numbers were identical (1234). When the scan reached port 65535 it stopped with its scan of that IP and moved on to another IP, and proceeded to port scan a total of five IP addresses in all.

5. **Attack Mechanism:**

   The scanner was scanning IP's for open ports.

6. **Correlations:**

   A call was made to the Incident Response team to tell them of an internal scan that was not scheduled or a possible compromised machine. What was determined was the scanner was our own penetration team performing a routine scan. These scans are performed at a regular basis and sufficient warnings are sent out ahead of time announcing these scans. In this case, none were made. Hence my initial reaction when I saw an internal machine scanning another internal machine.

7. **Active Targeting:**

   The scanning machine was actively targeting the source, but it was not intended to be malicious activity. The source was trusted.

8. **Severity:**

   (Criticality + Lethality) - (System Countermeasures + Network Countermeasures) = Severity

Criticality:	2	Unix and Windows machines were involved.
Lethality:	1	A vulnerability scan by internal machines. A trusted source.
Sys Counters:	4	There were no open machines detected.
Net Counters:	1	Attack originated within the firewall.

Severity:	-2	

9. Defensive Recommendations:

What needed to be done was an email alert stating when the scan was taking place and which hosts where being scanned. The person conducting the scan was a trusted associate of mine, but had forgot to send out a notice. They are allowed to run internal scans on the network.

It would have been a concern if someone inside had downloaded a scanning tool and started scanning internal machines without proper authorization. This type of action goes against my company's acceptable use policy.

Internal scans need to be made known when they are conducted. If internal scans are conducted without prior knowledge and unauthorized people are doing them, then the company's acceptable use policy should be adhered to.

If it is found that the scanning machine was a compromise, then it should be taken off the network immediately and forensics should be performed.

10. Multiple Choice:

What are the benefits of scanning internal machines?

a) Determine what ports are open
b) Find security holes and vulnerabilities
c) Determine if hosts are configured properly
d) All the above

The correct answer is D.

Detect #5

Incoming Traffic:
11:33:44.163293 bad.guy.net.111 > good.guy.net.111: SF 1163897957:1163897957(0) win 1028

1. **Source of Trace:**

   The trace originated from an ISP in India.

2. **Detect Generated By:**

   Data was found using TCPDUMP from my employer's network.

3. **Probability the Source Address was Spoofed:**

   It is possible that the source address was spoofed (explanation below).

4. **Description of Attack:**

   The detected traffic above was a single packet that came in from an ISP in India. Since both the SYN and FIN flags were set, the packet was most likely crafted. There were no other packets that came in with it. Most likely it is not spoofed because of the single crafted packet.

   The attacker, by sending a single packet to port 111 (portmapper), could be sweeping entire subnets looking for UNIX boxes with portmapper open. The attacker was probably hoping to avoid detection by using a single packet with the SYN-FIN flags set, but he did not thanks to the trusty tcpdump.

   Since this was a reconnaissance mission, the attacker received an 'ICMP host unreachable' message from the firewall. If the response had been different, the attacker would have a few options based on portmapping vulnerabilities. The next step the attacker should have taken would be to attempt an OS fingerprint in an effort to find out which version of operating system was running and what version of portmapper. Once this is found, the attacker has all sorts of options at his fingertips.

5. **Attack Mechanism:**

   The attack was used to collect information that could possibly be used at a later time. If the packet had gotten through the firewall, the Linux box would have responded differently if something was running on that port or not. A SYN-FIN to an open port on a Linux machine will return a SYN-ACK. A closed port will return a RESET-ACK. Thankfully the packet did not get through the firewall. If it would have, the host would have given out a lot of information about the box just from a single packet.

6. **Correlations:**

An example of this type of scan is posted by Terry Bidwell on the SANs website at http://www.sans.org/y2k/111600.htm**.** In his article, Terry suggests that this scan could be a complicated 3-way spoofed source bounced network scan. He believes it to be generated by a tool called Idlescan.

> There have been several postings to security mailing lists asking about IDS network detects showing SYN-FIN port cans where the source port == the destination port, the IP ID is always 39426, and the window size is always 1028 (0x404). The general question seems to be "what tool is generating these scans?" A general example of one of the IDS detects is below:
>
> [**] SCAN-SYN FIN [**]
> 10/26-18:14:45.311283 209.1.2.102:21 -> my.host1.com:21 TCP TTL:30
> TOS:0x0 ID:39426 **SF**** Seq: 0x117EB2A6 Ack: 0x2CB3E404 Win: 0x404

I am not suggesting that this scan is what Terry mentions in this article. It just looks very similar to the type of scan he is describing here and it could be the same.

7. Active Targeting:

Even though there was just one packet sent, it was sent to port 111, which by itself is active targeting. The attacker possibly had a script of IP addresses as target hosts, or the attacker had more than one compromised box to attack with

8. Severity:

(Criticality + Lethality) - (System Countermeasures + Network Countermeasures) = Severity

Criticality:	4	Because of the possibilities of vulnerabilities with portmapper
Lethality:	2	The attack failed, but he did get back a response
Sys Counters:	4	Because the firewall sent out 'host unreachable'
Net Counters:	5	Port 111 was not open and tcpdump caught it.

Severity	-3	

9. Defensive Recommendations:

A stateful inspection of the firewall would be in order. Find out if we want 'ICMP host unreachable' to be sent out. This targeted machine should be checked for vulnerabilities and it should have all the updated patches.

10. Multiple Choice:

What type of response is expected when a SYN-FIN packet is sent, from an open port and a closed port on a Linux box?

a) SYN from an open port and RESET from a closed port.

b) FIN from open port and RESET from closed port.
c) SYN-ACK from open port and RESET-ACK from closed port.
d) RESET from open port and RESET from closed port.

Answer: C

Section 3

Analyze This

In this analysis of provided snort logs, the analyst will attempt to identify areas for improvement in both the client's network and sensor configuration. The analyst will attempt to find signs of compromised machines and provide trends for the most frequent attacks while trying to lead the client through the analysis process so they can understand the intrusion analysis that has taken place. Throughout this document defensive recommendations will be provided where possible. To begin, a summary of alerts are provided for initial analysis.

