Global Information Assurance Certification Paper

Intrusion Detection In Depth

_____

GIAC Certified Intrusion Analyst Practical (GCIA) Version 3.0

SANS 2002 Miami
January 7- 14th

Martha Flick

_____

Table of Contents — Page Number

Assignment 1:

Protecting Servers From Attack - Vulnerability Scanning as a System Countermeasure

What is Vulnerability Scanning and Why it is used?

Vulnerability scanning is a process of using a commercial or open source product to probe a server for known vulnerabilities that leave it open to exploits and ultimately compromise. You may think to yourself "Why bother? I've patched my server." Are you sure you didn't miss anything? In a changing IT world, it's a sure bet that what was secure today, may not be tomorrow. Network Magazine's article on Vulnerability Assessment Tools (Ref 1.) puts it this way:

> "As the complexity of an enterprise network increases, so do its vulnerabilities….Scanning is one way to root out possible weak points in your network…. In fact, would-be intruders use these tools to scope out a network before attacking, so scanning is also a proactive security measure that lets you find the chinks in your armor before someone else does..…A vulnerability scan takes a hacker's-eye view of your network"

Scanning for vulnerabilities could range from probing for listening ports, incorrect permissions, guessable accounts and passwords, trojans or backdoors, known exploits of the operating system or application, and/or known exploits of services running on the operating system, etc.

Without question, scanning should be performed on servers placed into de-militarized zones (DMZ's) exposed to the Internet. Since DMZ servers are the company's public web, ftp or B2B servers, they receive numerous connections from untrusted outside sources. That leads to the perception that they are prime targets, which *cannot be ignored*. Once compromised, they become the entry point into the network. Several well known compromises have been a PR nightmare for their corporate owners.

"EggHead Hacked and Cracked"
"Egghead.Com Sales Soft in Q4"
"Lengthy Egghead Investigation Costs Banks Millions"
"Microsoft Downplays Hack Attack"
"Microsoft Hacked, FBI Steps In"

However DMZ servers are not the only targets; scanning can also benefit a company's internal servers. This would appear to be common sense to any security analyst involved in this type of activity; however getting corporate commitment of resources is whole another issue. Case in point, after CodeRed hit the Internet during the summer of 2001, CAIDA.ORG did an excellent in-depth summary and analysis. Ref 7. , "CAIDA Analysis of Code-Red, Code-Red Worms, A Global Threat".

> "On June 18, 2001 eEye released information about a buffer-overflow vulnerability in Microsoft's IIS webservers. The remotely exploitable vulnerability was discovered by Riley Hassell. It allows system-level execution of code and thus presents a serious security risk…..
>
> On July 12, 2001, a worm began to exploit the aforementioned buffer-overflow vulnerability in Microsoft's IIS webservers. Upon infecting a machine, the worm checks to see if the date (as kept by the system clock) is between the first and the nineteenth of the month. If so, the worm generates a random list of IP addresses and probes each machine on the list in an attempt to infect as many computers as possible….

> At approximately 10:00 UTC in the morning of July 19th, 2001 a random seed variant of the Code-Red worm (CRv2) began to infect hosts running unpatched versions of Microsoft's IIS webserver……
>
> On July 19, 2001 more than 359,000 computers were infected with the Code-Red (CRv2) worm in less than 14 hours. At the peak of the infection frenzy, more than 2,000 new hosts were infected each minute. 43% of all infected hosts were in the United States, while 11% originated in Korea followed by 5% in China and 4% in Taiwan. The .NET Top Level Domain (TLD) accounted for 19% of all compromised machines, followed by .COM with 14% and .EDU with 2%. We also observed 136 (0.04%) .MIL and 213 (0.05%) .GOV hosts infected by the worm.

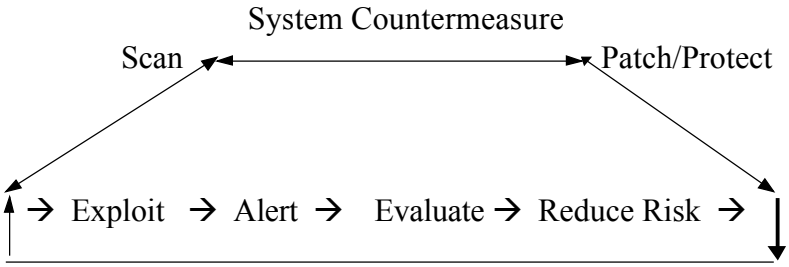
Amazing. All because an exploit in IIS had gone unpatched and left servers vulnerable. If a corporation has both internal and DMZ web servers but focuses only on scanning the DMZ servers, the odds of missing same vulnerabilities against internal servers may occur. Since CodeRed infected servers scan for vulnerable hosts, it would take only one compromised internal server to start a chain reaction against others. By the way, that's a whole lot of traffic crossing your network sucking up bandwidth. A simple scan of those servers would have confirmed that proper patches were missing exposing them to the buffer overflow exploit.

The Computer Security Institute conducts a computer crime survey ("Financial losses due to Internet intrusions, trade secret theft and other cyber crimes soar " Ref 8.), and has posted information on crime statistics for the last six years. The last survey for the year 2001, indicates 85% of the 538 companies polled detected security breaches over 12 months with 64% of the respondents acknowledging losses. Forty percent of the respondents detected breaches initiated externally to their network. This is up from 25% reported by CSI in the previous year's survey . So, although outside attacks are most notorious and public, there is still a high possibility that insiders instigate compromises and data loss against corporate systems. This activity is very different from the automated CodeRed style of exploit because the potential source is trusted and may already have administrative access. Scanning of internal servers is this case may help secure the server but other countermeasures are needed to monitor inappropriate use by authorized individuals.

So if the information presented above doesn't give you enough ammunition to go to management for permission to scan corporate systems, then why else would you do it? Scanning provides an *initial measure* of a secured server; at least to the best knowledge of the scan software and the analyst at that time. Without it, a server on a DMZ or internal segment may be a ticking time bomb waiting for someone to light the fuse. Although scanning may help to reduce possible compromise it doesn't eliminate it all together. It is only one of several system countermeasures available to the security analyst.

How is Vulnerability Scanning part of System Countermeasures and what are they?

Alerts that are generated by Intrusion Detection Systems (IDS) need to be prioritized in terms of risk to the company. These alerts are triggered because of exploits run against company owned network devices like servers that may be vulnerable to that exploit. So by evaluating risk, the analyst can prioritize what alert to investigate first. How does scanning relate to risk? If used by the analyst, it can reduce risk by enhancing system countermeasures. That in turn, protects servers and other network devices from exploits, minimize alerts, reduces more risk and makes the analysts job more efficient. The cycle of exploits, alerts and risk reduction will always



remain, because new exploits continue to be found. It's just that the cycle is minimized when scanning for vulnerabilities becomes part of it.

First, let's think about quantifying risk. Severity is one measure used by the GIAC approach (Ref 11, 12) to evaluate and quantify risk. It gives the analyst a means of associating risk with attacks that exploit system vulnerabilities. System countermeasures are one of four categories used to calculate Severity.

Severity = (Criticality + Lethality) - (System countermeasures + Network countermeasures) (Ref 11, 12)

Criticality and lethality refer to the target's importance and possible damage. Opposite to criticality and lethality is the ability to protect the potential target through system and network countermeasures. System countermeasures are really those practices in place in which the administrator patches, fixes, locks down, configures both the OS and the application to effectively protect both the server and the data it may house. So, scans provide confirmation that effective system countermeasures are in fact, in place and working. If a scan can't find an exposure, exploit, listening port etc, it is one door shut to potential hackers. Admittedly, a secured system can be compromised by insiders with proper authority. However without planned scanning, the security analyst may have a blind view of servers or the network they are responsible to protect.

What information will scanning provide?

- Confirmation that all appropriate operating system service packs and patches have been applied and are functional. In some cases, applying packs or patches out of order lead to the pack or patch being ineffective or even reversing previous security fixes.
- Configuration of the server operating system was properly done. That is, there are no unneccessary listening ports and/or services that might easily expose it to reconnaissance probes or exploits.
- There are no easily guessable accounts or blank passwords, especially those with administrative access. For in-depth analysis this requires additional software (readily available to both the hacker and security administrator from various internet sources).
- Any loaded applications have been properly configured to prevent unauthorized access or exploitation.

- Backdoors or Trojans have not been placed onto the server and such compromised it.

In reality, vulnerability scanning is critical not only after the initial build of OS and applications but during the machine's life. New exploits against OS's and applications are constantly being discovered. That means that a server thought to be safely secured after its initial install, may later be exploited if not properly maintained. It is all too easy to put off a patch or fix for a more convient time when the admin isn't as busy, only to have that patch or fix be forgotten in the process.

Although IDS alerts the analyst that inappropriate actions are occurring against a server or servers, system countermeasures protect not only a server but the corporation's network. Once a server is compromised, it becomes a stepping stone for the hostile party to continue to penetrate into a network. Vulnerability scans are a confirmation of that specific system countermeasures are in place and thus can be considered a countermeasure as well.

Where do I start ?

There are many different products available. There are even a wide range of companies that provide what are called vulnerability assessments of servers and devices for a fee. Usually this entails X number of scans against a certain IP segment or number of servers over X period of time. Some security companies go even further and provide risk assessments of a corporation's DMZ or internal network. A vulnerability scan of the servers is typically a small part of the overall risk assessment process. Of course risk assessments are very valuable but beyond the focus of this article. What is important, is that with the abundance and ease of scanning software available, commercial and open source, it makes sense to at least consider setting up scanning software to confirm that servers are secure. Ask yourself some simple questions:

- *What are your resources?* If you don't have the people or hardware or time available then those outside companies that provide scanning services may be a possibility. More expensive than the do-it yourself approach and still requires resources to take action after scanning is done.

- *What is your budget?* Both commercial products like ISS's Internet Scanner (Ref 9) as well as open source like Nessus (Ref 10) are very good choices. It's a budget and personal preference choice.

- *Create a process agreeable to all parties?* It doesn't do you any good to scan if the administrators of the servers do not take action based on the results. Get all and any parties (including management) to commit up front that scans results should be reviewed and the appropriate actions taken

- *Think about a schedule.* One scan doesn't mean the server is protected forever. Find an agreeable schedule (monthly, quarterly) and time of day or day of the week to run scans.

One particular example…

This document does not compare or rank scanners. It mentions one familiar to the author to show a generally methodology for vulnerability scans. Internet Security Systems Internet Scanner product is a basic multipurpose scanner. It typically is used to determine if the OS and non-database applications have been properly patched and configured. To date, it has over 800+ exploits it potentially can look for. It is easy to use and configure, is customizable and can export the data in htm, .doc or .dif format. Technical support is good and easy to get. On the down size, it can be expensive and is licensed per IP address, which can add up in a hurry for those with a limited budget.

It's a process in the making….Plan first!!

Prior to scanning…
Determine whether the scanner is on a dedicated system. While the inclination may be to install on you own PC, a word of caution here. It is advantageous to set aside a moderately powerful PC that is dedicated to this task. Why? Because although a single scan of a newly built server is straightforward; at some later date, all those previously scanned machines may need to be re-scanned. You may even decide to start scanning IP segments of your network (PC's) for possible exploits or trojans. The point is, realistically, if your personal PC is scanning several servers or even a segment of the network, you're not going to be using it during that time for anything requiring a lot of resources. So plan on **how your going to scan, when, what, and how much** at any one time.

After installation you're not yet ready…
Installation of any product is generally well documented and straightforward but be sure any plugins that may extend the software's ability have also been installed. For example, ISS's product also uses what is called Xpress Updates from the vendor. These are the additional exploits that have come out in the wild since the original release and which the vendor has packaged as a means to update the exploit signatures.

Decide on what to scan for and how to customize the software…
Scan policies used by scanners can be customizable according to the operating system and applications loaded. Why scan a Unix server for NT vulnerabilities or vice versa? Make sure if the server your scanning is running Sendmail you have SMTP vulnerabilities configured in the policy your using. If its IIS running with Coldfusion, the scanning policy will be totally different.

Plan and communicate for possible downtime…
If a major vulnerability is discovered and the server responds to it, all bets are off. The server may have to be rebooted, patched and then re-scanned.

Got the scan, What next?

Review and forward as necessary..

After scanning, create an electronic or printout of the results. It will describe the problem and may list possible actions to take. Some research on odd results may be required. Result summaries are generally all that is needed however; there are times when a new exploit or some unusual service is running where the verbose information is helpful. Preview and determine if the verbose information is necessary. Send it to the appropriate parties. Confirm after a pre-agreed time that remedial action has been taken.

ReScan…
If vulnerabilities were found, re-scan! When the scan is shows no unacceptable vulnerabilities, your done until the next scheduled scan. That's right, the next scheduled one (monthly, quarterly, etc).

The Big Picture…

The first couple of scans really give both the system administrator and the security administrator valuable information. It ends up creating a checklist or process of securing the OS and applications. After a while, the order and type of patches/fixes become routine. I have found that subsequent scans of newly built servers result in less exploits and vulnerabilities being found in future servers, because previous provide information on how and what the process should be for securing a server.

Coordinate with the network administrator (if its not you) who is responsible for maintaining the OS and/or applications. Once everyone involved understands the process and their part, scanning can become a beneficial and important system countermeasure to protecting servers on the network.

A possible checklist resulting from vulnerability scans is as follows:

- Build the OS.
- Use previous vulnerability scans to provide standard guideline of service packs, patches and general configuration to be applied.
- Apply the packs, patches.
- Setup the proper user accounts.
- Add the application and its patches.
- Run a vulnerability scan at a scheduled time.
- Review results and share with appropriate parties.
- Fix any resulting vulnerability listed.
- Re-run the same scan to confirm the vulnerability no longer exists.
- As new vulnerabilities are discovered and the scanning program is updated, run new scans to check that the server to confirm it is still secure.
- Additional scans may be used on a monthly or quarterly basis to test the server, as its applications or code is modified and upgraded by the applications developers.

References:

1. Conry-Murray, Andrew, "Vulnerability Assessments", Network Magazine, URL: http://www.networkmagazine.com/article/NMG20010321S0005 , April 05th , 2001.
2. Enos, Lois, "EggHead Hacked and Cracked", Ecommerce Times, URL http://www.ecommercetimes.com/perl/story/6286.html , December 22nd, 2000.
3. King, Carol, "Egghead.Com Sales Soft in Q4", URL http://www.internetnews.com/ec-news/article/0,,4_571601,00.html , January 26th, 2001.
4. Lemos, Robert, "Lengthy Egghead Investigation Costs Banks Millions", URL http://news.com.com/2009-1017-250745.html?legacy=cnet&tag=st.ne.1007.thed.sf , January 9th, 2001.
5. McDonald, Tim, "Microsoft Downplays Hack Attack", Ecommerce Times, URL http://www.ecommercetimes.com/perl/story/4679.html4 , October 30, 2000.
6. Festa, Paula, "Microsoft Hacked, FBI Steps In", URL http://news.com.com/2100-1001-247716.html?legacy=cnet , October 27th, 2000
7. Shannon, Colleen, CAIDA.ORG web site, "CAIDA Analysis of Code-Red, Code-Red Worms, A Global Threat", http://www.caida.org/analysis/security/code-red/#crii , 2001.
8. Raplaus, Patricia, "Financial Losses Due to Internet Intrusions, trade secret theft and other cyber crimes soar", URL http://www.gocsi.com/prelea/000321.html March 3rd, 2001.
9. Internet Security Systems Internet Scanner, URL http://www.iss.net/products_services/enterprise_protection/vulnerability_assessment/scanner_internet.php .
10. Nessus, URL http://www.nessus.org .
11. Northcutt, Steven, and Novak, Judy, "Network Intrusion Detection, An Analyst's Handbook" New Riders, pp. 151-153 2001.
12. SANS, Intrusion Detection In Depth Training Course, SANS Miami, January 7th, 2002.