Summary of Alerts

Snort data provided by http://www.research.umbc.edu/~andy/. Datasets that were used:

Dataset	Date
Alert files	10/15/2001 – 10/19/2001
Scan files	10/15/2001 – 10/19/2001
OOS files	10/15/2001 – 10/19/2001

Event Tally
There were 477103 scans and 207715 alert events over the given period of time.
Listed below are the top signatures from the provided datasets.

Alert Signatures	# of Alerts
MISC Large UDP Packet	38906
ICMP Echo Request speedera	27964
WEB-MISC Attempt to execute cmd	22346
spp_http_decode: IIS Unicode attack detected	12188
INFO MSN IM Chat data	8867
WEB-MISC prefix-get //	8220
ICMP Echo Request Nmap or HPING2	7586
Port 55850 tcp - Possible myserver activity - ref. 010313-1	7132
Watchlist 000220 IL-ISDNNET-990517	5159
MISC source port 53 to <1024	4724
ICMP Destination Unreachable (Communication Administratively Prohibited)	4328
CS WEBSERVER - external web traffic	4321
Incomplete Packet Fragments Discarded	4309
SMB Name Wildcard	3957
WEB-MISC 403 Forbidden	3574
MISC traceroute	3329
Possible trojan server activity	2996

Since such a large amount of data is collected in typical IDS deployments, there are some fundamental decisions that must be made when analyzing the datasets:

1. What are the events of interest that will stimulate a response?
2. How do we respond to these events of interest?
3. Are there other secondary events that we would be interested but are not able to respond due to lack of manpower and time?

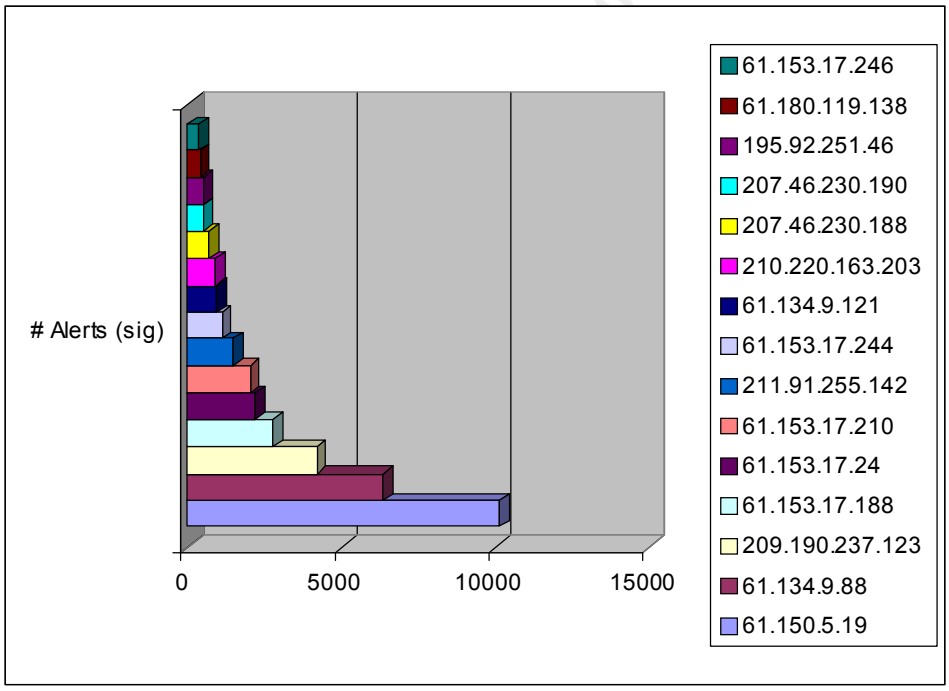
When approaching the given datasets, the same questions were faced above. The difference between this and an operational analysis is the difference in time. This is most definitely a historical dataset and is not constrained by time. If this were in a real-time operational environment, the analyst would be able to track potentially hostile probes that would be seen coming through the network.

Most of the alerts will fall in one of three categories: Hostile probes, Misuse and Compromise. These types of alerts will mainly be the ones analyzed in this section. Bear in mind, however, that not all the alerts will fall in these categories. It is these types of alerts that will fall in the category of false positives.

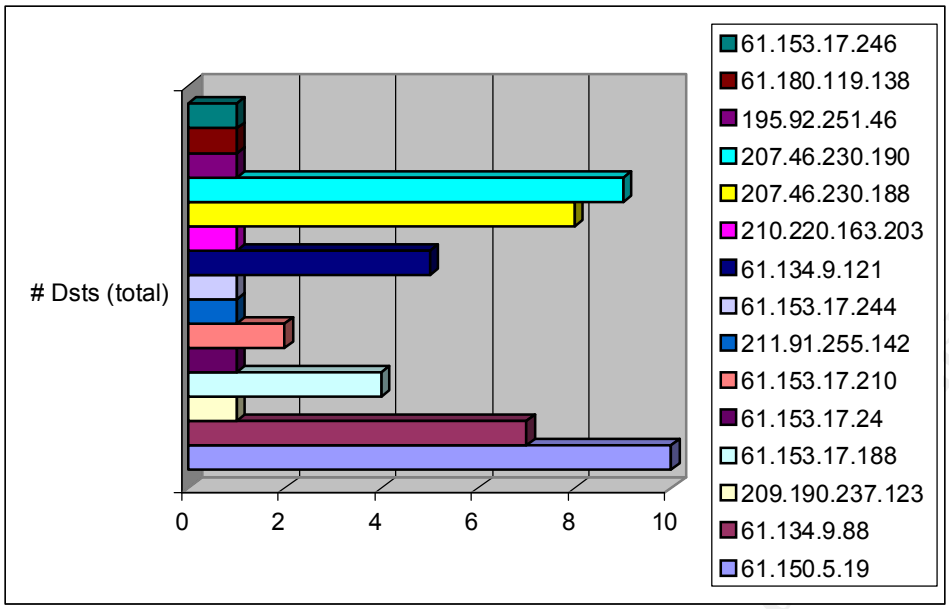
MISC Large UDP Packet

Description:

This alert was set off from the large number of MISC UDP Packets. The number of alerts seen:



Below is the number of destinations:



Below is some sample traffic:

10/16-18:34:51.050558 [**] MISC Large UDP Packet [**] 61.134.9.88:0 -> 10.10.111.142:0
10/16-18:34:53.925866 [**] MISC Large UDP Packet [**] 61.134.9.88:0 -> 10.10.111.142:0
10/16-18:34:54.229426 [**] MISC Large UDP Packet [**] 61.134.9.88:0 -> 10.10.111.142:0
10/16-18:34:54.832733 [**] MISC Large UDP Packet [**] 61.134.9.88:0 -> 10.10.111.142:0
10/16-18:34:55.907134 [**] MISC Large UDP Packet [**] 61.134.9.88:0 -> 10.10.111.142:0
10/16-18:34:56.522178 [**] MISC Large UDP Packet [**] 61.134.9.88:0 -> 10.10.111.142:0
10/16-18:34:59.267220 [**] MISC Large UDP Packet [**] 61.134.9.88:2035 -> 10.10.111.142:2085
10/16-18:35:05.968225 [**] MISC Large UDP Packet [**] 61.134.9.88:0 -> 10.10.111.142:0
10/16-18:35:09.197303 [**] MISC Large UDP Packet [**] 61.134.9.88:0 -> 10.10.111.142:0
10/16-18:35:15.448456 [**] MISC Large UDP Packet [**] 61.134.9.88:0 -> 10.10.111.142:0
10/16-18:35:15.553005 [**] MISC Large UDP Packet [**] 61.134.9.88:0 -> 10.10.111.142:0
10/16-18:35:16.460606 [**] MISC Large UDP Packet [**] 61.134.9.88:0 -> 10.10.111.142:0

Analysis:

The reason only this traffic was shown, is because all the traffic was generally the same. Why so much traffic seen only going to a few destinations compared to the number of alerts generated? Most likely what is being seen here is multiple AIX boxes, a Unix server made by IBM. By default, Path MTU Discovery is enabled. It sends an interface-MTU sized ICMP Echo request with the Don't Fragment bit set, and sees if any machine complains that fragging is needed. (http://cert.uni-stuttgart.de/archive/incidents/2001/07/msg00269.html)

Version 4.3.3 of AIX comes default with PMTU Discovery. To disable this on the AIX box, use this command:

no -o tcp_pmtu_discover=0

That is, if you have an AIX box. From the look of where the traffic is coming from, then those incoming are possibly the AIX boxes.

Security Recommendation:

I would not worry too much about this traffic. If it starts to be a hindrance to the network then I would recommend blocking the offending IP addresses if possible. What could also be done is get in touch with the owner's of those IP addresses and kindly ask them to disable PMTU Discovery.

ICMP Echo Request speedera:

Description and Analysis:

Source	Alerts	Destinations
10.10.205.130	27964	2

A sample of traffic seen:
10/18-03:25:25.326577 [**] ICMP Echo Request speedera [**] 10.10.205.130 -> 216.102.120.11
10/18-03:25:25.431055 [**] ICMP Echo Request speedera [**] 10.10.205.130 -> 216.102.120.11
10/18-03:25:25.431521 [**] ICMP Echo Request speedera [**] 10.10.205.130 -> 216.102.120.11
10/18-03:25:25.521669 [**] ICMP Echo Request speedera [**] 10.10.205.130 -> 216.102.120.11
10/18-03:25:25.821523 [**] ICMP Echo Request speedera [**] 10.10.205.130 -> 216.102.120.11
10/18-03:25:26.026691 [**] ICMP Echo Request speedera [**] 10.10.205.130 -> 216.102.120.11
10/18-03:25:26.275933 [**] ICMP Echo Request speedera [**] 10.10.205.130 -> 216.102.120.11

The source of this traffic is coming from an internal machine that at first glance makes one think that this comes from Speedera.net's "Global Traffic Management" system. Most likely this is a misconfigured box that is sending out these 'echo requests'. The reason being the source of the pings are internal rather than external. An explanation of what Speedera.net does taken from http://www.whitehats.com/IDS/IDS152:

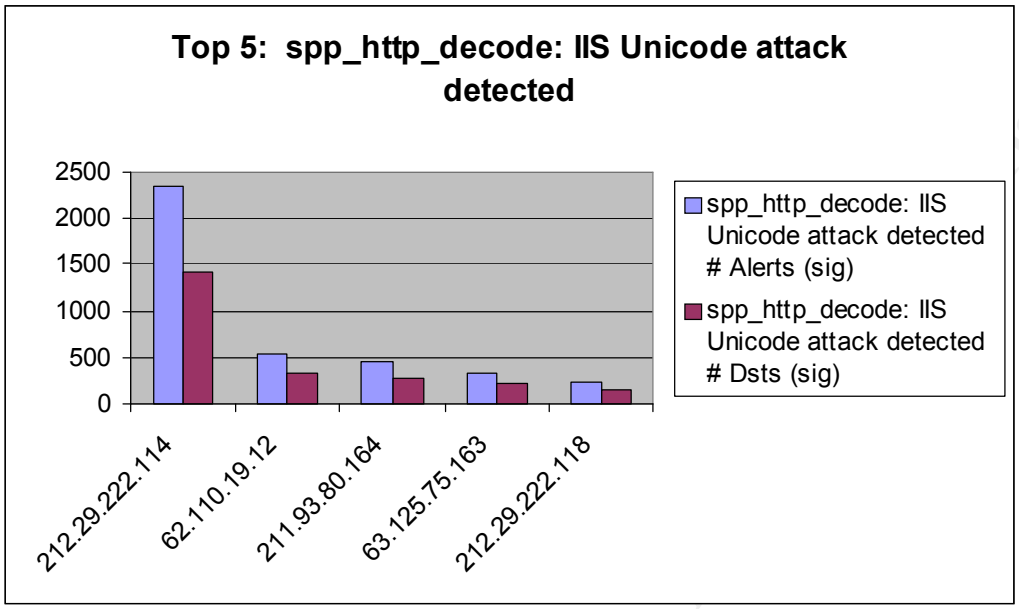
> A company named Speedera has a new technology that uses roughly 90 machines distributed around the world to detect the closest web server to you for large corporate sites. They seem to test internet latency using BSD type pings. Each time someone connects to a Speedera hosted site, you will see roughly 90 hosts ping you with a BSD type payload.