Assignment 2: Network Traces and Analysis

Detect #1 Nimda attack, port 80 (Back to top)

1. Source of Trace.
This trace was retrieved from the outside interface of my company's firewall using snoop. It is fairly long but it is easier to recognize the attacker's method by showing more than a couple of lines of detect. Some repeating packets were left out purposely.

{This trace has been sanitized to remove company information}
0.00151 202.108.60.148 -> **X.X.X.**7 HTTP GET /scripts/root.exe?/c+dir HTTP/1.0
0.00750 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 404 Object Not Found
0.00006 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3228
0.00190 202.108.60.148 -> **X.X.X.**7 HTTP GET /MSADC/root.exe?/c+dir HTTP/1.0
0.00488 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 404 Object Not Found
0.00007 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3241
0.00300 202.108.60.148 -> **X.X.X.**7 HTTP GET /c/winnt/system32/cmd.exe?/c+dir HTTP/1.0
0.00517 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 404 Object Not Found
0.00006 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3250
0.00295 202.108.60.148 -> **X.X.X.**7 HTTP GET /d/winnt/system32/cmd.exe?/c+dir HTTP/1.0
0.00485 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 404 Object Not Found
0.00012 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3255
0.00339 202.108.60.148 -> **X.X.X.**7 HTTP GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0
0.00225 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 500 Server Error
0.00222 202.108.60.148 -> **X.X.X.**7 HTTP GET
/_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.
0.00247 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 500 Server Error
0.00028 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3269
0.00241 202.108.60.148 -> **X.X.X.**7 HTTP GET
/_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.
0.00231 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 500 Server Error
0.00033 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3280
0.00536 202.108.60.148 -> **X.X.X.**7 HTTP GET
/msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/sys
0.00251 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 500 Server Error
0.00028 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3294
0.00150 202.108.60.148 -> **X.X.X.**7 HTTP GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir HTTP/1.0
0.00516 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 404 Object Not Found
0.00005 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3307
0.00269 202.108.60.148 -> **X.X.X.**7 HTTP GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir HTTP/1.0
0.00496 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 404 Object Not Found
0.00006 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3316
0.00203 202.108.60.148 -> **X.X.X.**7 HTTP GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir HTTP/1.0
0.00492 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 404 Object Not Found
0.00202 202.108.60.148 -> **X.X.X.**7 HTTP GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir HTTP/1.0
0.00484 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 404 Object Not Found
0.00006 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3331
0.00536 202.108.60.148 -> **X.X.X.**7 HTTP GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0
0.00187 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 500 Server Error
0.00021 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3337
0.00148 202.108.60.148 -> **X.X.X.**7 HTTP GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir HTTP/1.0

0.00178 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 500 Server Error
0.00018 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3349
0.00238 202.108.60.148 -> **X.X.X.**7 HTTP GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0
0.00176 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 500 Server Error
0.00021 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3359
0.00306 202.108.60.148 -> **X.X.X.**7 HTTP GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir HTTP/1.0
0.00206 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 500 Server Error
0.00019 **X.X.X.**7 -> 202.108.60.148 HTTP R port=3371

2. Detect was generated by:
The detect was generated by snoop. It is a packet-sniffing tool available on Solaris systems. I use it to diagnose and fix problems on our firewalls; it provides great diagnostics when determining how and if connections are occurring or in some cases, not occurring. Because it is a packet-sniffing tool, it also can be used as an intrusion detection method. I was actually tinkering with the options when I discovered this trace. This is the command I issued to record to a file.

snoop -ta -c 4000 -o ids020102.log -d qfe1 "!broadcast and !multicast"

(I basically wanted to focus on everything but the broadcast and multicast traffic, which can be very chatty.) Here's how the switches I used tell snoop what to do.

-ta	is the timestamp in absolute (I could've choose r -relative or d-delta which is the default, but hey, if you can't test options to see what they do, what's the use in having them.)
-c 4000	is to indicate to capture 4000 packets and then stop. Any number can be used after the -c. If the c option is not used snoop continues to run until the disk if full or it is stopped. I chose to capture several minutes of port 80 traffic but not get too large in size.
-o ids020102.log	read to file named ids020102.log
-i ids020102.log	display file named ids020102.log
-d qfe1	device that happens to be our outside interface.
"port 80"	capture only HTTP traffic.
"!broadcast"	not broadcast traffic.
"!multicast"	not multicast traffic.

Here's the command to parse the file for specific traffic.

snoop -i ids020102.log "port 80" | more

The command tells snoop to display the file called ids020102.log by screen page (using the pipe | and more command) and only show HTTP traffic (port 80).

The trace output is in the following format:
Time Source IP Traffic direction Dest IP Response
0.00300 202.108.60.148 -> **X.X.X.**7 HTTP GET /c/winnt/system32/cmd.exe?/c+dir HTTP/1.0

3. Probability the source address was spoofed:
No. The probe/attack is for the purpose of determining if our server was compromised by CodeRed and then if so compromised, gaining access to CMD.exe on our server via the Unicode Transversal (Nimda combination). For the attack to be useful the source address would not be spoofed.

4. Description of attack:
The attack is a Nimba attack which uses a combination of CodeRed and the Unicode Transversal exploit. Basically, the source 202.108.60.148 is first probing to see if CodeRed has already compromised the web server; the initial probe is for **root.exe**. Root.exe is actually a copy of cmd.exe put into the webroot area of the server. Then the source is trying to gain access to CMD.EXE either in C:\ or D:\.

0.00151 202.108.60.148 -> **X.X.X.**7 HTTP GET /scripts/**root.exe**?/c+dir HTTP/1.0…..
0.00190 202.108.60.148 -> **X.X.X.**7 HTTP GET /MSADC/**root.exe**?/c+dir HTTP/1.0……
0.00300 202.108.60.148 -> **X.X.X.**7 HTTP GET /**c**/winnt/system32/cmd.exe?/c+dir HTTP/1.0….
0.00295 202.108.60.148 -> **X.X.X.**7 HTTP GET /**d**/winnt/system32/cmd.exe?/c+dir HTTP/1.0…..
It then appears to be attempting the unicode transversal exploit against the web server in an attempt to gain unauthorized access to cmd.exe on the server. For the uninitiated, the Unicode transversal exploit basically uses Unicode characters to back out of the web directory structure and into the directory of the operating system. In this case, access is attempted to \winnt\system32.

A series of various Unicode expressions are attempted to see if the web server will return the cmd.exe prompt to the attacker. An attempt is made to escape out of the web directory and into other directories on the server by using various strings including as an example, a MSADC "bypass" technique. URL ref.
Http://www.securiteam.com/windowsntfocus/Web_Server_Folder_Traversal_vulnerability_Patch_available_exploit_.html

Looking at the trace, it shows at least 14 different transversal attempts to gain access to cmd.exe. This would then allow further compromise of the server by letting the attacker run additional commands. The entire scripted attack occurred within a very short time window. Since our server was patched, it was sending error responses and resets back to the source IP. Of course, the scripted attack ignored the resets and proceeded.

0.00750 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 **404 Object Not Found**
0.00006 **X.X.X.**7 -> 202.108.60.148 HTTP **R** port=3228
…..
0.00206 **X.X.X.**7 -> 202.108.60.148 HTTP HTTP/1.1 **500 Server Error**
0.00019 **X.X.X.**7 -> 202.108.60.148 HTTP **R** port=3371

5. Attack mechanism:
The Nimba attack mechanism depends on two items. Unpatched IIS servers exposed to buffer overflows against the IIS indexing service (.ida and .idq files), and then unpatched IIS servers that can be exploited with the Unicode transversal. The combination was a powerful one-two punch

in compromising and gaining unauthorized access to the server.

Discovery of root.exe on an IIS web server is indicative of infection by CodeRed and its variants, which swept across the Internet during late 2001. It exploited unprotected IIS web servers by using a buffer overflow against the IIS indexing service or .ida or idq files. This leaves it open to attack by Nimda, which searches for CodeRed exposure and then attempts the Transversal exploit.

One easy way to see who is attacking your server is to go look at the W3SVC log files created by IIS4 and IIS5. Normally, there are logfiles in \winnt\system32\logfiles directory although IIS can be reconfigured to put the logs where ever desired. Generally, a file is created for each day. If an infected server is attacking your server it will show in this file.

6. Correlation's:

It appears as though a Nimda infected server from somewhere in China is trying to attack our server. The attack appears to be coming from Bejing, China. Output from Arin WHOIS provided information on the owner of the IP address:

```
Asia Pacific Network Information Center (APNIC2)
 These addresses have been further assigned to Asia-Pacific users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.    AU
   Netname: APNIC-CIDR-BLK
   Netblock: 202.0.0.0 - 203.255.255.255
   Maintainer: AP
   Coordinator:
      Administrator, System  (SA90-ARIN)  [No mailbox]
      +61-7-3367-0490
   Domain System inverse mapping provided by:
   SVC00.APNIC.NET          202.12.28.131
   NS.APNIC.NET             203.37.255.97
   NS.TELSTRA.NET           203.50.0.137
   NS.RIPE.NET              193.0.0.193
```

Output from APNIC produced the following:

```
inetnum             202.108.60.0 - 202.108.60.255
netname             JXHERO-NET
descr               Beijing Jingxin Hero
descr               Telecommunication Network CO.,Ltd.
country             CN
admin-c             XN3-AP, inverse
tech-c              XN3-AP, inverse
mnt-by              MAINT-CHINANET-BJ, inverse
changed             hostmast@publicf.bta.net.cn 20010924
source              APNIC

person              Xu NingTao, inverse
address             No.2 Xizhaosi Street Chongwen
address             District Beijing,100061
phone               +86-10-67180828
fax-no              +86-10-67180827
e-mail              sd_hunter@263.net, inverse
```

```
nic-hdl              XN3-AP, inverse
mnt-by               MAINT-CHINANET-BJ, inverse
changed              hostmast@publicf.bta.net.cn 20010924
source               APNIC
```

Correlation's on Nimba , CodeRed and the Unicode Transversal Exploit.
Anti-virus Vendors (I tend to use several for cross-reference whenever I look up virus activity). Trend's site showed literally the directory pattern of attack seen in the trace. McAfee and Symantec describe payloads. All AV sites provide removal instructions and give links to Microsoft for the necessary patches to protect from the exploit.

- TrendMicro Virus Library, Nimda, http://www.antivirus.com/vinfo/virusencyclo/default5.asp?VName=PE_NIMDA.A-O&VSect=T
- McAfee Virus Encyclopedia, Nimda, http://vil.nai.com/vil/content/v_99209.htm
- Symantec, Nimda, http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html

General background information falls into several categories. To have a basic understanding of how Nimda functions, some information on Buffer Overflows in IIS leading to CodeRed and Transversal exploits is helpful.

- CERT® Advisory CA-2001-26 "Nimda Worm", URL: http://www.cert.org/advisories/CA-2001-26.html, Original Release date September 18, 2001.
- CERT® Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL", URL: http://www.cert.org/advisories/CA-2001-19.html , Original release date: July 19, 2001, Last revised: January 17, 2002
- CERT® Incident Note IN-2001-09,"Code Red II:" Another Worm Exploiting Buffer Overflow In IIS Indexing Service DLL ", URL: http://www.cert.org/incident_notes/IN-2001-09.html , AUgust 6, 2001.
- SecuriTeam Web Site, "Web Server Folder Traversal vulnerability (Patch available, exploit)", URL: http://www.securiteam.com/windowsntfocus/6U00B2000A.html, 10/17/2000
- Cert Vulnerability Note VU#111677, "Microsoft IIS 4.0 / 5.0 vulnerable to directory traversal via extended unicode in url (MS00-078)", URL: http://www.kb.cert.org/vuls/id/111677 , 10/10/2000.
- Mitre Web Site, Common Vulnerabilities and Exposures (CVE), CVE-2001-0333, Directory traversal vulnerability in IIS 5.0 and earlier allows remote attackers to execute arbitrary commands by encoding .. (dot dot) and "\" characters twice. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0333
- Mitre Web Site, Common Vulnerabilities and Exposures (CVE), CVE-1999-1011, "The Remote Data Service (RDS) DataFactory Component of Microsoft Data Access Components (MDAC) in IIS 3.x and 4.x exposes unsafe methods, which allows remote attackers to execute arbitrary commands", URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-1011
- Mitre Web Site, Common Vulnerabilities and Exposures (CVE), CVE-2000-0884, "IIS 4.0 and 5.0 allows remote attackers to read documents outside of web root and possibly execute arbitrary commands …" URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0884
- Microsoft Security Bulletin, MS01-026, "14 May 2001 CUMMULATIVE Patch for IIS" http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/security/bulletin/MS01-026.asp

7. Evidence of active targeting:
Most definitely. The probe repeated seeks **X.X.X.**7 to send its requests to.

8. Severity:
(Criticality + Lethality) - (System Countermeasures + Net Countermeasures) = Severity

Criticality = 5	It's an important web server**.**
Lethality= 5	If the exploit worked, our server would be compromised**.**
System Countermeasures = 5.	The server is fully patched**,** at least for now.
Net Countermeasures = 2.	Although the ruleset on the firewall is restrictive, it is configured to allow port 80 traffic. It can't distinguish between "good" port 80 and "bad" port 80. It's just allows port 80 web traffic.

Severity= 2.

9. Defensive recommendation
It is important to know and maintain patch levels on DMZ servers. We were protected this time because we had patched and secured IIS. Continual maintenance of this product is a necessity. Would recommend additional tools like signature based HIDS, file integrity checking tools.

10. Multiple choice test question:
What is a signature file found on CodeRed infected servers?
a) \winnt\system32\root.bat
b) \winnt\system32\root.cmd
c) \inetpub\scripts\root.exe
d) \inetpub\scripts\cmd.exe

Answer c)

Detect#2 FTP scan, port 21

(Back to top)

1. Source of Trace.
Network owned by employer.

2. Detect was generated by:
Generated by firewall logs from four different Checkpoint firewalls. Used command line to read current log.

fw log -nc drop -b stime etime > file where

-n don't resolve names. (This means you don't wait forever to see the log output).
-c action to focus on . (Possibilities are "accept", "drop", "reject", "encrypt").
-b stime etime (start and end time of log you want to look at. Put it in military format because that's how it is reported.)

file send results to "file".