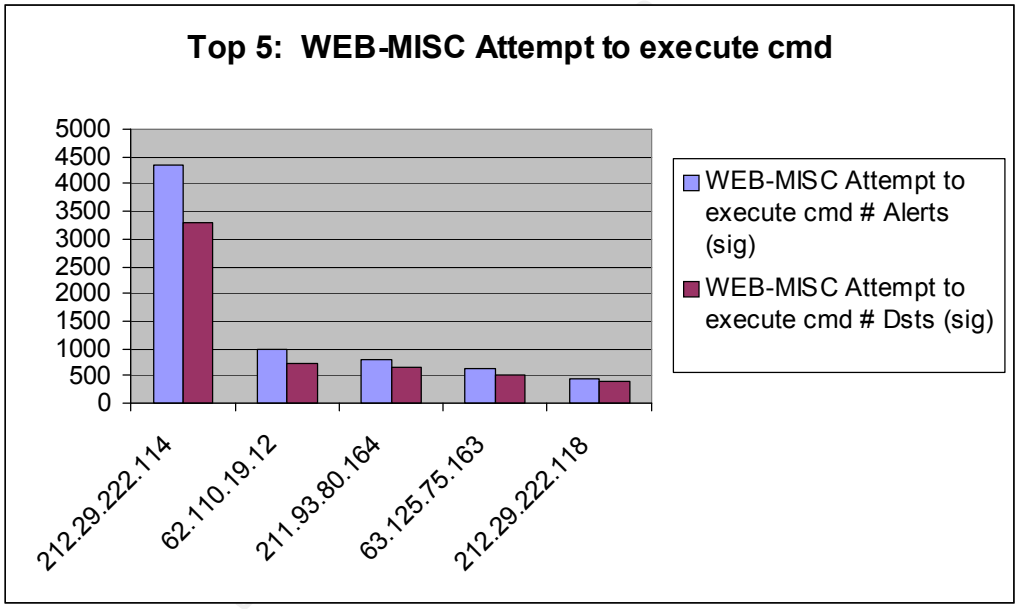
Most likely this is not the case since the IP addresses that were hit were not any of Speedera's machines. I would check this box to be sure that it is configured correctly, since it is pinging the same IP over and over.

WEB-MISC Attempt to execute cmd:
spp_http_decode: IIS Unicode attack detected:

Description and Analysis:
These two alert signatures have been put together since both refer to the Code Red worm.

WEB-MISC Attempt to execute cmd	7988 sources	7235 destinations
spp_http_decode: IIS Unicode attack detected	3779 sources	3547 destinations





As you can see, the top 5 source IP's for both alert signatures are the same. Whereas the destination IP's are all random, that is there is not a pattern of destination IP's that are being targeted.

Below is a traffic sample that represents both alerts:

10/15-00:25:33.114439 [**] spp_http_decode: IIS Unicode attack detected [**] 212.29.222.114:1119 -> 10.10.18.93:80
10/15-00:25:33.114439 [**] WEB-MISC Attempt to execute cmd [**] 212.29.222.114:1119 -> 10.10.18.93:80
10/15-00:25:40.226665 [**] spp_http_decode: IIS Unicode attack detected [**] 212.29.222.114:46926 -> 10.10.80.118:80
10/15-00:25:40.226665 [**] WEB-MISC Attempt to execute cmd [**] 212.29.222.114:46926 -> 10.10.80.118:80

10/15-00:25:41.205737 [**] WEB-MISC Attempt to execute cmd [**] 212.29.222.114:8701 -> 10.10.75.142:80
10/15-00:25:56.536688 [**] WEB-MISC Attempt to execute cmd [**] 212.29.222.114:47101 -> 10.10.120.33:80
10/15-00:26:00.394419 [**] spp_http_decode: IIS Unicode attack detected [**] 212.29.222.114:45657 -> 10.10.80.118:80
10/15-00:26:00.394419 [**] WEB-MISC Attempt to execute cmd [**] 212.29.222.114:45657 -> 10.10.80.118:80
10/15-00:26:00.619462 [**] spp_http_decode: IIS Unicode attack detected [**] 212.29.222.114:33385 -> 10.10.229.179:80
10/15-00:26:00.619462 [**] spp_http_decode: IIS Unicode attack detected [**] 212.29.222.114:33385 -> 10.10.229.179:80
10/15-00:26:00.619462 [**] spp_http_decode: IIS Unicode attack detected [**] 212.29.222.114:33385 -> 10.10.229.179:80
10/15-00:26:00.619462 [**] WEB-MISC Attempt to execute cmd [**] 212.29.222.114:33385 -> 10.10.229.179:80

Most likely what is seen here is a Code Red 2 attack or some variant. It seems that our machines have been properly patched with the latest updates. None of the machines responded to the attacks. All source IP's did not include or own.

IP address 212.29.222.114 was the most talkative of all the incoming IP's, with over 6700 instances of the two signatures.

The Code Red worm looks for systems running IIS that have not patched the unchecked buffer vulnerability in idq.dll or removed the ISAPI script mappings. The worm exploits the vulnerability to inject itself into a system. If any system is running Windows 2000 it may have a vulnerable IIS server installed. (http://www.incidents.org/react/code_redII.php)

INFO MSN IM Chat data

Description and Analysis:

INFO MSN IM Chat data	779 sources	666 destinations

What is most likely being seen here is the Microsoft Instant Messenger being run on various machines throughout the network. The traffic seems to be normal concerning this. What should be done is review the company's acceptable use policy on whether or not instant messaging is acceptable. If it is acceptable use, then filtering out this traffic on Snort would be advised.

WEB-MISC prefix-get //

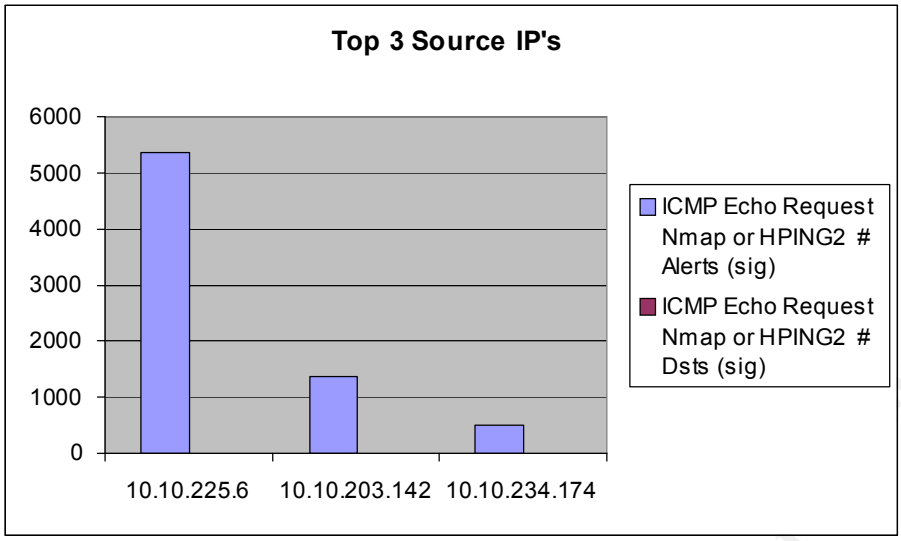
WEB-MISC prefix-get //	1811 sources	3 destinations

What is seen in this alert signature is probably someone typing in two slashes before the .com and before the rest of the URL. This is not seen as a problem.

ICMP Echo Request Nmap or HPING2

Description and Analysis:

ICMP Echo Request Nmap or HPING2	94 sources	98 destinations



The top three source IP addresses are shown above. Sample traffic is seen below:
10/15-02:33:01.878833 [**] ICMP Echo Request Nmap or HPING2 [**] 10.10.225.6 -> 204.71.200.75
10/15-02:34:52.884968 [**] ICMP Echo Request Nmap or HPING2 [**] 10.10.225.6 -> 206.79.171.51
10/15-02:35:19.390774 [**] ICMP Echo Request Nmap or HPING2 [**] 10.10.225.6 -> 206.79.171.51
10/15-02:36:48.390657 [**] ICMP Echo Request Nmap or HPING2 [**] 10.10.225.6 -> 204.152.190.70
10/15-02:40:51.912310 [**] ICMP Echo Request Nmap or HPING2 [**] 10.10.225.6 -> 206.79.171.51
10/15-02:44:33.923522 [**] ICMP Echo Request Nmap or HPING2 [**] 10.10.225.6 -> 204.152.190.70

Taken from http://www.hping.org/manpage.html, Hping2 is a network tool able to send custom TCP/IP packets and to display target replies like ping program does with ICMP replies. Using hping2 you can perform the following: Test firewall rules, advanced port scanning, test net performance using different protocols, packet size, type of service and fragmentation, Path MTU Discovery, and many others.

IP 10.10.225.6 is seen going to only 3 destinations, but triggering 5353 alerts. IP 10.10.203.142 was seen going to 7 destinations and 10.10.234.174 went to 6 destinations.

Traffic of this nature is not normal unless the box has been set up that way. I would recommend checking these three machines first and be sure that all patches have been installed.

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

Description and Analysis

Source	# Alerts (sig)	# Dsts (sig)
10.10.226.58	4538	1
194.213.87.193	2488	1

The two most talkative IP's are the two listed above, going back and forth to each other. The source port for IP 10.10.226.58 is 6346 and the destination is 55850. Port 6346 is the source port for Gnutella, a peer-to-peer file-sharing tool. Most likely what is being seen is an internal machine running Gnutella. I would recommend checking the acceptable use policy to see if running peer-to-peer programs is acceptable.

Watchlist 000220 IL-ISDNNET-990517

Description and Analysis:

This alert focuses on a block of IP addresses in the 212.179.x.x subnet. These alerts might be a security issue with the ISP. The first thing that was checked was whom this block of IP addresses belongs to. A search for the whois:

inetnum: 212.179.0.0 - 212.179.1.255
netname: AREL-NET
descr: arel-net
country: IL
admin-c: TP1233-RIPE
tech-c: TP1233-RIPE
status: ASSIGNED PA
notify: hostmaster@isdn.net.il
mnt-by: RIPE-NCC-NONE-MNT
changed: hostmaster@isdn.net.il 19990624
source: RIPE

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

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

Source	# Alerts (sig)	Target IP address	Ports Targeted
212.179.68.67	2436	10.10.x.x.	80
212.179.48.2	1070	10.10.237.26	4808

212.179.85.174	541	10.10.x.x.	80
212.179.88.222	185	10.10.202.142	1214
212.179.87.156	113	10.10.225.66	41110
212.179.27.6	77	10.10.227.58	1214
212.179.44.100	62	10.10.97.254	1214
212.179.127.53	58	10.10.237.122	1214
212.179.29.218	52	10.10.202.190	1962
212.179.86.37	28	10.10.237.224	1214

The only ports not known that are listed above 4808 and 4110. Port 80 is of course http. Port 1214 is KaZaa, a peer-to-peer file-sharing program like Gnutella. Port 1962 is BIAP-MP.

The two offending IP's that were targeting port 80 were scanning the subnet looking for port 80. Our traffic shows that none of our machines sent packets back to the offending IP addresses. This is always a good thing. It is best that a machine avoid returning traffic. Sometimes this is not always possible, but is desired.

For traffic concerning port 1214(KaZaa) it is best to observe the acceptable use policy regarding file-sharing over the network.

Traffic directed at ports 4808 and 4110 should be monitored closely. It is possible that there is a new exploit within these two ports. A scan of these ports should probably be conducted within the network.

MISC source port 53 to <1024

Description and Analysis:

Source	# Alerts (sig)	# Dsts (sig)
134.93.19.12	530	1
192.115.189.100	118	1

A sample of the traffic seen:

```
10/19-18:20:22.384753 [**] MISC source port 53 to <1024 [**] 134.93.19.12:53 -> 10.10.130.122:53
10/19-18:46:23.755895 [**] MISC source port 53 to <1024 [**] 134.93.19.12:53 -> 10.10.130.122:53
10/19-19:00:24.413216 [**] MISC source port 53 to <1024 [**] 134.93.19.12:53 -> 10.10.130.122:53
10/19-19:22:25.624211 [**] MISC source port 53 to <1024 [**] 134.93.19.12:53 -> 10.10.130.122:53
10/19-19:22:25.628369 [**] MISC source port 53 to <1024 [**] 134.93.19.12:53 -> 10.10.130.122:53
10/19-19:24:25.739331 [**] MISC source port 53 to <1024 [**] 134.93.19.12:53 -> 10.10.130.122:53
```

We are seeing traffic from a source port of 53 (DNS) to a destination port of 53. While traffic of this sort is atypical, it is not unheard of. The machine seems to be set up to receive DNS responses on port 53. This is not a major problem. A bit of tweaking of the Snort rule would

help. Setting up a rule that limits the responses accepted to be sourced from the DNS servers alone. This way you can still catch people trying to get to your LAN via port 53.