Format below is: (I didn't show all possible fields,.)

{This trace has been sanitized to remove company information}

Time	Action	Firewall	Interface	Protocol		Src IP		Dst IP	Service Source Port, Packet Length and Rule Applied
Feb 11, 2002									
15:52:01	drop	**fw-2**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**3 service **ftp** s_port 4690 len 48 rule 21
15:52:01	drop	**fw-2**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**6 service **ftp** s_port 4693 len 48 rule 21
15:52:01	drop	**fw-2**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**7 service **ftp** s_port 4694 len 48 rule 21
15:52:01	drop	**fw-2**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**8 service **ftp** s_port 4695 len 48 rule 21
15:52:01	drop	**fw-2**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**9 service **ftp** s_port 4696 len 48 rule 21
15:52:01	drop	**fw-2**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**10 service **ftp** s_port 4697 len 48 rule 21
15:52:01	drop	**fw-2**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**11 service **ftp** s_port 4698 len 48 rule 21
15:52:01	drop	**fw-2**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**18 service **ftp** s_port 4716 len 48 rule 21
…..Continues……									
15:52:06	drop	**fw-4**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**224 service **ftp** s_port 1093 len 48 rule 18
15:52:06	drop	**fw-4**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**231 service **ftp** s_port 1100 len 48 rule 18
15:52:06	drop	**fw-4**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**235 service **ftp** s_port 1104 len 48 rule 18
15:52:06	drop	**fw-4**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**239 service **ftp** s_port 1108 len 48 rule 18
15:52:06	drop	**fw-3**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**237 service **ftp** s_port 1106 len 48 rule 18
15:52:09	drop	**fw-4**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**238 service **ftp** s_port 1107 len 48 rule 18
15:52:11	drop	**fw-3**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**232 service **ftp** s_port 1101 len 48 rule 18
…..Continues…..									
15:52:15	drop	**fw-1**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**12 service **ftp** s_port 4699 len 48 rule 21
15:52:15	drop	**fw-1**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**13 service **ftp** s_port 4700 len 48 rule 21
15:52:15	drop	**fw-1**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**15 service **ftp** s_port 4702 len 48 rule 21
15:52:15	drop	**fw-1**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**16 service **ftp** s_port 4703 len 48 rule 21
15:52:15	drop	**fw-1**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**17 service **ftp** s_port 4704 len 48 rule 21
15:52:15	drop	**fw-1**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**19 service **ftp** s_port 4717 len 48 rule 21
15:52:15	drop	**fw-1**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**20 service **ftp** s_port 4718 len 48 rule 21
15:52:15	drop	**fw-1**	>qfe1	proto	tcp	src	**24.29.102.6**	dst	**X.X.X.**21 service **ftp** s_port 4719 len 48 rule 21

…..Continues…..

3. Probability the source address was spoofed:

No. The hostile is doing recon and wants to see the response.

In fact, Nslookup and Whois produced information on the owner of the IP space.

```
ServiceCo LLC - Road Runner (NET-ROAD-RUNNER-1)
   13241 Woodland Park Road
   Herndon, VA 20171    US
   Netname: ROAD-RUNNER-1
   Netblock: 24.24.0.0 - 24.29.255.255
   Maintainer: SCRR
   Coordinator:
      ServiceCo LLC  (ZS30-ARIN)  abuse@rr.com
      1-703-345-3416
   Domain System inverse mapping provided by:
   DNS1.RR.COM                  24.30.200.3
   DNS2.RR.COM                  24.30.201.3
   DNS3.RR.COM                  24.30.199.7
   DNS4.RR.COM                  65.24.0.172
   Record last updated on 30-Aug-2001.
   Database last updated on  16-Feb-2002 19:55:51 EDT.
```

Nslookup shows:

> server dns1.rr.com

Default Server: dns1.rr.com
Address: 24.30.200.3
> set type=a
> 24.29.102.6
Server: dns1.rr.com
Address: 24.30.200.3
Name: 24-29-102-6.nyc.rr.com
Address: 24.29.102.6
Going to Http://www.rr.com produces a web site with the following information:

> "Road Runner is... a high speed, online service that provides lightening-fast access to the Internet as well as to unique broadband content and services. Road Runner is delivered to your computer over the same upgraded cable systems that currently bring cable television into your home."

Thanks, cable modem land…

4. Description of attack:
The hostile is doing reconnaissance to find servers responding to ftp requests on our DMZ and is scanning against several of our firewalls almost simultaneously. I would believe that once our FTP server is discovered, the hostile would then try to determine the OS, the type of ftp program and then focus on known exploits based on that information. There are numerous ftp exploits depending on the type of ftp program used (Wu-ftp vs ServU vs War-ftp etc).

5. Attack mechanism:
Any port scanner that allows the user to put in an IP range and port number is sufficient to produce this trace. There are numerous commercial and shareware versions. Nmap, Hping and Xprobe are some of the best sources and when scanning and/or OS fingering printing, they are not nearly as noisy as this trace.

Interestingly enough is the fact that the scan crossed multiple firewalls on our network. Our firewalls are built for redundancy, which is why the drops show for different rules. It actually crossed two different pairs of firewalls used for different functions. The firewalls share the same outside IP address space and are paired. Thus, I would expect fw-1 and fw-2 would combine to show the range of the scan and fw-3 and fw-4 combined to also show the same range since their outside IP is in the same public range. What seems a little odd is that I actually don't see a full scan (x.x.x.1- x.x.x.254) for each pair. Most of it was detected on fw-1 and fw-2. So we probably didn't catch the entire scan.

6. Correlations:
- Northcutt, Stephen and Novak, Judy, "Network Intrusion Detection, An Analyst's Handbook, 2nd Edition", New Riders, 2001, "Host Scan Against FTP", pp156.
- Northcutt, Stephen, Cooper, Mark, Fearnow, Matt, and Frederick, Karen, "Intrusion Signatures and Analysis", New Riders, 2001, Port Scans, pp272.

Interesting, in the above-mentioned reference from "Intrusion Signatures and Analysis", the goal

of a port scan is mentioned.

> "A port scan helps determine which services are accessible remotely and also to fingerprint the operating system on each host. These techniques were discussed in Chapter 8, "Network Mapping" and Chapter 9 "Scans that Probe Systems for Information."

I would add after the fingerprint then the test for exploits.

7. Evidence of active targeting:
I would yes and no. Yes, they obviously know our public IP range and are targeting it; perhaps they did some DNS queries against several of our domain names or Corporation. No, at the time of the scan they did not have a specific single server in mind.

8. Severity:
(Criticality + Lethality) - (System Countermeasures + Net Countermeasures) = Severity

Criticality = 3	It's a general scan**.**
Lethality= 3	A scan is going to happen; the problem is if your server is vulnerable to FTP exploits.
System Countermeasures = 2.	Although we do have an ftp server; it is patched and the OS has been hardened**.** We have even scanned it with ISS to determine it we could break into it
Net Countermeasures = 2.	Although the ruleset on the firewall is restrictive, it is configured to allow port 21 traffic. Like web traffic, it can't distinguish between "good" port 21 and "bad" port 21.

Severity= 2.

9. Defensive recommendation
I recommend patching and monitoring our FTP servers. We also scan our own servers for vulnerabilities to determine if there is any action (patch to be added, service exposed) to be taken to protect the server. Host based intrusion detection is critical; and forensic programs like Tripwire also help.

10. Multiple choice test question:
What is the purpose of a port scan?
a) To find live hosts listening for a specific service.
b) To potentially exploit live hosts listening for a specific service.
c) To fingerprint live hosts listening for a specific service.
d) all of the above.
e) none of the above.

Answer d)

Detect #3, ICMP activity,

(Back to top)

1. Source of Trace.

Network owned by employer.

2. Detect was generated by:

Generated by snoop on the outside of my firewalls. I used the following command :

snoop -o ids.log -d qfe1 "icmp"

Basically, it collected packets of icmp only traffic from the outside interface and dumped to a file called ids.log. I actually have found that windump and snort provide an easier view with which to sort so I converted the log to file that either could read. I used ethereal for Windows to read in the Sun snoop log and save to plibcap format.

From windump I issued a:
windump -vr d:\sans\ids.log -n "icmp and net 216.52" > icmp.txt

{This trace has been sanitized to remove company information}

```
20:06:23.899533 216.52.193.70 > X.X.X.29: icmp: echo request (DF) (ttl 52, id 0)
20:06:37.666768 216.52.41.72 > X.X.X.24: icmp: echo request (DF) (ttl 50, id 0)
20:06:45.972775 216.52.129.75 > X.X.X.30: icmp: echo request (DF) (ttl 50, id 0)
20:06:50.734959 216.52.193.67 > X.X.X.29: icmp: echo request (DF) (ttl 52, id 0)
20:07:48.306918 216.52.65.67 > X.X.X.29: icmp: echo request (DF) (ttl 54, id 0)
20:07:49.649915 216.52.94.72 > X.X.X.29: icmp: echo request (DF) (ttl 52, id 0)
20:07:49.863608 216.52.94.68 > X.X.X.29: icmp: echo request (DF) (ttl 56, id 0)
20:08:25.221381 216.52.193.69 > X.X.X.29: icmp: echo request (DF) (ttl 54, id 0)
20:08:45.457184 216.52.169.65 > X.X.X.29: icmp: echo request (DF) (ttl 54, id 0)
20:09:14.775394 216.52.193.75 > X.X.X.29: icmp: echo request (DF) (ttl 53, id 0)
20:09:44.435279 216.52.129.68 > X.X.X.12: icmp: echo request (DF) (ttl 56, id 0)
20:09:50.068418 216.52.129.69 > X.X.X.12: icmp: echo request (DF) (ttl 53, id 0)
20:10:35.684332 216.52.65.70 > X.X.X.30: icmp: echo request (DF) (ttl 54, id 0)
20:10:59.056043 216.52.129.67 > X.X.X.1: icmp: echo request (DF) (ttl 54, id 0)
20:11:08.024756 216.52.129.72 > X.X.X.10: icmp: echo request (DF) (ttl 54, id 0)
20:12:07.054193 216.52.94.70 > X.X.X.29: icmp: echo request (DF) (ttl 52, id 0)
20:12:27.925859 216.52.129.71 > X.X.X.12: icmp: echo request (DF) (ttl 55, id 0)
20:13:00.553295 216.52.129.69 > X.X.X.10: icmp: echo request (DF) (ttl 53, id 0)
20:13:54.170765 216.52.129.68 > X.X.X.10: icmp: echo request (DF) (ttl 56, id 0)
20:14:53.442090 216.52.65.68 > X.X.X.30: icmp: echo request (DF) (ttl 59, id 0)
20:15:04.319277 216.52.169.68 > X.X.X.29: icmp: echo request (DF) (ttl 55, id 0)
20:15:08.571823 216.52.41.68 > X.X.X.24: icmp: echo request (DF) (ttl 55, id 0)
20:15:45.423077 216.52.65.75 > X.X.X.29: icmp: echo request (DF) (ttl 50, id 0)
20:17:15.990226 216.52.129.75 > X.X.X.24: icmp: echo request (DF) (ttl 50, id 0)
20:17:44.494449 216.52.129.68 > X.X.X.24: icmp: echo request (DF) (ttl 56, id 0)
20:18:48.556897 216.52.65.70 > X.X.X.29: icmp: echo request (DF) (ttl 54, id 0)
20:19:13.391353 216.52.129.73 > X.X.X.30: icmp: echo request (DF) (ttl 53, id 0)
20:19:53.089049 216.52.190.72 > X.X.X.29: icmp: echo request (DF) (ttl 54, id 0)
20:20:48.146439 216.52.193.73 > X.X.X.29: icmp: echo request (DF) (ttl 54, id 0)
20:23:28.907176 216.52.94.67 > X.X.X.29: icmp: echo request (DF) (ttl 55, id 0)
20:24:31.401430 216.52.94.65 > X.X.X.29: icmp: echo request (DF) (ttl 55, id 0)
20:25:29.411513 216.52.129.70 > X.X.X.10: icmp: echo request (DF) (ttl 53, id 0)
20:25:50.900512 216.52.129.67 > X.X.X.30: icmp: echo request (DF) (ttl 54, id 0)
20:27:37.224156 216.52.169.67 > X.X.X.29: icmp: echo request (DF) (ttl 51, id 0)
20:28:21.068738 216.52.169.75 > X.X.X.29: icmp: echo request (DF) (ttl 54, id 0)
20:28:39.275236 216.52.193.72 > X.X.X.29: icmp: echo request (DF) (ttl 54, id 0)
```

3. Probability the source address was spoofed:

No. The source is looking for a response to the request.

Whois shows the owner of the IP 216.52.0.0 block as:

```
InterNAP Network Services (NETBLK-PNAP-8-98)
   2001 Sixth Avenue Suite 800
   Seattle, WA 98121   US
   Netname: PNAP-8-98
   Netblock: 216.52.0.0 - 216.52.255.255
   Maintainer: PNAP
   Coordinator:
      Operations Center, InterNAP Network  (INO3-ARIN)  noc@INTERNAP.COM
      206.256.9500 (FAX) 206.256.9580
   Domain System inverse mapping provided by:
   NS1.PNAP.NET                206.253.194.65
   NS2.PNAP.NET                206.253.194.97
   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
   Record last updated on 16-May-1999.
   Database last updated on  16-Feb-2002 19:55:51 EDT.
```

38 different hosts sent a ping request to 6 live hosts on our DMZ; on the surface, that would look like the IP's were real and not spoofed, but it bothers me that the id of 0 from all the different hosts with a very similar TTL. That sends a red flag indicating a crafted set of packets. A little further investigation with NSLOOKUP shows all the hosts have a similar naming scheme:

performance-70.atl.pnap.net [216.52.193.70]
performance-72.den.pnap.net [216.52.41.72]
performance-75.chi.pnap.net [216.52.129.75]
performance-72.hou.pnap.net [216.52.169.72]
performance-70.nym.pnap.net [216.52.94.70]
performance-68.phi.pnap.net [216.52.65.68]
performance-72.dal.pnap.net [216.52.190.72]
performance-73.atl.pnap.net [216.52.193.73] etc….

Odd. Kind of looks like devices in different cities. After looking at their web site: http://www.internap.net I saw the following posting:

> "InterNap ™ has built an Overlay Network that continually analyzes the traffic situtation on every major Internet backbone, then selects the path of least resistance to deliver mission critical information and communication faster and more reliably."

Now, I'd say the hosts are real and not spoofed.

4. Description of attack:
Initially it looks to be some kind of weird reconnaissance ping to see who and what is alive. All echo requests are coming from the same IP block using multiple hosts to target 6 of our DMZ web servers. A classical signature of crafted packets is an id is 0; this shows on every packet from all the different hosts. Closer inspection of the packets using the hex output of windump shows the following: (I only show two, as they all look the same except for the source IP).