ICMP Destination Unreachable (Communication Administratively Prohibited)

Description and Analysis:

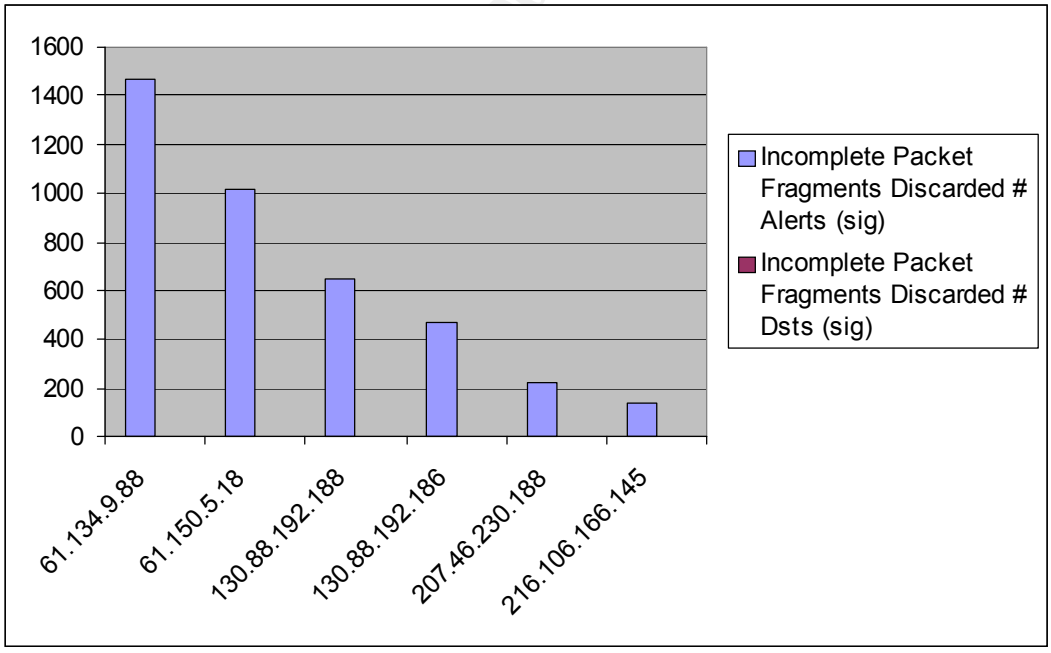
This alert is generated by a router if it cannot forward a datagram due to administrative filtering. This code is deemed optional and routers may be configured to not send it. (http://www.camtp.uni-mb.si/books/Internet-Book/ICMP_DestUnreach.html)

Source	# Alerts (sig)	# Dsts (sig)
10.10.14.1	3578	243

If the router has the option enabled to not allow these messages to be generated no ICMP error message is sent in response to a packet that is dropped because its forwarding is administratively prohibited. (http://sunsite.dk/RFC/rfc/rfc1812.html)

Incomplete Packet Fragments Discarded

Description and Analysis:



Seen above in the chart are the top talkers of this alert. From the traffic generated, there does not seem to be any malicious activity being done. Rather this message was given from the

defragmentation preprocessor when packets bigger than 8k are more than half empty when the last fragment is received and discarded.

This can be caused by:
-Transmission errors
-Broken stacks
-And fragmentation errors

Most likely the packet may be corrupted 'NFS' data and the preprocessor has failed to rebuild fragmented packet.

SMB Name Wildcard

Description and Analysis:

Microsoft Windows use port 137 (netbios) to resolve hostnames when given an IP address. SMB Name Wildcard alerts are usually normal traffic seen coming from Windows machines. Unless an extreme amount of traffic from port 137 is seen, treat this traffic as acceptable. If a machine exhibits an extreme amount of traffic, it is possible that that machine has been compromised or some other problem.
There was not a large amount of SMB Name Wildcard traffic seen in the alerts.

An open-eye should be kept on unusually large amounts of traffic on port 137.

WEB-MISC 403 Forbidden

Description and Analysis:

This alert revolves around receiving an error when trying to connect to a host that is forbidden to get to on the web. Either the IP address is not a server or the IP trying to gain access does not have the correct permissions to access the page.

There is just one internal IP that might garner some further investigation:

Source	# Alerts (sig)	# Dsts (sig)
10.10.100.165	2997	144

It is possible that this IP was trying to reach the destination 216.239.46.x and unable to do so because that subnet did not have a web server or is being denied access. If this traffic continues you might want to check into the IP it is trying to gain access to or see if this particular machine is set up properly.

MISC traceroute

Description and Analysis:

Taken from http://www.whitehats.com/IDS/3+MISC+traceroute&hl=en&start=1, this event indicates that there was a traceroute from an external source, probably from a Unix machine. This technique is implemented by TracerX to discover the route that packets take to reach your machines. Although a TCP packet caused this event, the packet is not thought to be a part of an existing TCP session. Therefore the source IP address could be easily forged. Since the intruders desire is to get response to their packets, it may be likely that the source IP address is not spoofed.

There is one internal IP that might need to be investigated further:

Destinations	# Alerts (sig)	# Srcs (sig)
10.10.140.9	3289	116

Of the 141 sources of this alert signature, 116 came to this particular IP. There was a lot of external IP's trying to gather information about this IP. This box should be check for any vulnerabilities and have all the latest patches installed.

Possible trojan server activity

Description and Analysis:

This alert is concentrated on port 27374, otherwise known as the source port for the SubSeven Trojan. SubSeven took over where NetBus left off. NetBus was the first ever point-and-click program that allowed hackers to abuse systems that were infected. SubSeven just took this idea further and give the intruders more control (http://www.hackfix.org/subseven/about.shtml). SubSeven will run as a server on both Microsoft Windows and Macintosh machines. The server module is usually wrapped in an executable file. The executable will run as expected, but hidden within is the SubSeven Trojan; while unbeknownst to the user will install itself onto the machine. Remote users run SubSeven by remote users using a GUI that allows them complete access to the compromised machine. Some of the features that make SubSeven the most popular Trojan on the Internet are: File Control – move, copy, rename, delete, and run executables. They can monitor everything that is seen on the screen including but not limited to: moving windows, moving the mouse and resizing windows, and complete network control. Another amazing feature is that it lets the remote host know when the infected computer is connected to the Internet by IRC, ICQ or email.

One internal IP of particular interest is 10.10.97.134. It was seen scanning various subnets for SubSeven. A sample of the traffic below:

10/18-22:39:22.954349 [**] Possible trojan server activity [**] 10.10.97.134:1735 -> 213.73.109.95:27374
10/18-22:39:35.754521 [**] Possible trojan server activity [**] 10.10.97.134:1811 -> 213.73.109.166:27374

10/18-22:39:36.139163 [**] Possible trojan server activity [**] 10.10.97.134:1828 -> 213.73.109.183:27374
10/18-22:39:38.734353 [**] Possible trojan server activity [**] 10.10.97.134:1811 -> 213.73.109.166:27374
10/18-22:39:38.751944 [**] Possible trojan server activity [**] 10.10.97.134:1812 -> 213.73.109.167:27374
10/18-22:39:39.135360 [**] Possible trojan server activity [**] 10.10.97.134:1824 -> 213.73.109.179:27374
10/18-22:39:39.199434 [**] Possible trojan server activity [**] 10.10.97.134:1831 -> 213.73.109.186:27374
10/18-22:39:39.230975 [**] Possible trojan server activity [**] 10.10.97.134:1834 -> 213.73.109.189:27374
10/18-22:39:57.381724 [**] Possible trojan server activity [**] 10.10.97.134:1852 -> 216.113.108.125:27374
10/18-22:39:57.462344 [**] Possible trojan server activity [**] 10.10.97.134:1859 -> 216.113.108.132:27374
10/18-22:39:59.464209 [**] Possible trojan server activity [**] 10.10.97.134:1863 -> 216.113.108.137:27374
10/18-22:39:59.561405 [**] Possible trojan server activity [**] 10.10.97.134:1873 -> 216.113.108.147:27374
10/18-22:39:59.608830 [**] Possible trojan server activity [**] 10.10.97.134:1877 -> 216.113.108.151:27374
10/18-22:39:59.624435 [**] Possible trojan server activity [**] 10.10.97.134:1878 -> 216.113.108.152:27374
10/18-22:40:02.460238 [**] Possible trojan server activity [**] 10.10.97.134:1868 -> 216.113.108.142:27374
10/18-22:40:02.508229 [**] Possible trojan server activity [**] 10.10.97.134:1861 -> 216.113.108.135:27374

Another internal IP seen scanning was 10.10.98.172. Traffic sample:

10/15-22:46:25.130272 [**] Possible trojan server activity [**] 10.10.98.172:2432 -> 212.253.124.13:27374
10/15-22:46:25.131494 [**] Possible trojan server activity [**] 10.10.98.172:2433 -> 212.253.124.14:27374
10/15-22:46:25.403451 [**] Possible trojan server activity [**] 10.10.98.172:2447 -> 212.253.124.28:27374
10/15-22:46:25.531488 [**] Possible trojan server activity [**] 10.10.98.172:2468 -> 212.253.124.49:27374
10/15-22:46:25.595166 [**] Possible trojan server activity [**] 10.10.98.172:2475 -> 212.253.124.56:27374
10/15-22:46:27.956293 [**] Possible trojan server activity [**] 10.10.98.172:2434 -> 212.253.124.15:27374
10/15-22:46:27.980799 [**] Possible trojan server activity [**] 10.10.98.172:2442 -> 212.253.124.23:27374
10/15-22:46:27.980861 [**] Possible trojan server activity [**] 10.10.98.172:2425 -> 212.253.124.6:27374
10/15-22:46:29.389788 [**] Possible trojan server activity [**] 10.10.98.172:2488 -> 212.253.124.69:27374
10/15-22:46:29.550461 [**] Possible trojan server activity [**] 10.10.98.172:2512 -> 212.253.124.93:27374
10/15-22:46:32.175051 [**] Possible trojan server activity [**] 10.10.98.172:2490 -> 212.253.124.71:27374
10/15-22:46:32.352011 [**] Possible trojan server activity [**] 10.10.98.172:2500 -> 212.253.124.81:27374
10/15-22:46:32.352080 [**] Possible trojan server activity [**] 10.10.98.172:2502 -> 212.253.124.83:27374
10/15-22:46:32.496225 [**] Possible trojan server activity [**] 10.10.98.172:2524 -> 212.253.124.105:27374
10/15-22:46:32.527359 [**] Possible trojan server activity [**] 10.10.98.172:2536 -> 212.253.124.117:27374

Both of these IP's could be compromised boxes. It is recommended that these two boxes be taken off the wire and be thoroughly investigated. If they are not compromised, then it could be someone internal using a tool to scan for SubSeven. If this is the case then the acceptable use policy should be adhered to and disciplinary action needs to be taken.

Another recommendation would be to block port 27374 and other known trojan ports at the firewall.

Scans:

There were a few IP's in the scan data that needs to be pointed out.

Internal IP 10.10.160.114 was seen UDP scanning heavily over the five day period.

Source	# Alerts (sig)	# Dsts (sig)
10.10.160.114	273576	9954

I would recommend that this machine be looked into. Most of the scanning was done on port 27005, which is the port for FLEX-LM. Most likely it just happened to be scanning on this port. This machine needs to be checked for improper configuration or possible compromise.

Most of the rest of the scans in the data revolved around incoming ftp port 21 scans and single instances of malformed packets. Nothing to be too worried about as long as the firewall is configured properly and the ports that need to be turned off are.

Recommendations

The following are my recommendations:

1. Be sure that the internal hosts are at the current level of Operating System patching.
2. Check the internal hosts for improper software that might be against the company's acceptable use policy, i.e. Gnutella, KaZaa.
3. Check all defense devices (firewall, routers) for correct software patching.
4. Check Security policies on perimeter defense devices for correct Access Lists and Rulebase settings.
5. Assuming this IDS is running in the DMZ, consider placing IDS systems on the Internal network to pick up activity that penetrates the Perimeter Security (Firewalls / Routers) and makes it into the private LAN.
6. It seems from all the traffic seen from the Watchlist 000220 IL-ISDNNET-990517 that this particular network could be known as an exploitable network. It could be worse since there is a lack of packet filtering.
7. A stateful firewall should be used so that packets can be filtered better and control access into the network.
8. Stricter user policies should be put in place. A policy for how peer-to-peer and IRC should be handled.
9. A regularly scheduled vulnerability scan of the network should be put in place.