{This trace has been sanitized to remove company information}
20:06:23.899533 216.52.193.70 > X.X.X.29: icmp: echo request (DF) (ttl 52, id 0)

4500 0054 0000 4000 3401 f295 d834 c146 **bold = IP header byte 0 - 20**
xxxx xxxx *0800 1a05 0589 93b4 0fd3 613c* *italic = icmp header and data*
db82 0d00 0000 0000 0000 0000 0000 0000
0200 0000 xxxx xxxx 0000 0000 0000 0000
80d5 ffbf 6b90 8c40 1d00 0000 b8d5 ffbf
e0d5 ffbf

20:06:27.018447 216.52.193.70 > X.X.X.29: icmp: echo request (DF) (ttl 52, id 0)

4500 0054 0000 4000 3401 f295 d834 c146
xxxx xxxx *0800 7147 0589 2be4 13d3 613c*
f510 0000 0000 0000 0000 0000 0000 0000
0200 0000 xxxx xxxx 0000 0000 0000 0000
80d5 ffbf 6b90 8c40 1d00 0000 b8d5 ffbf
e0d5 ffbf

The IP header looks somewhat standard. Byte 0 is normally represented in hex as 45 where 4 indicates standard IP v4 and 5 is indicates a standard header length of 20 bytes unless options are present. The IPLength is 20 bytes. TOS is byte 1= 00 which is standard. Byte 2 and 3 represent DgmLength and are hex 0054 = 84 bytes total (packet). Byte 4 and 5 represent IP Id = 0. It is unusual to have every IP id 0; they general increase as each packet is sent from the source. Byte 6 and Byte 7 represent flags and fragment offsets. In this case the DF (don't fragment) flag is set. Byte 8 is the time to live (hex 34, decimal=52) In the IP header, Byte 9 of the packet should indicate the protocol (TCP is hex 06, ICMP is hex 01 and UDP is hex 11 (ref 1).

The total datagram length is 84 bytes of which IP header is 20 bytes and intial ICMP header is 4 bytes. Therefore, the rest of the ICMP packet (84-20-4 = 60 bytes) is carrying additional data to the destination IP which is unusual. Generally, Echo request is a ping to a device and has no payload; the echo reply can carry payload which may be the original Echo request and any router hop information sent back to the originator of the request. These packets look like echo replies in that they carry additional echo information about our device, but the IP header information indicates that the packet is an *echo request not an echo reply*. Bizarre.

5. Attack mechanism:
All the packets from the InterNap hosts seem to be unusual for ICMP Echo requests. They appear to carry a data which is not normally seen with standard echo requests. It looks as though InterNap is using ICMP requests to build their overlay network as mentioned on their web site. It may be their ICMP requests map out routes to Corporations and their servers.

1. Northcutt, Stephen and Novak, Judy, "Network Intrusion Detection, An Analyst's Handbook, Second Edition, New Riders, 2001, pp 8 and pp35 -43.
2. Stevens, Richard, "TCP/IP Illustrated, Volume 1., Addison-Wesley, 1994 p34-35.

Correlations:
I did a quick search on Google with the keywords ICMP weird traffic. Johannes B. Ulrich posted to Lists.jammed.com a response to a request for help on strange ICMP traffic. The source Ips

were different and the hex was slightly different but had a few similarities. The ip version in hex was bogus; the TOS in hex was bogus and a good chunk of the ip header was also padded with zeros. The ICMP info contained the same IP header length and datagram length indicating there was additional information in the ICMP payload beyond the echo request. Etc. See posting.

- Ulrich, Johannes B., "ICMP help" on Lists.Jammed.com, URL: http://lists.jammed.com/incidents/2001/06/0247.html June 29,2001.

7. Evidence of active targeting:
Yes, definitely targeted to our servers on our DMZ segment. Repeatedly sent to the same servers.

8. Severity:
(Criticality + Lethality) - (System Countermeasures + Net Countermeasures) = Severity

Criticality = 4	Web servers on our DMZ**.**
Lethality= 1	Echo requests blocked at our firewall.
System Countermeasures = 5	Servers patched and monitored.
Net Countermeasures = 5	Firewall is restrictive
Severity= -5	Not severe, but incredably annonying.

9. Defensive recommendation
Continues to not allow echo requests to or host unreachables from our DMZ. Although this trace was found on the outside of our firewall, I would review the packet activity on the dmz to be sure that other echo requests were not coming in.

10. Multiple choice test question:
What in the IP packet header indicates ICMP traffic
a. Byte 0 = hex value 01
b. Byte 1= hex value 06
c. Byte 20 = hex value 11
d. Byte 9= hex value 01

Answer d)

Detect#4 DTSPCD activity, port 6112

(Back to top)

1. Source of Trace.
From incidents.org

2. Detect was generated by:
By snort although the author does not indicate it.

Date: Mon, 11 Feb 2002 17:21:05 -0600 *From*: Ernie Pritchard *Subject*: 6112 scans
http://www.incidents.org/archives/intrusions/msg03765.html
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.132:6112 SYN ******S*

```
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.133:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.134:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.136:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.137:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.138:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.139:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.151:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.152:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.153:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.154:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.155:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.156:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.158:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.159:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.160:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.162:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.191:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.193:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.194:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.200:6112 SYN ******S*
Feb 11 01:19:59 204.192.116.243:6112 -> xxx.xxx.xxx.201:6112 SYN ******S*
Feb 11 01:20:00 204.192.116.243:6112 -> xxx.xxx.xxx.64:6112 SYN ******S*
Feb 11 01:20:00 204.192.116.243:6112 -> xxx.xxx.xxx.65:6112 SYN ******S*
Feb 11 01:20:00 204.192.116.243:6112 -> xxx.xxx.xxx.66:6112 SYN ******S*
Feb 11 01:20:00 204.192.116.243:6112 -> xxx.xxx.xxx.67:6112 SYN ******S*

Feb 10 22:35:59 216.38.192.53:6112 -> xxx.xxx.xxx.131:6112 SYN ******S*
Feb 10 22:35:59 216.38.192.53:6112 -> xxx.xxx.xxx.132:6112 SYN ******S*
Feb 10 22:35:59 216.38.192.53:6112 -> xxx.xxx.xxx.134:6112 SYN ******S*
Feb 10 22:35:59 216.38.192.53:6112 -> xxx.xxx.xxx.130:6112 SYN ******S*
Feb 10 22:35:59 216.38.192.53:6112 -> xxx.xxx.xxx.143:6112 SYN ******S*
Feb 10 22:35:59 216.38.192.53:6112 -> xxx.xxx.xxx.153:6112 SYN ******S*
Feb 10 22:35:59 216.38.192.53:6112 -> xxx.xxx.xxx.152:6112 SYN ******S*
```

3. Probability the source address was spoofed:
No. Based on only the above information, the hostile is sending crafted SYN packets and more then likely looking for a response to see if the hosts is alive.

4. Description of attack:
Looks to be a standard port scan using SYN packets to TCP port 6112. The trace is extremely fast. The packets do appear to be crafted as the source and destination ports are exactly the same. Generally, a SYN scan is a reconnaissance effort to map out an IP range to determine live hosts and whether they are responsive to a particular port. It gives the hostile information that can be used to focus on live hosts for futher OS fingerprinting and exploitation.

5. Attack mechanism:
Iana port numbers indicate 6112/tcp and 6112/udp belongs to dtspcd service.
From the Iana web site, URL: http://www.iana.org/assignments/port-numbers . It appears as though port scans on 6112 may correlate to possible reconnaissance attempts in finding not only live hosts, but live hosts running the CDE package.

6. Correlations:
Since the buffer overflow for CDE affects every unix system, there are a lot of references to this simple search on Google yields several. Several of the vendors provide good details on what the vulnerability is which indicates that a port scan of this type is a serious attempt to find a server that is vulnerable and can be compromised, giving the attacker root access.

- Internet Security Systems, "Multi-vendor CDE dtspcd daemon buffer overflow", URL http://www.iss.net/security_center/static/7396.php

Symantec web site is a good resource not only in describing the problem but also in giving details on checking your system.

> "Symantec Corporation advises its customers to be aware of a remote root-access buffer overflow vulnerability in the Common Desktop Environment's (CDE's) desktop subprocess control service(dtspc). A remote intruder can cause arbitrary code to be run with root-level privileges on the targeted system, potentially gaining root access to the system.
>
> The CDE is an integrated graphical user interface that runs on Unix and Linux operating systems. "dtspcd" is a network daemon that accepts requests from clients to execute commands and launch applications remotely. On systems running CDE, dtspcd is spawned by the Internet services daemon (typically inetd or xinetd) in response to a CDE client request. The dtspcd is typically configured to run on port 6112/tcp with root privileges. dtspcd makes a function call to a shared library that contains a buffer overflow condition in the client connection routine…..
>
> … To determine if you have CDE installed and enabled, check for the following entries.
>
> In /etc/services check for "dtspc 6112/tcp"
> In /etc/inetd.conf check for "dtspc stream tcp nowait root /usr/dt/bin/dtspcd /usr/dt/bin/dtspcd "

Mitre's site lists many of the affected systems. The list is quite lengthy.

- Mitre.org, URL http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0803 , (under review).
- CERT web site, Vulnerability Note VU#172583, "Common Desktop Environment (CDE) Subprocess Control Service dtspcd contains buffer overflow", URL http://www.kb.cert.org/vuls/id/172583#systems , 11/12/2001.
- CERT Advisory CA-2001-31 Buffer Overflow in CDE Subprocess Control Service, URL http://www.cert.org/advisories/CA-2001-31.html Original release date: November 12, 2001.
- CERT Advisory CA-2002-01 Exploitation of Vulnerability in CDE Subprocess Control Service", URL http://www.cert.org/advisories/CA-2002-01.html , January 14th, 2002.

7. Evidence of active targeting:

This is a general scan of the hosts on this network. Although it is not targeted at a specific host, all the attacker needs to do is to find a host that responds. Of course, after that the active targeting begins.

8. Severity:

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

Criticality = 3	Scanning for live hosts answering to port 6112.
Lethality= 5	If the scan provided for real unix servers and they had the CDE process, there would be reason for concern**.**
System Countermeasures = 3.	Not really any info on this although network IDS did catch the scan. Assuming the server or servers in question are patched since this has been out since late 2001**.** No indication of any host IDS or forensics packages that could

	detect this attack against the servers.
Net Countermeasures = 1	Not really any info on this. Doesn't appear to blocked at firewall although the author of the trace doesn't indicate one way or another

Severity= 4.

9. Defensive recommendation

Immediately, check any unix servers running CDE that were not patched for the vulnerability using the information from Also, as recommended by SecurityFocus, disable the dtspcd service in /etc/inetd.conf if not being used.

10. Multiple choice test question:

What would a SYN packet to multiple hosts from the same source possibly indicate ?

A) an attack against live a host
B) a tcp connection established to live hosts
C) a recon for live hosts
D) a simple dns name query for the name servers

Answer C)

Detect#5 NETBIOS activity, port 137

(Back to top)

1. Source of Trace.

My employer's network.

2. Detect was generated by:

Snoop was originally used on the outside interface of the firewall to capture. Packets are displayed using snort against the capture file. Only part of the trace is shown below.

```
02/06-20:25:50.163424 80.132.77.48:137 -> X.X.X.23:137
UDP TTL:120 TOS:0x0 ID:379 IpLen:20 DgmLen:78
Len: 58
B5 FF 00 00 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAA..!
00 01                                            ..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/06-20:25:51.667160 80.132.77.48:137 -> X.X.X.23:137
UDP TTL:120 TOS:0x0 ID:514 IpLen:20 DgmLen:78
Len: 58
B5 01 00 00 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAA..!
00 01                                            ..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/06-20:26:02.189835 80.132.77.48:137 -> X.X.X.23:137
UDP TTL:120 TOS:0x0 ID:1611 IpLen:20 DgmLen:78
Len: 58
```

```
B5 0F 00 00 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..
```

=+

Interestingly, if the same trace is brought into ethereal, it looks like this:

```
20:25:50.163424 80.132.77.48.137 > X.X.X.23.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; UNICAST
20:25:51.667160 80.132.77.48.137 > X.X.X.23.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; UNICAST
20:26:02.189835 80.132.77.48.137 > X.X.X.23.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; UNICAST
20:26:03.682748 80.132.77.48.137 > X.X.X.23.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; UNICAST
20:26:05.189221 80.132.77.48.137 > X.X.X.23.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; UNICAST
20:26:06.688682 80.132.77.48.137 > X.X.X.23.137:
>>> NBT UDP PACKET(137): QUERY; REQUEST; UNICAST
```

3. Probability the source address was spoofed:
No. The hostile is looking for a NetBios response from our windows server in which to then try to exploit.

4. Description of attack:
It appears to be a NetBios port 137 scan against our Windows server. The trace shows the characteristics CKAAAAAAA… probe discussed by Hobbit and Bryce Alexander. Hobbit explained the mangling of the NetBios names into the packets. Interestingly, the article mentions how the wildcard or * character when used in a name query will force the listener to reply with with all of its resource records. The 16th bit character in a Windows Netbios name represents a service that is running and available on the machine. This would give the attacker additional information on which to act.

5. Attack mechanism:
The attack is based on forcing a Windows machine to response to the wildcard status request. Both Bryce Alexander and Hobbit showed the decodes for the packet which are repeated here. The reader is encouraged to review that documentation for decode analsis.

6. Correlations:
The trace detected above bears similar characteristics to Bryce Alexander posted trace and information.

- Alexander, Bryce, Intrusion Detection FAQ "Port 137 Scan" URL http://www.sans.org/newlook/resources/IDFAQ/port_137.htm ,May 10, 2000.

He indicated the internet worm called network.vbs was active in looking for NetBios host that would respond to NetBios requests.As mentioned in Alexander's information the network.vbs worm trace posted at Global Incident Analysis Center, Special Notice, "Followup on a HoneyPot Catch", URL http://www.sans.org/y2k/honeypot_catch.htm looks similar to the above.

Information on NetBios DOS can be located on the Cert site at

- CERT Vulnerability Note VU#32650, "Denial of Service Attack in NetBIOS Services", URL Http://www.kb.cert.org/vuls/id/32650
- CERT Incident Note IN-2000-02, "Exploitation of Unprotected Windows Networking Shares", URL http://www.cert.org/incident_notes/IN-2000-02.html

An excellent description of NetBios name queries was done by Hobbit in 1997and is posted at

- Hobbit, "CIFS: Common Insecurities Fail Scrutiny", URL http://packetstormsecurity.nl/UNIX/netcat/cifs_common_insecurities_fail_scrutiny.txt , 1997.

7. Evidence of active targeting:
Yes, the attacker has located a target and is focused on any responses that can be returned.

8. Severity:
(Criticality + Lethality) - (System Countermeasures + Net Countermeasures) = Severity

Criticality = 5	It's an important web server**.**
Lethality= 5	If the exploit worked, our server would be compromised**.**
System Countermeasures = 5.	The server is fully patched**.**
Net Countermeasures = 2.	Although the ruleset on the firewall is restrictive, it is configured to allow port 80 traffic. It can't distinguish between "good" port 80 and "bad" port 80. It's just allows port 80 web traffic.

Severity= 2.

9. Defensive recommendation
Prevent NetBios responses from entering or exiting the firewall. Verify servers have enabled, up-to-date and running anti-virus software. Monitor servers with a host based intrusion detection product or file integrity checker like Tripwire.

10. Multiple choice test question:
What is a port used for NetBIOS?
a) 21
b) 80
c) 137
d) 111

Answer c)

Assignment #3 Analyze This

Introduction - 5 days of IDS logs

The following analysis done from the IDS logs provided by the University of Maryland, Baltimore County. Logs were obtained from Http://www.research.umbc.edu/~andy. As given in Christof Voemel's GCIA paper (http://www.giac.org/practical/Christof_Voemel_GCIA.txt) information on the sensor's and their use can be determined from http://userpages.umbc.edu/~robin/Presentations/Snort/Snort_Final.ppt. It appears that two Intel systems with OpenBsd are running Snort and collecting information on a network of about 10,000 users as stated. With only two devices are in use, it can only be assumed that information being collected at the perimeter of the network, but this was not confirmed. Three types of IDS files: scans, out of spec, and alerts were provided. The alerts provide information about portscans and exploits attempted on the network. Out of spec files provide information on unusual packets which do not fit the norm. Scans provide information on scanned hosts and/or networks segments which may be concerted efforts at reconnaissance information that later may be used to direct attacks at specific targets.

List of files analyzed

Files were taken from http://www.research.umbc.edu/~andy for the 5 days posted .

ALERTS	OOS (Out of Spec)	SCANS
Alert.020125	OOS_Jan.25.2002	Scans.020125
Alert.020126	OOS_Jan.26.2002	Scans.020126
Alert.020127	OOS_Jan.27.2002	Scans.020127
Alert.020128	OOS_Jan.28.2002	Scans.020128
Alert.020129	OOS_Jan.29.2002	Scans.020129

These files were concated together to produce three files to analyze: alerts_all, OOS_all and Scans_all.

Executive Summary of Analysis (Back to top)

- Based on the following analysis given in subsequent pages, there is a great deal of activity on the University's MY.NET segments. 700,586 alerts and 1,693,417 scans were triggered. Some MY.NET host may require further investigation. Scanning activity is enormous. Several internal hosts appear to be generating this traffic. There definitely is a correlation between scanning activity, alerts and OOS. Following highlights the results of the analysis.
- MY.NET.88.163 was the top receiver of scans. It should be monitored as exploits may be attempted against it.
- MY.NET.60.43 was the top scanner host. It sent 449,300 scan packets out of a total of 1,693,417 scan packets in a five day period. This host should be immediately removed from network and examined.
- MY.NET.1.4, MY.NET.153.194, and MY.NET.153.173 should be also checked as they were involved in scans.
- MY.NET.88.162 and MY.NET.5.96 received scans, alerts and OOS packets which indicated OS fingerprinting attempts. These hosts may be targets for future exploits and should be monitored.

- MY.NET.153.159 may be compromised and is sending numerous CGI null byte attempts to an outside destination. This host should be investigated.
- Several Yahoo hosts are send large UDP packets. This activity should be monitored.
- MY.NET.88.103 may be compromised and is sending numerous Adore (Red Worm) attempts to an outside destination. This host should be investigated.
- Large numbers of SNMP traffic were detected. At this time, it remains internal.
- SMB NetBIOS traffic was detected and appears to be internal.
- L3 retriever scans/mapping was detected. This is assumed authorized by the university.
- Chat was detected. This is assumed authorized by the university.

Alert Analysis: (Back to top)

In the five day period, there were 700,586 alerts detected and recorded; portscan alerts accounted a very large percentage (70.8 %) of that and will be discussed in the scan analysis. Of the remaining 204,267 alerts, the following are listed as the top ten during that time period. As observed, roughly the top four occurrences represent over 75% of the total non portscan alerts. Table 1 shows the top ten alerts triggered during the five day period.

Table 1. Top Ten Alert types

Detect	Occurences (%)	Description
55269	27.06	Connect to 515 from inside
53179	26.03	Spp_http_decode: IIS Unicode attack detected
24042	11.77	SNMP public access
23908	11.70	MISC Large UDP Packet
17454	8.54	SMB Name Wildcard
8599	4.21	ICMP Echo Request L3retriever Ping
5445	2.67	INFO MSN IM Chat data
3995	1.96	High port 65535 udp - possible Red Worm - traffic
2502	1.22	Spp_http_decode: CGI Null Byte attack detected
1807	0.88	Watchlist 000220 IL-ISDNNET-990517

For each alert an example is shown with a brief description, statistics and recommendations.

Connect to 515 from inside

Analysis:

01/25-07:58:18.133939 ** connect to 515 from inside ** MY.NET.153.118:1188 -> MY.NET.150.198:515
01/25-07:58:18.134004 ** connect to 515 from inside ** MY.NET.153.118:1188 -> MY.NET.150.198:515
01/25-07:58:18.134213 ** connect to 515 from inside ** MY.NET.153.118:1188 -> MY.NET.150.198:515

This is the alert tops the list. Normally this activity may be an attempt by the source IP to look for unpatched Unix server with the LPD/LPR package. In unpatched systems the LPD service is exploitable due to a buffer overflow vulnerability. The service can crash or run code at elevated privilege. The vulnerability may lead to root level access. Two sources of information from the SANS web site provide additional details.

- SANS.ORG, "Alerts: Increased Probes to TCP 515", URL http://www.sans.org/newlook/alerts/port515.htm , November 20,2000.
- SANS.ORG, "The Twenty Most Critical Internet Security Vulnerabilities", URL

http://www.sans.org/top20.htm#_Toc526136821 January 30, 2002.

Based on the alerts, I believe this may be a false positive because *all 103 hosts* are from MY.NET and are sending this to *only MY.NET.150.198* which leads me to believe that MY.NET.150.198 must be a print server for these hosts. There are no external sources triggering this alert.

Recommendations:
No action appears to be currently needed but for future record, any MY.NET hosts running the LPD/LPR package should always be patched against possible exploits that may trigger this alert.

IIS Unicode Attack detected
Analysis:
01/25-05:21:38.357697 ** spp_http_decode: IIS Unicode attack detected ** 203.227.74.100:53565 -> MY.NET.5.141:80
01/25-05:21:39.256386 ** spp_http_decode: IIS Unicode attack detected ** 203.227.74.100:10242 -> MY.NET.5.141:80
01/25-05:21:39.759422 ** spp_http_decode: IIS Unicode attack detected ** 203.227.74.100:57819 -> MY.NET.5.141:80

This was discussed in detail during the analysis of Network Trace #1, which described a Nimda attack that uses this as part of its mechanism. No further analysis will be done except to note that this attack is listed in the SANS web site as one of the top twenty Internet vulnerabilities.

- SANS.ORG, "The Twenty Most Critical Internet Security Vulnerabilities", URL http://www.sans.org/top20.htm#_Toc526136821 January 30, 2002.

Based on the alerts, 10 external hosts triggered this alert to MY.NET. hosts (MY.NET.150.83, MY.NET.5.96, MY.NET.5.92, MY.NET.5.141, MY.NET.5.95). There is a tremendous amount of outbound alerts from MY.NET hosts to external hosts and will not be shown here except to mention most of it is from MY.NET.153 segment and the top outbound talker is MY.NET.153.123.

Recommendations:
This is alerts to attempts to compromise unpatched Microsoft IIS servers. Any MY.NET hosts running IIS, which are unpatched for this vulnerability and are receiving these packets should be immediately checked for compromise and removed from the network. If the MY.NET hosts are patched this vulnerability fails. Microsoft IIS has been known for several spectacular exploits. For the peace of mind of the University security administrator, it is recommended that any IIS server on site have either a real-time host intrusion detection product or file integrity product like Tripwire running on it. Patch maintenance is critical on these servers.

SNMP public Access
Analysis:
01/25-00:00:09.407176 ** SNMP public access ** MY.NET.153.220:1242 -> MY.NET.152.109:161
01/25-00:00:15.406092 ** SNMP public access ** MY.NET.153.220:1242 -> MY.NET.152.109:161
01/25-00:00:21.407144 ** SNMP public access ** MY.NET.153.220:1242 -> MY.NET.152.109:161

SNMP (a.k.a., Simple Network Management Protocol) is used to manage and monitor servers, printers, routers and switches. It is listed as one of SAN's top twenty security vulnerabilities because many devices come enabled with SNMP using a community string of public or blank. In

some cases, it is configured with easily guessable strings like "private" or the name of the corporation. Because SNMP is so widely used in so many network devices, attackers who can gain access to devices via SNMP can glean sensitive network information, create DOS (denials of service), or do a remote shutdown. Recently, CERT issued an advisory about SNMPv1 (version 1) to that effect adding to it the fact that some SNMP vulnerabilities could occur even without using of the real community string.

- CERT Advisory CA-2002-03 Multiple Vulnerabilities in Many Implementation of the Simple Network Management Protocol, URL http://www.cert.org/advisories/CA-2002-03.html , Original release date: February 12, 2002
- SANS.ORG, "The Twenty Most Critical Internet Security Vulnerabilities", URL http://www.sans.org/top20.htm#_Toc526136821 January 30, 2002.

Based on the alerts, 17 hosts triggered alerts to 143 hosts. A breakdown of the SNMP traffic is shown. 50% of the traffic is coming from MY.NET.70.177 and 30% is being sent to MY.NET.152.109. Either way, it appears as though all SNMP traffic is internal.

Sources of SNMP alerts

Occurrences	%	Source
11908	50	MY.NET.70.177
3863	16	MY.NET.150.198
2459	10	MY.NET.150.41
1994	8	MY.NET.150.245
1940	8	MY.NET.153.220
1276	5	MY.NET.88.240
254	1	MY.NET.84.155
180	1	MY.NET.150.49

.

Recommendations:

Although all traffic is currently internal, unauthorized attempts to access network devices via SNMP should be taken seriously, especially with the current advisory that has been issued. Any SNMP traffic from internal MY.NET hosts not authorized for that activity should be noted and prohibited. Any MY.NET hosts and devices running SNMP should be immediately patched assuming the vendor of the device has issued one.

MISC Large UDP Packet

Analysis:

```
01/28-12:54:37.527114  ** MISC Large UDP Packet ** 63.250.209.34:0 -> MY.NET.153.210:0
01/28-12:54:38.626261  ** MISC Large UDP Packet ** 63.250.209.34:4964 -> MY.NET.153.210:1353
01/28-12:54:38.728609  ** MISC Large UDP Packet ** 63.250.209.34:4964 -> MY.NET.153.210:1353
01/28-12:54:38.825648  ** MISC Large UDP Packet ** 63.250.209.34:4964 -> MY.NET.153.210:1353
01/28-12:54:39.433764  ** MISC Large UDP Packet ** 63.250.209.34:4964 -> MY.NET.153.210:1353
01/28-12:54:39.934361  ** MISC Large UDP Packet ** 63.250.209.34:0 -> MY.NET.153.210:0
01/28-12:54:40.229859  ** MISC Large UDP Packet ** 63.250.209.34:4964 -> MY.NET.153.210:1353
01/28-12:54:40.637131  ** MISC Large UDP Packet ** 63.250.209.34:4964 -> MY.NET.153.210:1353
01/28-12:54:43.135717  ** MISC Large UDP Packet ** 63.250.209.34:4964 -> MY.NET.153.210:1353
01/28-12:54:45.135904  ** MISC Large UDP Packet ** 63.250.209.34:0 -> MY.NET.153.210:0
```

The above trace shows that this type of packet is generally crafted. Notice the port is 0 on both

source and destination or it is always the same number on both source and destination. Generally clients use ephemeral ports which are 1024 and above; they generally increase as the client sends packets out. The three top sources as show above have IP address space registered with Yahoo as determined from Arin's WHOIS database. These three sources triggered alerts directed to MY.NET.151.63, MY.NET.153.210, and MY.NET153.193. It would have been very useful to have the tcpdump information to look at the hex from these sources to find out what kind of mischieve they were up to.The alert was triggered by the following sources**:**

Occurrences	Source IP
10740	63.250.209.34
7246	63.250.210.50
4970	63.250.208.34
677	64.152.216.77
199	211,233,27,142
47	211.233.27.142
27	63.250.211.197

The alert was triggered to the following destinations**:**

Occurrences	Destination IP
22737	MY.NET.151.63
677	MY.NET.153.194
266	MY.NET.153.210
199	MY.NET.153.160
27	MY.NET.153.193

Interestingly, MY.NET.151.63 received tremendous numbers of scan packets as discussed later. The other four top destinations sent large numbers of SYN scan packets; this was determined from the scan data. I don't know if there's any correlation without more information, but it seems odd that together these same hosts are all top targets here.

One correlation about similar activity was made by Gregory Lajon in his practical on MISC Large UDP packets. He mentions an article on Network Gaming. Could MY.NET.151.63 now be involved in heavy gaming. Doesn't really fit the pattern here, based on the ports listed in his information, but it is an interesting idea. No other good correlations fit this activity to data, except that is unusual and out of the norm.

- Scarborough, Matt, "What are some signs of Internet Gaming", http://www.incidents.org/detect/gaming.php , May 5th, 2001.

Recommendations:
It is advisable to at least closely check MY.NET.151.63 for any signs of compromise. Frankly, I'd check the other four MY.NET hosts as well, simply because of all the activity to them.

SMB Name Wildcard
Analysis:

```
01/29-22:33:12.995537  ** SMB Name Wildcard ** 169.254.22.29:137 -> MY.NET.153.166:137
01/29-22:33:14.497294  ** SMB Name Wildcard ** 169.254.22.29:137 -> MY.NET.153.166:137
01/29-22:33:15.996996  ** SMB Name Wildcard ** 169.254.22.29:137 -> MY.NET.153.166:137
01/29-23:22:03.559727  ** SMB Name Wildcard ** 169.254.22.29:137 -> MY.NET.151.124:137
01/29-23:22:05.057256  ** SMB Name Wildcard ** 169.254.22.29:137 -> MY.NET.151.124:137
01/29-23:22:06.556463  ** SMB Name Wildcard ** 169.254.22.29:137 -> MY.NET.151.124:137
```

- SANS.ORG, "The Twenty Most Critical Internet Security Vulnerabilities", URL http://www.sans.org/top20.htm#_Toc526136821 January 30, 2002.

This alert is occurs when the source IP is attempting a NetBIOS query from the destination address. Windows machines share information via SMB's (Server Message Blocks) and NetBIOS queries allow users to location shared files/directories, but an authorized query can provide a hostile party with sensitive user, group, and password information. Misconfigure NetBIOS activity has been listed as one of the SAN's top twenty Internet vulnerabilities to be aware of.

Based on the alerts, 172 hosts attempted this attack 17454 times to 233 MY.NET HOSTs. 6961 attempts were directed to MY.NET.11.7 and MY.NET.11.6. These attempts generally appear as Microsoft traffic, which could be considered Network browsing traffic between Microsoft machines. Without knowing the layout of the University network it is assumed that the top destination MY.NET hosts (MY.NET.11.7 and MY.NET.11.6) are either domain controllers, master or backup browsers. These types of machines would by default normally receive NetBIOS queries for various services and network browsing.

Recommendations:
NetBIOS activity is standard on Microsoft networks and should be expected. However, external attempts to internal hosts should be viewed as suspicious. In addition, the potential for abuse by internal users to try and elevate their privileges can always exist. Properly configured servers that are patched via Microsoft security directives will help, but some mechanism for auditing of the server's security logs would be a good idea.

Additionally, it is without question that no NetBIOS activity should be allowed across the firewall external sources (that is ports 137- 139).

ICMP Echo Request L3 Retriever Ping
Analysis:

```
01/25-01:58:19.764080  ** ICMP Echo Request L3retriever Ping ** MY.NET.150.127 -> MY.NET.5.4
01/25-01:58:22.013620  ** ICMP Echo Request L3retriever Ping ** MY.NET.150.127 -> MY.NET.5.4
```

This alert is based on a product that is used to map and test a network for vulnerabilities. The Retriever product can send out pings or SNMP queries to gather information. It appears that a MY.NET.150.127 host may have this product installed and is being used for network reconnaissance. It also does vulnerability testing of network devices.

Security Magazine web site, "L3 Network Security Retriever", http://www.scmagazine.com/scmagazine/standalone/l3/l3_retriever.htm , No date posted.

Symantec web site, "L-3 Network Security Unveils Expanded Network Security Management Tool", URL http://www.symantec.com/press/security/n990923_ns.html , No date posted.

Recommendations:
If MY.NET.150.127 is a host being used by the University security administrator to map the network then this alert is not important. However, if the IT security group is not using this product then investigation of who is using it on MY.NET.150.127 host should be done.

INFO MSN IM Chat data
Analysis:
01/25-09:37:34.934374 ** INFO MSN IM Chat data ** 64.4.12.193:1863 -> MY.NET.153.106:2673
01/25-09:37:47.355233 ** INFO MSN IM Chat data ** MY.NET.153.106:2673 -> 64.4.12.193:1863
01/25-09:37:50.788196 ** INFO MSN IM Chat data ** MY.NET.153.106:2673 -> 64.4.12.193:1863

This alert is an example of Microsoft Instant Messenger.
Based on the alerts, there were 32 MY.NET hosts engaged in MSN traffic.

Recommendations:
It depends on the policy of the University as to how this traffic should be viewed. There currently are no known vulnerabilities for MSN although recently vulnerabilities for AOL Instant Messenger have been posted.

High port 65535 udp- possible RedWorm Traffic
01/25-08:40:25.863793 ** High port 65535 udp - possible Red Worm - traffic ** MY.NET.6.48:65535 -> MY.NET.153.189:65535
01/25-08:40:26.174239 ** High port 65535 udp - possible Red Worm - traffic ** MY.NET.6.48:65280 -> MY.NET.153.189:65535
01/25-08:40:26.177270 ** High port 65535 udp - possible Red Worm - traffic ** MY.NET.6.48:65280 -> MY.NET.153.189:65535

This alert is based on possible traffic from a worm called Red Worm, also known as Adore. The worm seeks unpatched Linux hosts vulnerable to any of several exploits against LPRng, rpc-statd, wu-ftp or BIND. It replaces ps which is a Unix command to show running processes and sends sensitive information like /etc/shadow out to several external emails. There is a good posting on the SANS web site that give all the pertinent details and variants. It also contains links to the various vulnerabilities that the worm is attempting to exploit.

SANS.ORG, Global Incident Analysis Center, "Adore Worm - Version 0.8", URL http://www.sans.org/y2k/adore.htm , April 12th ,200l.

Based on the alerts, 32 external hosts triggering this alert to MY.NET HOSTs. The two major external hosts were 64.152.108.141 (278 times) and 64.152.108.142 (398 times). They would appear to be compromised themselves. The owners of these IP's are

```
Streaming Media Corporation (NETBLK-NETBLK-STRM9)
   6446 South Kenton Street, Suite 130
   Englewood, CO 80111   US
   Netname: NETBLK-STRM9
   Netblock: 64.152.108.0 - 64.152.108.255
   Coordinator:
      Hostmaster, SMC  (ZH58-ARIN)  hostmaster@smc.net
      720-875-0700
```

```
Record last updated on 20-Jun-2001.
Database last updated on  4-Mar-2002 19:58:28 EDT.
```

These two sources both triggered alerts to MY.NET.88.163. The other four top MY.NET destination targets were MY.NET.153.194, MY.NET.153.189, MY.NET.153.157, and MY.NET.153.210.

Recommendations:
Any MY.NET hosts running unpatched Linux operating systems, which have received this connection, should immediately be checked for compromise. In particular, it looks as though MY.NET.88.163 was the intended target for a large percentage of the traffic. This host should be checked for compromised if it has not been patched. The Global Incident Analysis site listed above has a link for a utility called Adorefind which will detect and remove the files the worm places on a server. If compromise has occurred, the server should be removed from the network until verified it is clean.

SPP_HTTP_DECODE: CGI Null Byte Attack
Analysis:

```
01/25-09:36:19.661827  ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.171:3493 -> 216.33.88.53:80
01/25-09:36:19.661827  ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.171:3493 -> 216.33.88.53:80
01/25-09:36:20.521019  ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.171:3494 -> 216.33.88.53:80
```

This alert is triggered via an null byte exploit against CGI scripts running on a web server. The exploit may allow an attacker to read or modify files. Since many web servers use CGI scripts in their web pages, this exploit can be serious. Rain Forest Puppy posted information regarding this.

- Rain Forest Puppy, "InternetWeek: New Web Attacks (Poison Null Byte)", URL http://www.wiretrip.net/rfp/p/doc.asp/i2/d37.htm , February 7th, 2001
- SANS.ORG, "The Twenty Most Critical Internet Security Vulnerabilities", URL http://www.sans.org/top20.htm#_Toc526136821 January 30, 2002.

SANS.ORG list CGI vulnerabilites as part of their top twenty list and provide a link to tool called Whisker. It can be used to test for CGI vulnerabilities. There are also links on the SANs page to many of the CERT advisories on CGI vulnerabilities.

Based on the alerts, 8 MY.NET hosts triggered this alert but to external hosts only. In particular, MY.NET.153.159 triggered this attack 2320 times out of the 2502 times detected only to 209.10.239.135.

Recommendations:
It would be advisable to check MY.NET.153.159 to determine why it is triggering so many outbound alerts.

As a general rule of thumb, any MY.NET host running web services (and therefore potentially CGI scripts) should always be checked for compromise. It is advisable to run vulnerability tools like Whisker to scan MY.NET hosts of this type whenever new content or scripting is posted. Other risk assessments can be made against web servers to confirm they are not vulnerable to

scripting errors or defacements.

Watchlist 000220 IL-ISDNNET-990517
Analysis:
01/25-09:37:47.317800 ** Watchlist 000220 IL-ISDNNET-990517 ** 212.179.18.80:1791 -> MY.NET.150.133:1214
01/25-09:37:48.273096 ** Watchlist 000220 IL-ISDNNET-990517 ** 212.179.18.80:1791 -> MY.NET.150.133:1214
01/25-09:37:49.212311 ** Watchlist 000220 IL-ISDNNET-990517 ** 212.179.18.80:1791 -> MY.NET.150.133:1214

This alert in the past has triggered on gnutella traffic which uses a destination port of 6346. This is not so with the above trace. The traffic is coming from a host in Israel but is destined for port 1214 which is typical of Kazaa file sharing. A practical by Christof Voemel confirms this activity. Highly susipicous. References to several other practicals show Watchlist attempts to 6346. Oddly, the packets above use the same source port, which is also unusual. This indicates packet crafting, since client ports are usually ephemeral.

George Bakos, URL http://openbsd.org.br/ouah/George_Bakos.html#snort
Rick Winkey Yuen practical, URL http://www.giac.org/GCIA.php October 11, 2001.
Christof Voemel practical, URL http://www.giac.org/GCIA.php , September 7th, 2000.

Based on the alerts, all 34 hosts were external and associated with the Israel IP address space. One host 212.179.35.8 triggered this alert 1419 out of the 1807 total alerts detected. Six MY.NET hosts were the targets of this alert as shown.

Occurrences	Targets
1419	MY.NET.151.85
138	MY.NET.88.162
128	MY.NET.150.141
85	MY.NET.150.133
22	MY.NET.153.203
15	MY.NET.150.220

Recommendations:
Check the MY.NET. hosts to which this traffic was directed to. Look for file sharing activity or possible unauthorized software on this host. Monitor MY.NET hosts for further alerts from this source. Filtering or blocking the IP address range from Israel could be a consideration if this traffic continues or presents itself as a threat.

Overall Alert Information
As calculated from the summarized alert information, the top ten talkers are listed by source IP address and destination IP address. Interestingly, six MY.NET.153 hosts were the initiators (sources) of alert activity and represented 50.6 % of the total top ten activity and 17.5 % of the total overall alert activity

Top Ten Talkers (by source and destination)

Occurrences	%	Top Source IP Address	Occurrences	%	Top Destination IP Address
11930	5.84	MY.NET.70.177	55270	32.02	MY.NET.150.198

10740	5.26	63.250.209.34	22737	13.17	MY.NET.151.63
8239	4.03	MY.NET.153.119	12106	7.01	211.115.213.202
7735	3.79	MY.NET.153.114	7693	4.46	MY.NET.11.6
7246	3.55	63.250.210.50	7441	4.31	MY.NET.11.7
5622	2.75	MY.NET.153.123	6417	3.72	MY.NET.152.109
5143	2.52	MY.NET.153.111	3438	1.99	MY.NET.5.96
4970	2.43	63.250.208.34	2576	1.49	64.12.184.141
4611	2.26	MY.NET.153.118	2320	1.34	209.10.239.135
4402	2.16	MY.NET.153.122	2103	1.22	MY.NET.151.114

Also noted was the number of 1 to 3 packet activity. There was not enough time for analysis to determine what was producing single packets to so many hosts and whether these packets were anomlous or represented threatening behavior. The reason this activity may need further review is that various tools, (Nmap, Xprobe to name a few) can sent small numbers of packets for OS fingerprinting. That information gives attackers options on future exploitation of the target. The information below is simply to alert the University of this activity.

Small Packet history Alerted (where 3 or less sent from a source or received by host)

Hosts sending	# of packets sent	Hosts receiving	# of packets received
81	**3**	**72**	**3**
105	**2**	**197**	**2**
219	**1**	**152**	**1**

From external talkers listed above, IP address space and activities

1. Three of the top ten source IP's (63.250.210.34 , 63.250.210.50 and 63.250.208.34) appear to be part of Yahoo addresses. These were involve in MISC Large UDP alerts and SYN-Scans.

```
Yahoo! Broadcast Services, Inc. (NETBLK-NETBLK2-YAHOOBS)
   2914 Taylor st
   Dallas, TX 75226   US
   Netname: NETBLK2-YAHOOBS
   Netblock: 63.250.192.0 - 63.250.223.255
   Maintainer: YAHO
   Coordinator:
      Bonin, Troy  (TB501-ARIN)  netops@broadcast.com
      214.782.4278 ext. 2278
   Domain System inverse mapping provided by:
   NS.BROADCAST.COM          206.190.32.2
   NS2.BROADCAST.COM         206.190.32.3
   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
   Record last updated on 29-Jun-2001.
   Database last updated on  1-Mar-2002 19:57:27 EDT.
```

3. An initial search of ARIN 's WHOIS database yields information about the owner of 211.115.213.202 destination IP. A further search of APNIC's WHOIS database gives the actual owner of the IP block of addresses where 211.115.213.202 resides in Korea. This IP address was involved with 12106 attempts at the IIS Unicode attack and receive 23 SYN scans from MY.NET hosts (MY.NET.153.184, MY.NET.153.119, MY.NET.153.157, MY.NET.153.123) on port 80.

```
Asia Pacific Network Information Center (NETBLK-APNIC-CIDR-BLK)
   These addresses have been further assigned to Asia-Pacific users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   Netname: APNIC-CIDR-BLK2
   Netblock: 210.0.0.0 - 211.255.255.255
   Coordinator:
      Administrator, System  (SA90-ARIN)  [No mailbox]
      +61-7-3367-0490
   Domain System inverse mapping provided by:
   NS.APNIC.NET              203.37.255.97
   SVC00.APNIC.NET           202.12.28.131
   NS.TELSTRA.NET      203.50.0.137
   NS.RIPE.NET               193.0.0.193
   Regional Internet Registry for the Asia-Pacific Region

inetnum              211.104.0.0 - 211.119.255.255
netname              KRNIC-KR
descr                KRNIC
descr                Korea Network Information Center
country              KR
admin-c              HM127-AP, inverse
tech-c               HM127-AP, inverse
remarks              ****************************************
remarks              KRNIC is the National Internet Registry
remarks              in Korea under APNIC. If you would like to
remarks              find assignment information in detail
remarks              please refer to the KRNIC Whois DB
remarks              http://whois.nic.or.kr/english/index.html
remarks              ****************************************
mnt-by               APNIC-HM, inverse
mnt-lower            MNT-KRNIC-AP, inverse
changed              hostmaster@apnic.net 20000414
changed              hostmaster@apnic.net 20010606
source               APNIC
```

3. Top destination (64.12.184.141) is in the IP address ranged owned by AOL. This IP address was involved with 2576 attempts at the IIS Unicode attack and receive 67 SYN scans from MY.NET hosts.

```
America Online, Inc. (NETBLK-AOL-MTC)
   10600 Infantry Ridge Road
   Manassas, VA 20109   US
   Netname: AOL-MTC
   Netblock: 64.12.0.0 - 64.12.255.255
   Coordinator:
      America Online, Inc.  (AOL-NOC-ARIN)  domains@AOL.NET
      703-265-4670
   Domain System inverse mapping provided by:
   DNS-01.NS.AOL.COM         152.163.159.232
   DNS-02.NS.AOL.COM         205.188.157.232
   Record last updated on 16-Dec-1999.
   Database last updated on  1-Mar-2002 19:57:27 EDT.
```

4. Top Ten Destination (209.10.239.135) is located with Globix Corporation. They provide ….

Services as based on their web site information (http://www.globix.com) It recevied 2320 CGI-Null Byte alerts and several SYN-scan packets from MY.NET.153.159. MY.NET.153.159 always sent CGI-Null Byte alerts in sets of 12 packets repeatively.

1. 01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
2. 01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
3. 01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
4. 01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
5. 01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
6. 01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
7. 01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
8. 01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
9. 01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
10 .01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
11. 01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
12. 01/29-20:21:12.189086 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1670 -> 209.10.239.135:80
1. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80
2. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80
3. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80
4. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80
5. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80
6. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80
7. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80
8. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80
9. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80
10. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80
11. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80
12. 01/29-20:21:17.896204 ** spp_http_decode: CGI Null Byte attack detected ** MY.NET.153.159:1674 -> 209.10.239.135:80

```
Globix Corporation (NETBLK-GLOBIXBLK3)
   295 Lafayette St- 3rd Fl
   NY, NY 10012
   US
   Netname: GLOBIXBLK3
   Netblock: 209.10.0.0 - 209.11.223.255
   Maintainer: PFMC
   Coordinator:
      Hostmaster, Globix Corporation  (GCH2-ARIN)  arin-admin@GLOBIX.NET
      +1-212-334-8500 (FAX) 212.334.8615
   Domain System inverse mapping provided by:
   Z1.NS.NYC1.GLOBIX.NET      209.10.66.55
   Z1.NS.SJC1.GLOBIX.NET      209.10.34.55
   Z1.NS.LHR1.GLOBIX.NET      212.111.32.38
   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
   Record last updated on 05-Apr-2001.
   Database last updated on  1-Mar-2002 19:57:27 EDT.
```

OOS Analysis: (Back to top)

Since Out Of Spec (OOS) data is relatively small all packets are shown here from the five day period. In the original data, snaplength was stated at 68 bytes. Four source IP's issued the packets. Day 5 had no OOS data.

Case 1. OOS packet detected

01/25-14:24:25.974971 65.93.180.243:3476 -> MY.NET.88.162:1214
TCP TTL:110 TOS:0x0 ID:48811 DF
SFA* Seq: 0xB3CBCCE1 Ack: 0x5F5698EF Win: 0x13F6
00 00 00 00 00 00 ......

=+

Source IP 65.93.180.243 is owned by Bell Nexxia (High Speed), Quebec as registered in WHOIS using ARIN's database. This is simply summarized and output is not shown here. An NSLookup against one of Bell Nexxia (BellGlobal) DNS servers produced a device associated with 65.93.180.243 listed as Sherbrooke-HSE-ppp3609574.sympatico.ca. It's odd that I was able to look up the name of the device. Devices that send packets of this nature do not always have DNS entries. Normal TCP handshaking requires initial SYN packets to open a session with a host. FIN packets terminate the session gracefully. Crafted packets can have unusual combinations of TCP flags (S-SYN, A-ACK, R-RESET, P-PUSH, U-Urgent, F-FIN).

Associated traffic from 65.93.180.243 was examined in both the scan and alert data. No alerts were generated from this IP source; however this source sent the following scan packets to MY.NET.88.162. They are highly unusual.

```
Jan 25 11:45:17 65.93.180.243:2975 -> MY.NET.88.162:1214 INVALIDACK ***AP*S*
Jan 25 12:55:39 65.93.180.243:3055 -> MY.NET.88.162:1214 SYN ******S*
Jan 25 12:55:36 65.93.180.243:3049 -> MY.NET.88.162:1214 NOACK *****RS*
Jan 25 13:01:31 65.93.180.243:3063 -> MY.NET.88.162:1214 SYN ******S*
Jan 25 13:01:31 65.93.180.243:3059 -> MY.NET.88.162:1214 NOACK *****RS*
Jan 25 13:13:34 65.93.180.243:3104 -> MY.NET.88.162:1214 SYN ******S*
Jan 25 13:13:34 65.93.180.243:3103 -> MY.NET.88.162:1214 NOACK *****RS*
Jan 25 13:13:57 65.93.180.243:3104 -> MY.NET.88.162:1214 NOACK *****RS*
Jan 25 13:15:33 65.93.180.243:3108 -> MY.NET.88.162:1214 SYN ******S*
Jan 25 13:15:33 65.93.180.243:3108 -> MY.NET.88.162:1214 INVALIDACK ***AP*S*
Jan 25 13:16:46 65.93.180.243:3110 -> MY.NET.88.162:1214 SYN ******S*
Jan 25 13:16:43 65.93.180.243:3108 -> MY.NET.88.162:1214 NOACK *****RS*
Jan 25 13:16:47 65.93.180.243:3110 -> MY.NET.88.162:1214 INVALIDACK ***AP*S*
Jan 25 13:16:53 65.93.180.243:3110 -> MY.NET.88.162:1214 INVALIDACK ***AP*S*
Jan 25 13:20:45 65.93.180.243:3141 -> MY.NET.88.162:1214 INVALIDACK ***AP*S*
Jan 25 13:23:51 65.93.180.243:3224 -> MY.NET.88.162:1214 SYN ******S*
Jan 25 13:23:51 65.93.180.243:3141 -> MY.NET.88.162:1214 NOACK *****RS*
Jan 25 14:11:39 65.93.180.243:3456 -> MY.NET.88.162:1214 SYN ******S*
Jan 25 14:11:40 65.93.180.243:3456 -> MY.NET.88.162:1214 INVALIDACK ***AP*S*
Jan 25 14:24:21 65.93.180.243:3476 -> MY.NET.88.162:1214 INVALIDACK ***A**SF
```

It appears that the OOS packet may just be part of the other attempts from 65.93.180.243. This source IP looks to be attempting OS fingering printing by sending various crafted combinations of TCP flags to elicit a response from MY.NET.88.162. The packets are always sent to port 1214 (Kazaa). This is typical of tools like NMAP; by examining the responses from the host to various flags, the source can then guess the host OS and running services. It may be that 65.93.180.243 has been compromised and is being used to OS fingerprint MY.NET.88.162. It would be worthwhile to contact BellNexxia about this source, especially since it is registered in DNS. OS fingerprinting is usually a prelude to an attack. Traffic to MY.NET.88.162 may need to be monitored. Hopefully, MY.NET.88.162 is a well patched system.

Case 2. OOS packets detected

```
01/26-12:28:28.903591 145.236.140.74:1214 -> MY.NET.150.133:1214
TCP TTL:49 TOS:0x0 ID:21505  DF
```

```
21S***** Seq: 0xEE04156C   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 2572653 0 EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/26-12:31:26.534501 145.236.140.74:1251 -> MY.NET.150.133:1214
TCP TTL:49 TOS:0x0 ID:62090  DF
21S***** Seq: 0xF82E745A   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 2590416 0 EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/26-12:39:41.355563 145.236.140.74:1361 -> MY.NET.150.133:1214
TCP TTL:49 TOS:0x0 ID:32438  DF
21S***** Seq: 0x189636AC   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 2639898 0 EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/26-12:43:38.677969 145.236.140.74:1417 -> MY.NET.150.133:1214
TCP TTL:49 TOS:0x0 ID:63693  DF
21S***** Seq: 0x27AA5C02   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 2663631 0 EOL EOL EOL EOL
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Source IP 145.236.140.74 is owned by Hungarian Telecom, Budapest as registered in ARIN's WHOIS database. This is simply summarized and output is not shown here. An NSLookup against one of Hungarian's DNS servers produced a device associated with 145.236.140.74 listed as line-140-74.dial.matav.net. If the name means anything, this may be a dial-in device associated with this company.

Associated traffic from 145.236.140.74 was examined in both the alert and scan data.

```
01/26-12:28:24.960670  ** Queso fingerprint ** 145.236.140.74:1214 -> MY.NET.150.133:1214
01/26-12:31:22.573004  ** Queso fingerprint ** 145.236.140.74:1251 -> MY.NET.150.133:1214
01/26-12:39:37.342370  ** Queso fingerprint ** 145.236.140.74:1361 -> MY.NET.150.133:1214
01/26-12:43:34.640037  ** Queso fingerprint ** 145.236.140.74:1417 -> MY.NET.150.133:1214
Jan 26 12:28:24 145.236.140.74:1214 -> MY.NET.150.133:1214 SYN 12****S* RESERVED BITS
Jan 26 12:31:22 145.236.140.74:1251 -> MY.NET.150.133:1214 SYN 12****S* RESERVED BITS
Jan 26 12:39:37 145.236.140.74:1361 -> MY.NET.150.133:1214 SYN 12****S* RESERVED BITS
Jan 26 12:43:34 145.236.140.74:1417 -> MY.NET.150.133:1214 SYN 12****S* RESERVED BITS
```

The source 145.236.140.74 may be up no good based on OOS, Scan and alert activity which all agree. Normally, scans with reserved bits are highly suspicious and unusual. The Queso alert occurs at the very same time which would indicate that 145.236.140.74 is attempting to fingerprint MY.NET.150.133, but in this case, more information is needed to determine whether this is a real alert or a false positive. This decision is based on correlation with traces provided by Christof Voemel and his mention of a SANS writeup on ECN flags being used in SYN packets. Basically, set ECN flags in SYN packets during the initial TCP handshaking can be considered valid, which can lead to false positives in IDS systems.

- Voemel, Christof, GCIA practical, URL http://www.giac.org/practical/Christof_Voemel_GIAC.txt
- SANS.ORG, "ECN and Its Impact on Intrusion Detection", URL http://www.incidents.org/detect/ecn.php , 5/31/01.

Still, until it's determined whether this is a false positive or real, it would be better to error on the side of caution and thus monitor traffic from this source IP address range to MY.NET hosts. If it

is fingerprinting and real, this IP source may attempt future exploits against MY.NET.150.133.

Case 3. OOS packets detected

```
01/27-06:25:02.953126 65.129.33.127:18245 -> MY.NET.5.96:21536
TCP TTL:21 TOS:0x0 ID:18458  DF
2*SFRP*U Seq: 0x2F62696E   Ack: 0x2F636F6D   Win: 0x6E2F
2E 70 6C 20 48 54 54 50 2F 31                 .pl HTTP/1
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/27-06:25:06.806710 65.129.33.127:18245 -> MY.NET.5.96:21536
TCP TTL:21 TOS:0x0 ID:20506  DF
2*SFRP*U Seq: 0x2F62696E   Ack: 0x2F636F6D   Win: 0x6E2F
2E 70 6C 3F 62 62 61 74 74 3D                 .pl?bbatt=
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Source IP is owned by Qwest Communications as registered in WHOIS using ARIN's database. This is simply summarized and output is not shown here. Associated traffic from 65.93.180.243 was examined in both the scan and alert data. No alerts were detected from this source, however, the scan data agrees with the OOS observed.

```
Jan 27 06:25:04 65.129.33.127:1406 -> MY.NET.5.96:80 SYN ******S*
Jan 27 06:25:01 65.129.33.127:18245 -> MY.NET.5.96:21536 NOACK *2U*PRSF RESERVED BITS
Jan 27 06:25:05 65.129.33.127:18245 -> MY.NET.5.96:21536 NOACK *2U*PRSF RESERVED BITS
Jan 27 06:25:08 65.129.33.127:1407 -> MY.NET.5.96:80 SYN ******S*
Jan 27 06:25:09 65.129.33.127:18245 -> MY.NET.5.96:21536 NOACK *2U*PR*F RESERVED BITS
```

Notice the combination of the TCP flags(Reserved bits on, S-SYN, F-FIN, R-Reset, P-Push, and U-Urgent). Not normal. This is crafted. It looks like the SYN scan in combination with the other OOS is meant to OS fingerprint and/or send possible payload to MY.NET.5.96. The packet with the TCP flag combination looks to have contained some type of payload which is confirmed by the OOS packet hex information. It would be advisable to check MY.NET5.96.
It might also be useful to look at the hex dump of the scan information in addition to the OOS packets to see if there is any additional correlation on the intent of 65.129.33.127.

Case 4.

```
01/28-20:46:33.703718 64.166.209.137 -> MY.NET.88.162
TCP TTL:110 TOS:0x0 ID:33339  DF MF
Frag Offset: 0x0   Frag Size: 0x22
5A 1D 6B 5E 5B 1D 6C 99 22 37 5C 74 DD D3 2A 0C  Z.k^[.l."7\t..*.
C6 7A 15 8E E0 DC 01 2D 3E D6 87 7A D4 83 DF 32  .z.....->..z...2
3D B0                                            =.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
01/28-20:46:33.815778 64.166.209.137 -> MY.NET.88.162
TCP TTL:110 TOS:0x0 ID:33595  DF MF
Frag Offset: 0x0   Frag Size: 0x22
5B DC 9B FC 60 DE C0 BA E9 C4 23 F9 DA E5 95 D9  [...`.....#.....
E5 9A C7 D1 02 A6 EA 8D E4 6F 39 A3 53 B2 EB 18  .........o9.S...
75 57                                            uW
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Source IP 145.236.140.74 is owned by Hungarian Telecom as registered in ARIN's WHOIS database. This is simply summarized and output is not shown here. Associated traffic from 145.236.140.74 was examined in both the alert and scan data. The OOS data is really unusual;

notice that both the DF (don't frag) and MF (more fragments) flags are both set. There is even a fragment offset of 0 and a size of 22 bytes. Crafted and suspicious. It would appear to be OS fingerprinting. Nmap, hping2 and other mapping tools send bogus TCP flag combinations to attempt OS discovery, since different OS's IP stacks respond differently to various flag combinations. Look at the alert below. It correlates to the fact that 64.16.209.137 is sending

01/28-20:48:17.074547 ** ICMP Fragment Reassembly Time Exceeded ** MY.NET.88.162 -> 64.166.209.137
01/28-20:48:17.074622 ** ICMP Fragment Reassembly Time Exceeded ** MY.NET.88.162 -> 64.166.209.137
01/28-20:48:17.078272 ** ICMP Fragment Reassembly Time Exceeded ** MY.NET.88.162 -> 64.166.209.137
01/28-20:48:17.078347 ** ICMP Fragment Reassembly Time Exceeded ** MY.NET.88.162 -> 64.166.209.137
01/28-20:48:18.090002 ** ICMP Fragment Reassembly Time Exceeded ** MY.NET.88.162 -> 64.166.209.137
01/28-20:48:19.076475 ** ICMP Fragment Reassembly Time Exceeded ** MY.NET.88.162 -> 64.166.209.137
01/28-20:48:22.080740 ** ICMP Fragment Reassembly Time Exceeded ** MY.NET.88.162 -> 64.166.209.137
01/28-20:48:29.090724 ** ICMP Fragment Reassembly Time Exceeded ** MY.NET.88.162 -> 64.166.209.137
01/28-20:48:43.135954 ** ICMP Fragment Reassembly Time Exceeded ** MY.NET.88.162 -> 64.166.209.137
01/28-20:49:10.153816 ** ICMP Fragment Reassembly Time Exceeded ** MY.NET.88.162 -> 64.166.209.137
The OOS packet appears crafted.

Scan Analysis: (Back to top)

Based on the collected data there were 1,693,417 scan events over a five day period.

Top ten source IP that sent scan requests

Occurrences	% of top ten	IP Address	Top ports used in order listed
449300	47.94	MY.NET.60.43	123,7000,0,7001
72894	7.78	MY.NET.6.45	7000,0,123,7001
70159	7.49	MY.NET.6.52	7000,0,7001,256
69794	7.45	MY.NET.6.50	7000,0,7001,256
68536	7.31	MY.NET.6.49	7000,0,7001,256
62762	6.70	MY.NET.6.48	7000,0,7001,256
45757	4.88	64.152.108.141	0,7000,2253,7001,256
34393	3.67	64.152.108.142	0.7000,2287,7001,65535,256
32542	3.47	MY.NET.6.53	7000,0,7001,256
31032	3.31	MY.NET.6.60	7000,0,7001,256

Top ten destination host are listed that were scanned. They are all on the MY.NET. In fact, there is an enormous amount of scanning occurring on various sections of MY.NET. Most of the scanning is occurring on the MY.NET.153 segment where four of the top ten scanned hosts are.

Top Ten Hosts and Top Ten Network Segments scanned on MY.NET

Top Host scanned	%	Top Network w/ scanning activity	%
MY.NET.88.163	8	MY.NET.153.0	55
MY.NET.1.3	6	MY.NET.1.0	9

MY.NET.1.4	4	MY.NET.152.0	9
MY.NET.6.45	3	MY.NET.6.0	8
MY.NET.60.43	3	My.NET.88.0	8
MY.NET.153.194	2	MY.NET.60.0	3
MY.NET.153.172	2	MY.NET.151.0	2
MY.NET.153.210	2	MY.NET.11.0	2
MY.NET.153.209	2	MY.NET.5.0	2
MY.NET.153.173	2		

Top seven ports of interest used by source IP 's detected scanning the network were ports**: 123, 7000, 7001 , 0, 6970, 6112, 137 .** http://www.iana.org/assignments/port-numbers

Top ten scanned ports (destination)

Port	IANA associated service
80	**http**
7000	**afs3-fileserver**
7001	**afs3-callback**
53	**Dns**
6112	**Dtspcd**
0	**Reserved**
111	**Sun RPC**
123	**Ntp**
1214	**Kazaa**

From the table above five source hosts were selected for further evaluation MY.NET.88.163, MY.NET.1.4, MY.NET.60.43, MY.NET.153.194 and MY.NET.153.194. MY.NET.88.163 was chosen because it received the most scan packets. MY.NET.1.4 was chosen because it was receiving packets on port 53 which is used for DNS traffic. MY.NET.153.194 and MY.NET.153.194 were chosen because the MY.NET.153.0 network received the most scanned traffic overall and these two hosts were the top destinations on that network. As it turns out, MY.NET.60.43 was also evaluated because it was participating and initiating the scanning traffic amongst the other hosts in addition to receiving scanned traffic**.**

MY.NET.88.163 Host (Top scanned host)

This host is very busy sending SYN scans and receiving UDP scans from various hosts. It would appear to be compromised. There is no SYN-ACK or ACK recorded as a part of the standard TCP three-way handshake that would considered to be normal traffic. The host is simply sending out dozens of SYN packets and then going into a period of receiving UDP packets.

Jan 25 07:12:47 MY.NET.88.163:1078 -> 207.46.230.181:80 SYN ******S*
Jan 25 07:12:47 MY.NET.88.163:1062 -> 205.138.3.230:80 SYN ******S***
Jan 25 07:12:48 MY.NET.88.163:1085 -> 207.68.182.125:80 SYN ******S***
Jan 25 07:12:48 MY.NET.88.163:1086 -> 63.162.230.35:80 SYN ******S***
Jan 25 07:13:52 MY.NET.88.163:1151 -> 64.58.76.171:80 SYN ******S***
Jan 25 07:13:54 MY.NET.88.163:1158 -> 63.250.208.56:80 SYN ******S***

Jan 25 07:13:55 MY.NET.88.163:1176 -> 131.118.254.38:80 SYN ******S*
Jan 25 07:13:54 MY.NET.88.163:1162 -> 64.58.77.175:80 SYN ******S*
Jan 25 07:21:56 MY.NET.88.163:1043 -> 207.46.131.80:80 SYN ******S*
Continues……

The numerous UDP scans from 66.38.171.142 and 64.152.104.141 are examples below; several of the scans appear to be crafted as the sending and receiving ports have be set to 0. Other crafted packets show that send and receive ports do not change (1787 and 1189). Both types of activity are highly suspicious. It is possible that MY.NET.88.163 has been compromised and is being used to scan other potential victims. More information may uncover any covert channels being manipulated by these hosts.

Jan 25 07:18:28 66.38.171.142:17191 -> MY.NET.88.163:39610 UDP
Jan 25 07:18:30 66.38.171.142:0 -> MY.NET.88.163:0 UDP
Jan 25 07:18:30 66.38.171.142:1787 -> MY.NET.88.163:1189 UDP
Jan 25 07:18:31 66.38.171.142:1787 -> MY.NET.88.163:1189 UDP
Jan 25 07:18:31 66.38.171.142:35922 -> MY.NET.88.163:64145 UDP
Jan 25 07:18:34 66.38.171.142:594 -> MY.NET.88.163:21139 UDP
Jan 25 07:18:35 66.38.171.142:0 -> MY.NET.88.163:0 UDP
Jan 25 07:18:37 66.38.171.142:1787 -> MY.NET.88.163:1189 UDP
Jan 25 07:18:39 66.38.171.142:0 -> MY.NET.88.163:0 UDP
Jan 25 07:18:42 66.38.171.142:0 -> MY.NET.88.163:0 UDP
Jan 25 07:18:45 66.38.171.142:44258 -> MY.NET.88.163:63430 UDP
Jan 25 07:18:46 66.38.171.142:0 -> MY.NET.88.163:0 UDP
Jan 25 07:18:47 66.38.171.142:1787 -> MY.NET.88.163:1189 UDP
Jan 25 07:18:50 66.38.171.142:0 -> MY.NET.88.163:0 UDP

Continues…
Jan 28 07:27:48 64.152.108.141:2253 -> MY.NET.88.163:1183 UDP
Jan 28 07:27:51 64.152.108.141:2253 -> MY.NET.88.163:1183 UDP
Jan 28 07:27:51 64.152.108.141:43184 -> MY.NET.88.163:43522 UDP
Jan 28 07:27:52 64.152.108.141:0 -> MY.NET.88.163:0 UDP
Jan 28 07:27:55 64.152.108.141:10 -> MY.NET.88.163:16079 UDP
Jan 28 07:27:56 64.152.108.141:2253 -> MY.NET.88.163:1183 UDP
Jan 28 07:27:57 64.152.108.141:0 -> MY.NET.88.163:0 UDP
Jan 28 07:27:59 64.152.108.141:2253 -> MY.NET.88.163:1183 UDP

Recommendations:
This host should be investigated and as necessary, removed from the network.

MY.NET.60.43 Host (Top scanning source)
Jan 25 00:00:01 MY.NET.60.43:123 -> MY.NET.153.153:1252 UDP
Jan 25 00:00:01 MY.NET.60.43:123 -> MY.NET.153.204:1252 UDP
Jan 25 00:00:01 MY.NET.60.43:123 -> MY.NET.153.148:1492 UDP
Jan 25 00:00:03 MY.NET.60.43:123 -> MY.NET.153.182:1446 UDP
Jan 25 00:00:04 MY.NET.60.43:123 -> MY.NET.153.210:1605 UDP
Jan 25 00:00:05 MY.NET.60.43:123 -> MY.NET.153.191:1266 UDP
Jan 25 00:00:07 MY.NET.60.43:123 -> MY.NET.153.141:1361 UDP

Over a five day period MY.NET.60.43 has sent 407236 packets to 77 different hosts. This is the

largest number of scan packets (47.94%) and represents an enormous amount of traffic. It is highly out of the ordinary. This host should immediately be pulled off the network. A detailed examination of this host is in order to determine what software has been installed. It may be necessary to rebuild this host.

MY.NET.1.4

This host appears to be receiving numerous UDP packets from multiple hosts on port 123 (NTP - network time procotol) and port 53 (DNS queries) as listed by IANA. It may be unusual to see source and destination IP using the same ports; however DNS is a protocol that does do this). The port 53 UDP activity can indicate that other devices are querying this host for name resolution; however DNS traces usually look different from what is shown. They include answer id and record information. This just appears to be a scan against the DNS port in addition to port 123. The scans on look crafted, but more information is needed.

Jan 25 06:22:12 MY.NET.149.103:123 -> MY.NET.1.4:123 UDP
Jan 25 06:34:03 MY.NET.149.95:123 -> MY.NET.1.4:123 UDP
Jan 25 06:37:08 MY.NET.149.103:123 -> MY.NET.1.4:123 UDP
Jan 25 06:48:59 MY.NET.149.95:123 -> MY.NET.1.4:123 UDP
Jan 25 06:56:55 MY.NET.150.75:1085 -> MY.NET.1.4:53 UDP
Jan 25 06:57:02 MY.NET.150.75:1105 -> MY.NET.1.4:53 UDP …
Jan 25 08:06:02 MY.NET.153.150:1736 -> MY.NET.1.4:53 UDP
Jan 25 08:06:08 MY.NET.153.150:1742 -> MY.NET.1.4:53 UDP…
Jan 25 08:08:53 MY.NET.153.171:2023 -> MY.NET.1.4:53 UDP
Jan 25 08:08:59 MY.NET.153.171:2029 -> MY.NET.1.4:53 UDP…

Recommendations:

Monitor all traffic to MY.NET.1.4 for a period of time if possible. Check MY.NET.1.4 for any unusual files or listening ports. Review OS logs on MY.NET.1.4.

MY.NET.153.194 and MY.NET.153.172

Hosts MY.NET.153.194 and MY.NET.153.172 both appear to be receiving scans from MY.NET.60.43 using port 123. This appears to be crafted packets sent at 1 second intervals. This app After a period of time, UDP packets from MY.NET.153.194 are sent to MY.NET.60.43 on 7001 and receiving on 7000.

Jan 25 00:39:46 MY.NET.60.43:123 -> MY.NET.153.194:1277 UDP
Jan 25 00:40:46 MY.NET.60.43:123 -> MY.NET.153.194:1278 UDP
Jan 25 00:41:46 MY.NET.60.43:123 -> MY.NET.153.194:1279 UDP
Jan 25 00:42:46 MY.NET.60.43:123 -> MY.NET.153.194:1280 UDP
Jan 25 00:43:46 MY.NET.60.43:123 -> MY.NET.153.194:1281 UDP
Jan 25 00:44:46 MY.NET.60.43:123 -> MY.NET.153.194:1282 UDP
Jan 25 00:45:46 MY.NET.60.43:123 -> MY.NET.153.194:1283 UDP

Jan 25 09:07:10 MY.NET.153.194:1804 -> 207.46.230.220:80 SYN ******S*
Jan 25 09:07:11 MY.NET.153.194:1811 -> MY.NET.1.4:53 UDP
Jan 25 09:07:11 MY.NET.153.194:1810 -> 207.68.172.246:80 SYN ******S*
Jan 25 09:07:11 MY.NET.153.194:1813 -> 207.68.177.126:80 SYN ******S*
Jan 25 09:07:17 MY.NET.153.194:1818 -> MY.NET.1.4:53 UDP

The host MY.NET.153.172 has the same appearance as 153.194.

Jan 25 00:00:23 MY.NET.60.43:123 -> MY.NET.153.172:1420 UDP
Jan 25 00:01:23 MY.NET.60.43:123 -> MY.NET.153.172:1421 UDP
Jan 25 00:02:23 MY.NET.60.43:123 -> MY.NET.153.172:1422 UDP

Jan 25 00:03:23 MY.NET.60.43:123 -> MY.NET.153.172:1423 UDP
….

Recommendations:
Determine why MY.NET.60.43 is scanning these hosts. MY.NET.60.43, MY.NET.153.172 and MY.NET.153.194 should be examined.

Analysis Methodology: (Back to top)
I initially was interested in using MYSQL because I felt a relational database would provide the best correlation. Unfortunately, due to time constraints and unfamiliarity with MYSQL, I was not able to use it. If I have to do this again, I would take the time to use MYSQL, snortsnarf or create scripts for the Unix commands I used, since the commands are rather repetitious. It however was a good exercise because it made me focus on what I was looking for and whether or not it made any sense. Basically, I used cat, grep, cut, awk, sort, and uniq. I also used several Excel spreadsheets to do some simply calculations. Several student's GIAC papers were helpful in getting started with the actual analysis.
Rickey Winkey Yuen practical ,URL http://www.giac.org/GCIA.php
Gregory Lajon practical, URL http://www.giac.org/GCIA.php

Examples below:
First I concated each category of IDS files.
cat alert*.* > alerts_all
cat oos*.* > oos_all
cat scans*.* > scans_all

Total statistics
cat alerts_all | wc -l (same with scans)
cat scanall | wc -l

Then starting with the overall alert file I removed the spp_portscan alerts so that I could focus on the non scan alerts
#cat alert-all | grep 'spp_portscan' > alert_nospp

Count total alerts left over
cat alert_nospp | wc -l

Start looking for alert titles, sort alphabetically and count numbers of times, print to sortalert file
cat alert_nospp | awk -F'* '{print $3} | sort | uniq -c | sort -nr > sortedalerts

Find top ten overall source ips (removes dst ip by searching for port if listed with source ip or by -> if no port listed.
cat alert_nospp | awk -F'** '{print $5}' | awk -F':' '{print $1}' | awk -F'-' '{print $1}'| sort | uniq -c | sort -nr > srcip_alert

find top ten overall dst ips

```
# cat alert_nospp | awk -F'\*\* '{print $5}' | awk -F'>' '{print $2}' | awk -F':' '{print $1}' | sort | uniq -c | sort -nr > dstip_alert
#
```

Build Excel spreadsheet of sources and destinations, top alerts.

#

Look for a specific alert, find source, then destinations

```
#cat alert_nospp | grep 'connect to 515' | awk -F'*' '{print $2}' | awk -F':' {print $1}' | sort | uniq -c | sort -nr | more
```

Show specific examples of alerts for a specific source or host

```
# cat alert_nospp | grep 'connect to 515' |  awk -F'*' '{print $2}' | grep 'MY.NET.x.x' | more
```

where x.x is the rest host IP.

Find alerts initiated by external hosts

```
# cat alert_nospp | grep 'connect to 515' |  awk -F'*' '{print $2}' | grep -v 'MY.NET' | more
```

Similar types and combinations were done on the scan data.

General References:

(Back to top)

The following were very useful for background material on the detects and analysis this assignments. You must select the person's name at this site.

Lajon, Gregory, GIAC practical, URL http://www.giac.org/GCIA.php
Voemel, Christof, GIAC practical, URL http://www.giac.org/GCIA.php
Yuen, Rick Winkey, GIAC practical, URL http://www.giac.org/GCIA.php
Grazi, Alberto, GIAC practical, URL http://www.giac.org/GCIA.php
Gray, Shelby, GIAC practical, URL http://www.giac.org/GCIA.php

The following web sites were very useful.

http://www.sans.org , http://www.incidents.org, http://www.cert.org, http://www.mitre.org, http://www.google.com , http://icat.nist.gov

The following books were very useful.

Stevens, Richard, "TCP/IP Illustrated, Volume 1.", Addison-Wesley, 1994.
Northcutt, Stephen, and Novak, Judy, "Network Intrusion Detection, An Analyst's Handbook", New Riders, 2000.
Northcutt, Stephen, Cooper, Mark, Fearnow, Matt, and Frederick, Karen, "Intrusion Signatures and Analysis", New Riders, 2001.