A list of external IP addresses that should be closely monitored:

212.29.222.114, from Israel, should be monitored due to the fact of the heavy Code Red scanning. It might be wise to go ahead and block this IP at the firewall, especially if this IP is compromised in some way. Whois information below:

```
inetnum:        212.29.222.96 - 212.29.222.127
netname:        EFRAT-1
descr:          Efrat
country:        IL
admin-c:        OH624-RIPE
tech-c:         DB1523-RIPE
status:         ASSIGNED PA
mnt-by:         RIPE-NCC-NONE-MNT
changed:        dbenjamin@barakitc.co.il 19981018
source:         RIPE


person:         Oleg Hanokov
```

```
address:        Efrat
address:        Israel
phone:          + 972 3 6452222
fax-no:         + 972 3 6452333
nic-hdl:        OH624-RIPE
notify:         dbenjamin@barak.net.il
changed:        dbenjamin@barak.net.il 19981119
source:         RIPE


person:         Dana Benjamin
address:        Barak I.T.C
address:        15 Hmelacha St Rosh Ha'ayin
address:        Israel 48091
phone:          + 972 3 9001102
fax-no:         + 972 3 9001515
e-mail:         dbenjamin@barakitc.co.il
nic-hdl:        DB1523-RIPE
changed:        dbenjamin@barakitc.co.il 19981126
source:         RIPE
```

211.93.80.164, from China, should also be monitored closely, due to the fact that it was seen scanning heavily for Code Red II. Blocking this IP should also be a consideration. Whois information below:

```
inetnum:      211.93.80.0 - 211.93.95.255
netname:      LNSY
descr:        sy city ,Liao Ning province POP, China.P.R
country:      CN
admin-c:      XL31-AP
tech-c:       XL31-AP
mnt-by:       MAINT-CNNIC-AP
changed:      wangch@cnnic.net.cn 20000401
source:       APNIC

person:       XiaoMing Li
address:      6F Office Tower 3, Henderson Centre, Beijing China
country:      CN
phone:        +86-10-65181800-291
fax-no:       +86-10-65181800-777
e-mail:       lxmlxm@public3.bta.net.cn
nic-hdl:      XL31-AP
mnt-by:       MAINT-CNNIC-AP
changed:      wangch@cnnic.net.cn 20000331
source:       APNIC
```

63.125.75.163 should also be monitored for the Code Red exploit. Again, this address should be considered to be block if the scans remain constant. Whois information is below:

```
City of Baldwin Park (NETBLK-UU-63-125-75-160)
   14403 E. Pacific Ave
   Baldwin Park, CA 91706
   US

   Netname: UU-63-125-75-160
```

```
Netblock: 63.125.75.160 - 63.125.75.191

Coordinator:
   Yeung, John  (JY176-ARIN)  jyeung@baldwinpark.com
   626-960-4011x129

Record last updated on 10-Nov-2000.
Database last updated on  10-Dec-2001 19:56:24 EDT.
```

212.179.x.x, because of the amount of traffic seen coming from this particular subnet. It is on the watchlist and should consider being blocked. Whois information below:

inetnum: 212.179.0.0 - 212.179.1.255
netname: AREL-NET
descr: arel-net
country: IL
admin-c: TP1233-RIPE
tech-c: TP1233-RIPE
status: ASSIGNED PA
notify: hostmaster@isdn.net.il
mnt-by: RIPE-NCC-NONE-MNT
changed: hostmaster@isdn.net.il 19990624
source: RIPE

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

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

217.136.4.18, from Belgium, was seen heavily scanning for ftp services. Not that this is active targeting, it should be closely monitored if scanning continues. Whois information below:

```
inetnum:        217.136.0.0 - 217.136.31.255
netname:        BE-SKYNET-20010125
descr:          Belgacom Skynet SA/NV
descr:          ADSL BAS bru-stro TL GO/PLUS
country:        BE
admin-c:        SN2068-RIPE
tech-c:         SN2068-RIPE
```

```
rev-srv:        ns.ripe.net
rev-srv:        ns1.skynet.be
rev-srv:        ns2.skynet.be
rev-srv:        ns3.skynet.be
rev-srv:        ns4.skynet.be
status:         ASSIGNED PA
mnt-by:         SKYNETBE-MNT
changed:        piet@skynet.be 20010203
source:         RIPE

route:          217.136.0.0/16
descr:          SKYNETBE-CUSTOMERS
origin:         AS5432
notify:         noc@skynet.be
mnt-by:         SKYNETBE-MNT
changed:        jfs@skynet.be 20010125
source:         RIPE

role:           Skynet NOC administrators
address:        Belgacom Skynet  SA/NV
address:        rue colonel Bourg 124
address:        B-1140 Brussels
address:        Belgium
phone:          +3227061311
fax-no:         +3227269311
email:          ripe@skynet.be
admin-c:        JFS1-RIPE
tech-c:         PDH16-RIPE
nic-hdl:        SN2068-RIPE
remarks:        ----------------------------------------
remarks:        Abuse notifications to: abuse@skynet.be
remarks:        Network problems to: noc@skynet.be
remarks:        Peering requests to: peering@skynet.be
remarks:        ----------------------------------------
notify:         noc@skynet.be
mnt-by:         SKYNETBE-MNT
changed:        ripe@skynet.be 20010907
source:         RIPE
```

Last Updated: May 23rd, 2017

Upcoming Training

Mentor Session - SEC503	Oceanside, CA	May 29, 2017 - Jun 29, 2017	Mentor
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC503: Intrusion Detection In-Depth	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS vLive - SEC503: Intrusion Detection In-Depth	SEC503 - 201709,	Sep 11, 2017 - Oct 18, 2017	vLive
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
Baltimore September 2017 - SEC503: Intrusion Detection In-Depth	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Scottsdale SEC503	Scottsdale, AZ	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS October Singapore 2017	Singapore, Singapore	Oct 09, 2017 - Oct 28, 2017	Live Event
Community SANS Ottawa SEC503	Ottawa, ON	Oct 16, 2017 - Oct 21, 2017	Community SANS
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC503: Intrusion Detection In-Depth	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SIEM & Tactical Analytics Summit & Training	Scottsdale, AZ	Nov 28, 2017 - Dec 05, 2017	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced