Intrusion Detection in Depth
GCIA Practical Assignment
Version 3.0

Keith Alexander

Assignment 1- Describe the State of Intrusion Detection

What is active response

Active Response is a mechanism in intrusion detection systems (IDS) that provides the IDS with capability to respond to an attack when it has been detected. There are two methods that the IDS can take to circumvent an attack. The first method of circumventing attacks would be Session disruption, and the second is Filter rule manipulation. The specific feature varies with each IDS product and each countermeasure method possesses its own strengths and weaknesses.

Method 1- Session disruption

Session disruption is the most popular method of circumvention because of the ease of its implementation. Depending on the type of session established, UDP or TCP, an IDS that is configured for session disruption can reset or knock down the established connection. This does not prevent the attacker from launching additional attacks, but it does prevent the attacker from causing any further damage in conjunction with the "broke" session. When using the session disruption method, if an attacker launches subsequent attacks, the IDS must continually attempt to close every initiated attack session.

With sessions disruption the IDS uses different methods for breaking the connection depending on the type of traffic it sees. If an attacker uses TCP sessions, they are reset by RST packet that is sent to reset one or both hosts in a session from the IDS. In the case of UDP, a session can be broken by sending various ICMP packets to the host from the IDS box.

Why might the IDS send RSTs to the attacker and victim host?

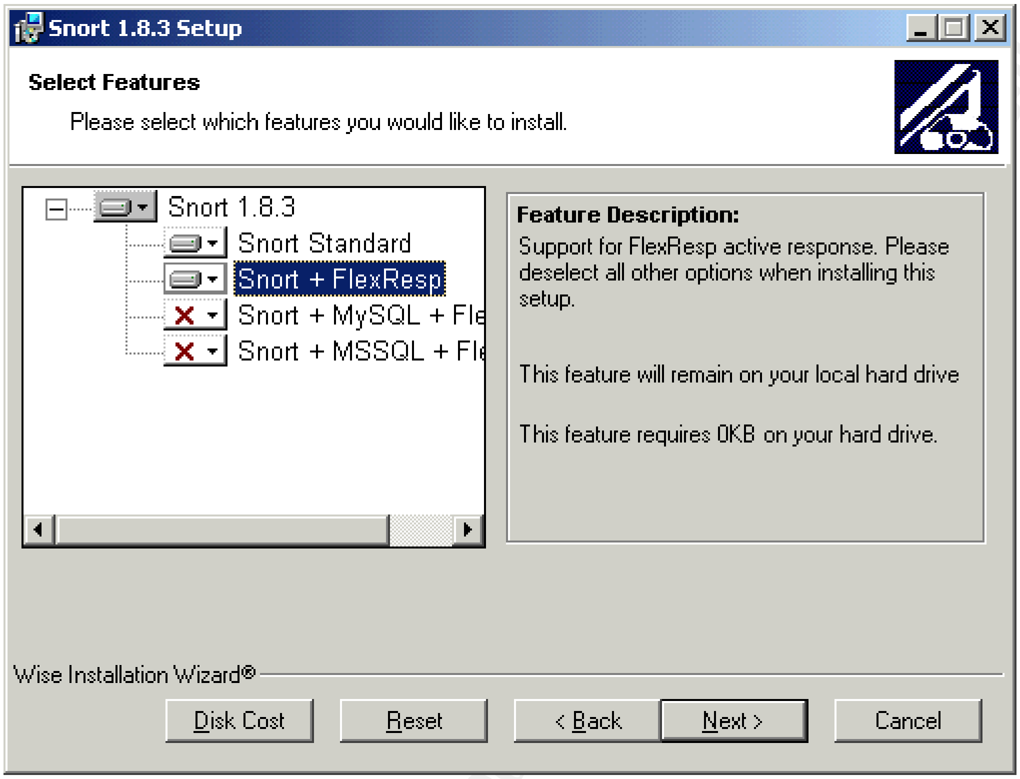
An IDS might send a TCP RST packet to an attacker and victim after detecting malicious traffic like an established Sub seven connection.

There are a few IDS systems that provide the session disruption, but for discussion I will focus on Snort, which is a lightweight network intrusion system that runs on different platforms. When Snort is configured with the Flexresp feature it provides session disruption. Flexresp is a feature that allows Snort to automatically respond to an attack if the corresponding option is specified in the snort rule. In order to enable active response on Unix, Snort must be compiled with Flexresp enable as shown below.

```
Configure –enable-flexresp
```

When installing on a Win32 system, Flexresp is enabled by selecting the Snort +FlexResp option as shown in Fig 1.1 below.

Fig. 1.1



Snort 1.8.3 Setup

Select Features

Please select which features you would like to install.

- Snort 1.8.3
  - Snort Standard
  - Snort + FlexResp
  - Snort + MySQL + Fle
  - Snort + MSSQL + Fle

Feature Description:
Support for FlexResp active response. Please deselect all other options when installing this setup.

This feature will remain on your local hard drive

This feature requires 0KB on your hard drive.

<u>Below in Fig 1.2 is an example of a Snort rule configured to respond to an attack</u>

Fig 1.2

```
                    Rule Header                         Rule Options
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-IIS
CodeRed v2 root.exe access"; flags: A+; uricontent:"scripts/root.exe?";
nocase;resp:rst_snd;)

alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN Webtrends
Scanner UDP Probe"; content: "|0A 68 65 6C 70 0A 71 75 69 74 0A|";
resp:icmp_port,icmp_host;)
```

Rules define what traffic snort considers as hostile and consists of two parts, Rule Header and Rule Options. The Rule Header contains an action field, protocol field, source IP and port fields, direction of traffic field and destination IP and port fields, which all basically define who is involved. The Rule options define what packet attributes to search to consider if traffic is hostile.

When we examine the Rule Header of the first rule in Fig 1.2 we see that Snort will alert us to any TCP session connecting to the web server at port 80. Let's look at the second part of the rule, the Rule Options. Snort inspects packets that meet the Rule header requirements for the TCP ACK flag and any other TCP flags that are set, and searches the payload for the character string scripts/root.exe. The resp:rst_snd value sends a forge packet with the TCP reset flag set to the sender.

The second rule is for any Webtrends UDP Scans with the character string of "0A 68 65 6C 70 0A 71 75 69 74 0A" in the payload. If this rule meet both the header rule and option rule the flexresp values **icmp_port,icmp_host** tells snort to send an icmp port unreachable and host unreachable packet to knock down connection.

Why does Snort send ICMP packets to UDP stimulus?

ICMP packets are sent to a host initiating a UDP connection to inform the sender that a requested port/host is unavailable. The reason ICMP packets are sent to a UDP stimulus is UDP does not have the capability to report errors, so ICMP is used to assist. Snort use this normal process to send a spoofed ICMP packet to the host initiating the connection in an attempt to fool the host in to thinking that the host is unavailable.

Session disruptions in action

Snort Rule

```
alert tcp 192.168.1.1 any > $HOME_NET 135 (msg:"Block host";
flags:S+; resp:rst_snd;)
```

This rule was created to rest any TCP session initiated by host 192.168.1.1 with the SYN TCP flag and any other TCP flags set.

The traffic below was generated in my lab between two machines. The targeted pc is configured with Snort 1.8.3 for Win32 systems and runs on windows 2000 professional. The attacking host is a Red Hat Linux 7.0 machine. Nmap was used to port scan the target machine by typing nmap –p 135 –sF 192.168.1.2, which triggered alert.

Tcpdump snip

```
 08:17:23.477034 Attacker.4634 > Target.135: S 3719449388:3719449388(0) win 5840 <mss
1460,sackOK,timestamp 733205394 0,nop,wscale 0> (DF)

08:17:23.477203 Target.135 > Attacker.4634: R 0:0(0) ack 3719449389 win 0

08:17:23.477275 Attacker.4635 > Target.135: S 3715810638:3715810638(0) win 5840 <mss
1460,sackOK,timestamp 733205394 0,nop,wscale 0> (DF)

08:17:23.477346 Target.135 > Attacker.4635: R 0:0(0) ack 3715810639 win 0
```

There are a few techniques which can be used that allow an attacker to bypass session disruption enable IDS. An attacker with basic knowledge of TCP/IP can defeat this mechanism as stated in a paper by Jason Larsen and Jed Haile on Understanding IDS Active Response Mechanisms. Here they wrote about techniques that could defeat session disruption. One of the methods they talked about was trying to have the host disregard the tcp reset packet sent from the IDS system. The session disruption bypass techniques took advantage of the time it took for the IDS to examine network traffic, detect an exploit and respond to an attack. Also the tcp stack and the way it receives data were used to circumvent session disruption.

An attacker could also attack the IDS with a Denial of Service in an attempt to crash the machine or starve it of it's resources and render the use of session disruption. Any Evasion techniques where an attacker tries to prevent the IDS from detecting the rule would also work. Session disruption is only useful when the IDS can identify the traffic.

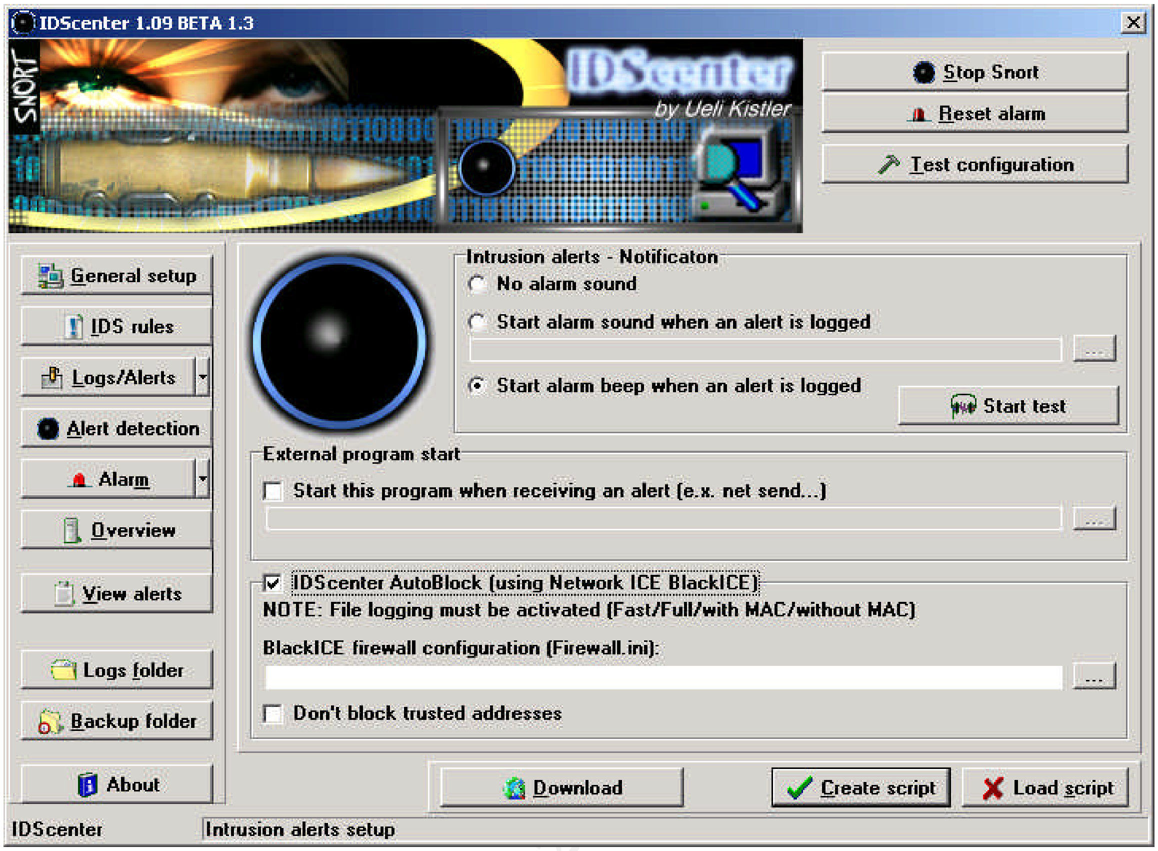
Method 2- Filter rule manipulation

The second countermeasure is filter rule manipulation. This mechanism works by modifying the access control list (acl) on a firewall or router. Filter rule manipulation block the IP of the attacker preventing any further attacks. This option should be used with extreme care, because an attacker can use it to Dos the network. If an attacker used the IP address of a partner they could spoof the address. When the IDS sees the attack and goes to respond, it would block your partner access.

There are a few IDS products that provide filter rule manipulation. Real Secure has the ability to modify Checkpoint firewall. Cisco Intrusion Detection System (IDS), formerly known as Cisco NetRanger, is a hardware based IDS that can respond to an attack by adding an access control list to a router.

Snort can provide filter rule manipulation when used with IDScenter, a tool used to manage snort, when run on a Win32 systems and BlackICE Defender. Attackers are blocked by IDScenter after an alert is triggered which modifies the file firewall.ini access lists that is used by BlackICE Defender, a personal desktop firewall that only protects the machine it is installed on. This can be accomplished by checking on the IDScenter Auto block box as shown in Fig 1.3 below.

Fig 1.3



Intrusion alerts - Notificaton

○ No alarm sound

○ Start alarm sound when an alert is logged

● Start alarm beep when an alert is logged

External program start

□ Start this program when receiving an alert (e.x. net send...)

☑ IDScenter AutoBlock (using Network ICE BlackICE)

NOTE: File logging must be activated (Fast/Full/with MAC/without MAC)

BlackICE firewall configuration (Firewall.ini):

□ Don't block trusted addresses

Then you provide the path to the firewall.ini file that is used by BlackICE, which is C:\Program files\Network ICE\BlackICE by default.

This method can be evaded by tricking the user behind the firewall in to installing a backdoor via email. Once the backdoor is installed the attacker can remotely admin PC and can launch attacks from within. Jason Larsen and Jed Haile wrote paper on Understanding IDS Active Response Mechanisms mentions launching an attack with a spoof address of an popular website like CNN.com, AOL.com, and Ebay.com. This would block traffic from these sites to enter your site. Users would call the helpdesk about not being able to access site and demand a resolution. This would result in the disabling of the rule manipulation feature allowing the attacker to attack without blocking.

Conclusion

Active Response mechanisms is an effective tool when used within its limitation. When used in conjunction with other network security devices it enhances network security. Session disruption should not be configured to respond to every alert just serious attack like Denial of service. Rule manipulation should be used with care because of the effect it could cause if turn on a network. Active Response is by no means meant to be fool proof.

REFERENCE:

Larsen, Jason, & Haile, Jed. Understanding IDS Active Response Mechanisms
http://online.securityfocus.com/infocus/1540

Brenton, Chris. Mastering Network Security. Pages 263 – 265

ISS RealSecure 6.5 FAQ
http://documents.iss.net/literature/RealSecure/RS6.5ExternalFAQ.pdf

Cisco Secure Intrusion Detection Family Overview
http://www.cisco.com/warp/public/cc/pd/sqsw/sqidsz/prodlit/idsf_ov.htm

Roesch, Martin. Intrusion Detection: Snort Style. Pages 171 - 179

Ptacek, Thomas H, Newsham, Timothy N. Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection
http://www.snort.org/docs/idspaper/

Assignment 2 – Network Detects

Detect #1: Port Scan

```
Feb  4 10:36:54 211.20.221.61:2111 -> xxx.yyy.1.1:21 SYN ******S*
Feb  4 10:36:54 211.20.221.61:2116 -> xxx.yyy.1.11:21 SYN ******S*
Feb  4 10:36:57 211.20.221.61:2128 -> xxx.yyy.1.31:21 SYN ******S*
Feb  4 10:36:54 211.20.221.61:2121 -> xxx.yyy.1.6:21 SYN ******S*
Feb  4 10:36:54 211.20.221.61:2136 -> xxx.yyy.1.23:21 SYN ******S*
Feb  4 10:36:54 211.20.221.61:2171 -> xxx.yyy.1.41:21 SYN ******S*
Feb  4 10:36:57 211.20.221.61:2126 -> xxx.yyy.1.16:21 SYN ******S*
Feb  4 10:36:57 211.20.221.61:2127 -> xxx.yyy.1.32:21 SYN ******S*
[...]
Feb  4 11:08:33 211.20.221.61:2792 -> xxx.yyy.255.237:21 SYN ******S*
Feb  4 11:08:33 211.20.221.61:2795 -> xxx.yyy.255.234:21 SYN ******S*
Feb  4 11:08:33 211.20.221.61:2798 -> xxx.yyy.255.239:21 SYN ******S*
Feb  4 11:08:33 211.20.221.61:2805 -> xxx.yyy.255.244:21 SYN ******S*
Feb  4 11:08:33 211.20.221.61:2809 -> xxx.yyy.255.246:21 SYN ******S*
Feb  4 11:08:33 211.20.221.61:2826 -> xxx.yyy.255.252:21 SYN ******S*
Feb  4 11:08:34 211.20.221.61:2828 -> xxx.yyy.255.253:21 SYN ******S*
Feb  4 11:08:34 211.20.221.61:2833 -> xxx.yyy.255.255:21 SYN ******S*
23957

Feb  4 20:12:16 218.76.0.156:4519 -> xxx.yyy.241.29:80 SYN ******S*
Feb  4 20:12:15 218.76.0.156:4410 -> xxx.yyy.78.252:80 SYN ******S*
Feb  4 20:12:13 218.76.0.156:4416 -> xxx.yyy.147.101:80 SYN ******S*
Feb  4 20:12:13 218.76.0.156:4418 -> xxx.yyy.170.221:80 SYN ******S*
Feb  4 20:12:13 218.76.0.156:4420 -> xxx.yyy.193.85:80 SYN ******S*
Feb  4 20:12:13 218.76.0.156:4440 -> xxx.yyy.169.5:80 SYN ******S*
Feb  4 20:12:13 218.76.0.156:4442 -> xxx.yyy.192.125:80 SYN ******S*
Feb  4 20:12:15 218.76.0.156:4454 -> xxx.yyy.75.77:80 SYN ******S*
[...]
Feb  4 21:08:32 218.76.0.156:1037 -> xxx.yyy.163.140:80 SYN ******S*
Feb  4 21:08:32 218.76.0.156:1042 -> xxx.yyy.62.136:80 SYN ******S*
Feb  4 21:08:32 218.76.0.156:1110 -> xxx.yyy.220.241:80 SYN ******S*
Feb  4 21:08:32 218.76.0.156:1056 -> xxx.yyy.128.249:80 SYN ******S*
Feb  4 21:08:33 218.76.0.156:1088 -> xxx.yyy.233.15:80 SYN ******S*
Feb  4 21:08:33 218.76.0.156:1097 -> xxx.yyy.229.45:80 SYN ******S*
Feb  4 21:08:33 218.76.0.156:1096 -> xxx.yyy.90.113:80 SYN ******S*
Feb  4 21:08:33 218.76.0.156:1100 -> xxx.yyy.79.136:80 SYN ******S*
9491
```

Source of Trace:

This trace was posted on incident.org by Ken Connelly on Feb 5 2002.
http://www.incidents.org/archives/intrusions/date36.html

Detect was generated by:

This log looks like the portscan file that's generated by Snort. I do not know the version of snort that is running or the rule set that was used since that information is not provided.

The format of the portscan file is listed below.

```
[   Time stamp   ]    [Source IP and port]    [Destination IP and port]  [UDP/TCP]
 Feb  4 21:08:33       218.76.0.156:1096  ->   xxx.yyy.90.113:80        SYN ******S*
```

Portscans alerts are trigger by the following on systems running Snort:

preprocessor portscan: $HOME_NET 4 3 portscan.log

Preprocessors allow Snort to examine and manipulate network traffic. Preprocessor are configured in the snort.conf file. This preprocessors look for Portscans that connect to 4 or more ports in three seconds and output the results to the portscan.log file.

Probability the source address was spoofed:

The probability the source is spoofed is low. This conclusion is drawn based on the information that is attempting to be gathered for reconnaissance purpose would return to the spoofed host rather than the attacker who spoofed the host. This traffic appears to be a reconnaissance for the web and FTP servers. The goal of a reconnaissance is to gain information about a host to properly plan an attack, and this cannot be accomplished if the address is forged. However the possibility exists that the attacking IP could be forged the attacker would not receive the information unless they were on the same physical network segment, and sniffing. But this would be very difficult to do in a switch environment.

Description of attack:

This is a reconnaissance of FTP and HTTP servers. Scans of port 21 and 80 are quite common, not to mention attacks of these ports. Incident.org list these two ports as the most attacked ports. Port 80 is classified as the number 1 attacked port while port 21 occupies number 2 position at the time of this practical. There are also many CVE that are associated with attacks directed at these ports. The vulnerabilities vary according to the operating system and the type of FTP and HTTP servers.

CVE for FTP

Name	Description

CVE-1999-0017	FTP servers can allow an attacker to connect to arbitrary ports on machines other than the FTP client, aka FTP bounce.
CVE-1999-0035	Race condition in signal handling routine in ftpd, allowing read/write arbitrary files.
CVE-1999-0054	Sun's ftpd daemon can be subjected to a denial of service.
CVE-1999-0075	PASV core dump in wu-ftpd daemon when attacker uses a QUOTE PASV command after specifying a username and password.
CVE-1999-0079	Remote attackers can cause a denial of service in FTP by issuing multiple PASV commands, causing the server to run out of available ports.
CVE-1999-0080	wu-ftp FTP server allows root access via "site exec" command.

CVE for HTTP

Name	Description
CVE-1999-0071	Apache httpd cookie buffer overflow for versions 1.1.1 and earlier.
CVE-1999-0236	ScriptAlias directory in NCSA and Apache httpd allowed attackers to read CGI programs.
CVE-1999-0267	Buffer overflow in NCSA HTTP daemon v1.3 allows remote command execution.
CVE-1999-0378	InterScan VirusWall for Solaris doesn't scan files for viruses when a single HTTP request includes two GET commands.

Attack mechanism:

Is this a stimulus or response?
This traffic is a stimulus because the attacker sent a TCP packet with only the SYN flag set indicating that they initiated connection.

What services is being targeted?
Http (TCP port 80) and Ftp (TCP port 21)

Does the service have known vulnerabilities or exposures?

Yes, they're many known vulnerabilities for these services running on various operating systems.

Is this benign, an exploit, a denial of service, or reconnaissance?
This is a reconnaissance.

This scan could have been a SYN scan but without seeing a the full port scan for signs if the target responded.   If was SYN scan it would work by sending TCP SYN packets to a host to find out what ports are listening, this scanning technique is also known as half-open scanning. The attacker is interested in what ports are listening because each listening port has a services running on it. In this scan we see the attacker is scanning for FTP (port 21) and HTTP (port 80). Half open scanning works by sending a SYN packet to a host port. If the remote port is listening it will send a SYN-ACK packet, this is normally the second step in the TCP Three way handshake, or a RST packet if the port not listening. There are many tools that can perform this type of scan. Additionally, it appears the attacker is a script kiddie based upon the traffic below.

```
Feb  4 11:08:34 211.20.221.61:2833 -> xxx.yyy.255.255:21 SYN ******S*
```

xxx.yyy.255.255 would be broadcast address.

Correlation:

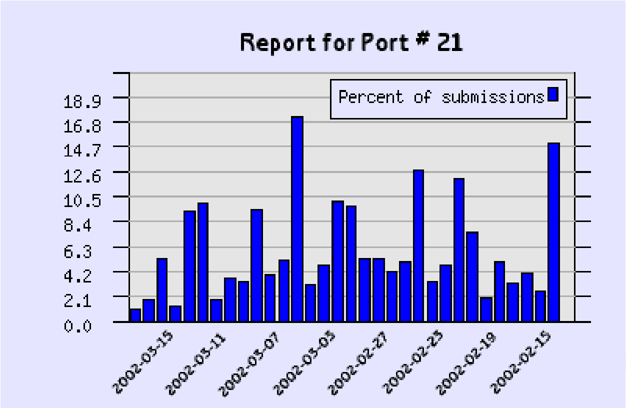
Scans of port 21 and 80 are quite common, not to mention attacks of these ports. Below are charts from http://www.incidents.org/ showing activity for port FTP and HTTP, which is their top 2 ports attacked.

Report for Port # 80 - HTTP

Date	Count	Percent of Submissions	
2002-03-15	414673	83.29%	
2002-03-14	614810	60.03%	
2002-03-13	607980	56.67%	
2002-03-12	576797	59.18%	
2002-03-11	607066	50.20%	

2002-03-10	726448	53.57%	
2002-03-09	593759	59.42%	
2002-03-08	412236	47.86%	
2002-03-07	326485	38.05%	
2002-03-06	217215	30.54%	
2002-03-05	172767	28.67%	|
2002-03-04	178379	30.26%	
2002-03-03	169514	20.94%	|
2002-03-02	209533	33.44%	
2002-03-01	154753	19.18%	|

Report for Port # 21 - FTP



<u>Evidence of active targeting:</u>

There are no signs of active targeting by attacker. While this may not be the case it appears the attacker is scanning the network seeking to find systems to exploit. The attacker is most likely running a script that scans for open ports on 21 and 80.

<u>Severity:</u>

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

(3+2) – (1+1) = 3

Criticality

3 = The criticality of this attack would be moderate since the host could have both these services running.

Lethality

2 = The lethality is a two since this is just a portscan.

System countermeasure

1 = The system countermeasures in this case would be safe to assume there is no protection on the host system. Additionally there are many targets on this network and at least 1 could be open.

Network countermeasure

1 = The level of protection at the network layer does not appear to be present. Though there is no conclusive evidence in this instance. For this analysis I will assume there is no network protection. This could change if there was some sort of firewall/or host-based protection measures in place.

Defensive recommendation:

- Make sure the operating system, ftp, and http servers are patched.
- Update IDS with current signatures to verify attacks.
- Check FTP and HTTP server configurations for server misconfiguration.
- Use a stateful proxy firewall that protects the web and ftp servers.
- Place FTP and HTTP in a DMZ.
- Run frequent security assessment tools on serves to identify vulnerabilities.

Multiple choice test question:

Ftp uses which port(s)?

a. 22
b. 20 & 21
c. 23

The answer is B. Ftp uses port 20 to send data and port 21 for commands.

Detect # 2: Portscan of port 111

Black Ice defender 2.5

```
1, 2002-03-12 23:51:06, 2003016, RPC TCP port probe, 200.68.32.185, , A.B.C.208, ,
port=111&reason=Firewalled, 2, A, 1166, 111, 0x22e06
```

Snort alert

```
[**] [1:0:0] RPC PORT PROBE [**]
03/12/02-23:50:35.635322 0:10:67:0:4E:5B -> 0:D0:9:EC:E1:6D type:0x800 len:0x4A
200.68.32.185:1166 -> A.B.C.208:111 TCP TTL:46 TOS:0x0 ID:52476 IpLen:20
DgmLen:60 DF
******S* Seq: 0xABF528F2  Ack: 0x0  Win: 0x7D78  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 165095450 0 NOP WS: 0

[**] [1:0:0] RPC PORT PROBE [**]
03/12/02-23:50:44.304324 0:10:67:0:4E:5B -> 0:D0:9:EC:E1:6D type:0x800 len:0x4A
200.68.32.185:1166 -> A.B.C.208:111 TCP TTL:46 TOS:0x0 ID:60184 IpLen:20
DgmLen:60 DF
******S* Seq: 0xABF528F2  Ack: 0x0  Win: 0x7D78  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 165096350 0 NOP WS: 0
```

Source:

This detect came from a colleagues home network. His network consists of a NT 4.0 server, which runs Snort 1.8.3 and a XP home desktop.

Detect was generated by:

The events were generated by snort 1.8.3 for windows with a default rule set (3/09/02) and Black Ice Defender 2.5CAR.

This alert was trigger by the following rule:

```
alert tcp $EXTERNAL_NET any > $HOME_NET 111 (msg:"RPC PORT PROBE";)
```

This rule alerts the system when any external host connects to port 111 using TCP.

The format for the black Ice log is as follow.

```
#Severity          issueId              intruderIp  intruderName victimIp                         count
      timestamp           issueName                                        victimName parameters
      (GMT)
```

Severity 1, This is a number from 1-99 that indicates the severity of an attack, where 1 is not very severe, and 99 is the most severe attack.
Time Stamp 2002-03-12 23:51:06, This indicates the time and date of the **last** time the attack occurred.
Issue ID 2003016, A numeric identifier for this attack type. There are over 300 attack that
IssueName RPC TCP port probe, The name of the attack.
IntruderIP 200.68.32.185, The IP address of the attacker.
Intruder name , The dns or netbois name of the attacker. This only appears if name can be relsoved.
VictimIP A.B.C.208, This is the IP address of who the intruder was attacking.The name of the intruder.
Victim Name , This display your hostname.
Parameters port=111&reason=Firewalled,
This contains some detailed information about the attack.
Count 2, A, 1166, 111, 0x22e06 The number of times this attack was seen.

Probability the source address was spoofed:

The IP address is probably not spoofed. This traffic is a reconnaissance for the portmapper port. The goal of a reconnaissance is to gain information about a host to properly plan your attack, and this cannot be accomplished if the address is forged. Also there were no sign of any other IP address triggering this alert around the time host 200.68.32.185 was scanning. Hiding real source IP address among spoofed addresses is sometimes used to confuse the victim by not being able to identify the real IP address from the forged ones. There is a small possibility that the attacker could have spoof the IP address and sniff replies from network traffic but that would require the attacker to be on the same network.

Description of attack:

This is a port scan of port 111. Port 111 is the well-known port for portmapper. It listens on TCP/UDP port 111 and on some versions of Unix like Solaris 2.x also listen on UDP ports greater than 32770. Portmapper informs a client what port a RPC services is running on after being query. This information is very useful to an attacker because of the many vulnerabilities existing for RPC services. Services like Network File Service (NFS) and Network Information service (NIS) could allow an attack to execute arbitrary commands on a system with permission as root. Portmapper is also vulnerable to Denial of Service attacks as indicated in CVE-1999-0168.

Below are list of a few CVE and CERT advisory

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0189
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-1349
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0195
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0168

CA-99-08 - Buffer Overflow Vulnerability in rpc.cmsd
http://www.cert.org/advisories/CA-99-08-cmsd.html

CA-99-05 - Vulnerability in statd exposes vulnerability in automountd
http://www.cert.org/advisories/CA-99-05-statd-automountd.html

CA-98.11 - Vulnerability in ToolTalk RPC Service
http://www.cert.org/advisories/CA-98.11.tooltalk.html

Attack mechanism:

Is this a stimulus or response?
This traffic is a stimulus because the attacker sent a TCP packet with only the SYN flag set initiating connection.

What services is being targeted?
Portmapper (TCP/UDP port 111) is being targeted. Rpcbind, rpcmountd, rpc-update and rpc.ypupdated could be exploited if running on host.

Does the service have known vulnerabilities or exposures?

Yes there many known vulnerabilities for this service running on various operating systems.

Is this benign, an exploit, a denial of service, or reconnaissance?
This is a reconnaissance.

Correlations:

Examining the logs, we see that this probe was blocked by Black Ice Defender firewall.

1, 2002-03-12 23:51:06, 2003016, RPC TCP port probe, 200.68.32.185, , A.B.C.208, , port=111&reason=**Firewalled**, 2, A, 1166, 111, 0x22e06

Matt Fear posted the following port scan of port 111 below to incident.org.
SANS Global Incident Analysis Center > Current Report

Detect 4: FIN-SYN scan for Sun RPC portmapper. Risk: none

```
Nov 28 01:22:28 [firewall.ip.address] %PIX-6-106015: Deny TCP (no connection) from
  4.41.123.68/111 to dmz.ip.addr.2/111 flags FIN SYN  on interface outside
Nov 28 01:22:28 [firewall.ip.address] %PIX-7-106011: Deny inbound (No xlate) tcp src
  outside:4.41.123.68/111 dst outside:cidr.net.addr.101/111
Nov 28 01:22:28 [firewall.ip.address] %PIX-7-106011: Deny inbound (No xlate) tcp src
  outside:4.41.123.68/111 dst outside:cidr.net.addr.102/111
Nov 28 01:22:28 [firewall.ip.address] %PIX-6-106015: Deny TCP (no connection) from
  4.41.123.68/111 to internal.network.sys.2/111 flags FIN SYN  on interface outside
Nov 28 01:22:28 [firewall.ip.address] %PIX-7-106011: Deny inbound (No xlate) tcp src
  outside:4.41.123.68/111 dst outside:cidr.net.addr.105/111
```

Scott also posted rpc portmapper scan to incidents.org.

```
03:47:34.116255 192.168.42.6.1763 > 10.1.86.0.111: S 467400212:467400212(0) win 32120 <mss
1460,sackOK,timestamp 233549098[|tcp]> (DF) (ttl 47, id 4282)
        4500 003c 10ba 4000 2f06 b889 c0a8 2a06
        1001 5600 06e3 006f 1bdb f614 0000 0000
        a002 7d78 73be 0000 0204 05b4 0402 080a
        0deb ad2a 0000

03:47:34.116596 192.168.42.6.1764 > 10.1.86.1.111: S 463993015:463993015(0) win 32120 <mss
1460,sackOK,timestamp 233549098[|tcp]> (DF) (ttl 47, id 4283)
        4500 003c 10bb 4000 2f06 b887 c0a8 2a06
        1001 5601 06e4 006f 1ba7 f8b7 0000 0000
        a002 7d78 714d 0000 0204 05b4 0402 080a
        0deb ad2a 0000

03:47:34.129896 192.168.42.6.1766 > 10.1.86.3.111: S 469522651:469522651(0) win 32120 <mss
1460,sackOK,timestamp 233549098[|tcp]> (DF) (ttl 47, id 4285)
        4500 003c 10bd 4000 2f06 b883 c0a8 2a06
        1001 5603 06e6 006f 1bfc 58db 0000 0000
```

```
          a002 7d78 10d1 0000 0204 05b4 0402 080a
          0deb ad2a 0000

03:47:34.130398 192.168.42.6.1765 > 10.1.86.2.111: S 462931947:462931947(0) win 32120 <mss
1460,sackOK,timestamp 233549098[|tcp]> (DF) (ttl 47, id 4284)
          4500 003c 10bc 4000 2f06 b885 c0a8 2a06
          1001 5602 06e5 006f 1b97 c7eb 0000 0000
          a002 7d78 a227 0000 0204 05b4 0402 080a
          0deb ad2a 0000

03:47:34.141774 192.168.42.6.1767 > 10.1.86.4.111: S 474133031:474133031(0) win 32120 <mss
1460,sackOK,timestamp 233549098[|tcp]> (DF) (ttl 47, id 4286)
          4500 003c 10be 4000 2f06 b881 c0a8 2a06
          1001 5604 06e7 006f 1c42 b227 0000 0000
          a002 7d78 b73c 0000 0204 05b4 0402 080a
          0deb ad2a 0000

03:47:34.142260 192.168.42.6.1768 > 10.1.86.5.111: S 465718578:465718578(0) win 32120 <mss
1460,sackOK,timestamp 233549098[|tcp]> (DF) (ttl 47, id 4287)
          4500 003c 10bf 4000 2f06 b87f c0a8 2a06
          1001 5605 06e8 006f 1bc2 4d32 0000 0000
          a002 7d78 1cb0 0000 0204 05b4 0402 080a
          0deb ad2a 0000

03:47:34.155054 192.168.42.6.1769 > 10.1.86.6.111: S 468785692:468785692(0) win 32120 <mss
1460,sackOK,timestamp 233549098[|tcp]> (DF) (ttl 47, id 4288)
          4500 003c 10c0 4000 2f06 b87d c0a8 2a06
          1001 5606 06e9 006f 1bf1 1a1c 0000 0000
          a002 7d78 4f95 0000 0204 05b4 0402 080a
          0deb ad2a 0000

03:47:34.156225 192.168.42.6.1770 > 10.1.86.7.111: S 459457511:459457511(0) win 32120 <mss
1460,sackOK,timestamp 233549098[|tcp]> (DF) (ttl 47, id 4289)
          4500 003c 10c1 4000 2f06 b87b c0a8 2a06
          1001 5607 06ea 006f 1b62 c3e7 0000 0000
          a002 7d78 a656 0000 0204 05b4 0402 080a
          0deb ad2a 0000


03:47:34.182231 192.168.42.6.1773 > 10.1.86.10.111: S 470460823:470460823(0) win 32120
<mss 1460,sackOK,timestamp 233549098[|tcp]> (DF) (ttl 47, id 4292)
          4500 003c 10c4 4000 2f06 b875 c0a8 2a06
          1001 560a 06ed 006f 1c0a a997 0000 0000
```

Evidence of active targeting:

This traffic appears to be a scanning on the network segment. The remote host triggered the alert below.

```
[**] [1:0:0] RPC PORT PROBE [**]
03/12/02-23:50:35.635322 0:10:67:0:4E:5B -> 0:D0:9:EC:E1:6D type:0x800 len:0x4A
200.68.32.185:1166 -> A.B.C7.208:111 TCP TTL:46 TOS:0x0 ID:52476 IpLen:20 DgmLen:60
DF
******S* Seq: 0xABF528F2  Ack: 0x0  Win: 0x7D78  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 165095450 0 NOP WS: 0

[**] [1:0:0] RPC PORT PROBE [**]
03/12/02-23:50:35.638190 0:10:67:0:4E:5B -> 0:0:39:61:50:2E type:0x800 len:0x4A
200.68.32.185:1167 -> A.B.C.209:111 TCP TTL:46 TOS:0x0 ID:52477 IpLen:20 DgmLen:60
DF
******S* Seq: 0xAC734793  Ack: 0x0  Win: 0x7D78  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 165095450 0 NOP WS: 0
```

Both network hosts were scanned for port 111.

Severity:

(Critical + Lethal) – (system & network countermeasures)=severity

(2+1)-(5+0)=-2

Critical

2 = The attack is against a personal desktop computer.

Lethality

1 = This attack is very unlikely to succeed because port 111 is not open on this box.

System Countermeasures

5 = This PC is running windows XP home edition and is update with all patches. Also BlackIce defender is installed as a personal firewall.

Network Countermeasures

There are no network defenses in place.

Defensive recommendation:

- Block TCP and UDP port 111 and 32770-32789 at your perimeter defenses
- Apply patch for service to vulnerable systems.

Multiple choice test question:

What part of the three way hand shake does this traffic represents?

```
03/12/02-23:50:35.635322 0:10:67:0:4E:5B -> 0:D0:9:EC:E1:6D type:0x800 len:0x4A
MY.NET.32.185:1166 -> MY.NET.47.208:111 TCP TTL:46 TOS:0x0 ID:52476 IpLen:20
DgmLen:60 DF
******S* Seq: 0xABF528F2  Ack: 0x0  Win: 0x7D78  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 165095450 0 NOP WS: 0
```

a. First
b. Second
c. Thrid

Answer: A. There are three parts to TCP handshake the first send a packet to host with the only the SYN Flags set. The remote host reply with a packet with the SYN and ACK flags set. The host who initiated the connection sends a final packet with the ACK flag set completing the three-way handshake establishing the connection.

Detect 3: WEB-MISC musicat access

Snort Rule

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC musicat access";
flags:A+; uricontent:"/empower"; nocase; classtype:attempted-recon; sid:1221; rev:1;)
```

Snort alert log

```
[**] [1:1221:1] WEB-MISC musicat access [**]
[Classification: Attempted Information Leak] [Priority: 2]
03/22-22:00:42.425305 0:60:8:13:80:C2 -> 0:8:20:CA:60:70 type:0x800 len:0x191
MY.NET.151.11:1254 -> 213.219.48.130:80 TCP TTL:128 TOS:0x0 ID:9239 IpLen:20
DgmLen:387 DF
***AP*** Seq: 0x10526446  Ack: 0x856EF1EC  Win: 0xF950  TcpLen: 20
```

Snort Binary Dump

```
03/22-21:48:10.745222 MY.NET.151.11:1073 -> 213.219.48.130:80
TCP TTL:128 TOS:0x0 ID:3941 IpLen:20 DgmLen:344 DF
***AP*** Seq: 0x495140B  Ack: 0x79315534  Win: 0xFAF0  TcpLen: 20
47 45 54 20 2F 49 63 6F 6E 2F 49 63 6F 6E 73 2F  GET /Icon/Icons/
45 6D 70 6F 77 65 72 6D 65 6E 74 53 6D 61 6C 6C  EmpowermentSmall
2E 67 69 66 20 48 54 54 50 2F 31 2E 31 0D 0A 41  .gif HTTP/1.1..A
63 63 65 70 74 3A 20 2A 2F 2A 0D 0A 52 65 66 65  ccept: */*..Refe
72 65 72 3A 20 68 74 74 70 3A 2F 2F 69 64 73 63  rer: http://idsc
2E 65 6D 6F 6A 6F 2E 63 6F 6D 2F 48 6F 6D 65 2F  .emojo.com/Home/
49 6E 64 65 78 2E 63 66 6D 3F 43 46 49 44 3D 35  Index.cfm?CFID=5
37 39 36 33 31 26 43 46 54 4F 4B 45 4E 3D 33 33  79631&CFTOKEN=33
32 34 34 37 38 35 0D 0A 41 63 63 65 70 74 2D 4C  244785..Accept-L
61 6E 67 75 61 67 65 3A 20 65 6E 2D 75 73 0D 0A  anguage: en-us..
41 63 63 65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A  Accept-Encoding:
20 67 7A 69 70 2C 20 64 65 66 6C 61 74 65 0D 0A   gzip, deflate..
55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69  User-Agent: Mozi
6C 6C 61 2F 34 2E 30 20 28 63 6F 6D 70 61 74 69  lla/4.0 (compati
62 6C 65 3B 20 4D 53 49 45 20 36 2E 30 3B 20 57  ble; MSIE 6.0; W
69 6E 64 6F 77 73 20 4E 54 20 35 2E 31 29 0D 0A  indows NT 5.1)..
48 6F 73 74 3A 20 69 64 73 63 2E 65 6D 6F 6A 6F  Host: idsc.emojo
2E 63 6F 6D 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E  .com..Connection
3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 0D 0A  : Keep-Alive....

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

03/22-22:00:42.425305 MY.NET.151.11:1254 -> 213.219.48.130:80
TCP TTL:128 TOS:0x0 ID:9239 IpLen:20 DgmLen:387 DF
***AP*** Seq: 0x10526446  Ack: 0x856EF1EC  Win: 0xF950  TcpLen: 20
47 45 54 20 2F 49 63 6F 6E 2F 49 63 6F 6E 73 2F  GET /Icon/Icons/
```

```
45 6D 70 6F 77 65 72 6D 65 6E 74 53 6D 61 6C 6C  EmpowermentSmall
2E 67 69 66 20 48 54 54 50 2F 31 2E 31 0D 0A 41  .gif HTTP/1.1..A
63 63 65 70 74 3A 20 2A 2F 2A 0D 0A 52 65 66 65  ccept: */*..Refe
72 65 72 3A 20 68 74 74 70 3A 2F 2F 69 64 73 63  rer: http://idsc
2E 65 6D 6F 6A 6F 2E 63 6F 6D 2F 0D 0A 41 63 63  .emojo.com/..Acc
65 70 74 2D 4C 61 6E 67 75 61 67 65 3A 20 65 6E  ept-Language: en
2D 75 73 0D 0A 41 63 63 65 70 74 2D 45 6E 63 6F  -us..Accept-Enco
64 69 6E 67 3A 20 67 7A 69 70 2C 20 64 65 66 6C  ding: gzip, defl
61 74 65 0D 0A 49 66 2D 4D 6F 64 69 66 69 65 64  ate..If-Modified
2D 53 69 6E 63 65 3A 20 4D 6F 6E 2C 20 31 32 20  -Since: Mon, 12
4D 61 72 20 32 30 30 31 20 31 35 3A 30 30 3A 33  Mar 2001 15:00:3
30 20 47 4D 54 0D 0A 49 66 2D 4E 6F 6E 65 2D 4D  0 GMT..If-None-M
61 74 63 68 3A 20 22 30 66 62 65 62 32 64 35 61  atch: "0fbeb2d5a
62 63 30 31 3A 61 35 64 22 0D 0A 55 73 65 72 2D  bc01:a5d"..User-
41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 34  Agent: Mozilla/4
2E 30 20 28 63 6F 6D 70 61 74 69 62 6C 65 3B 20  .0 (compatible;
4D 53 49 45 20 36 2E 30 3B 20 57 69 6E 64 6F 77  MSIE 6.0; Window
73 20 4E 54 20 35 2E 31 29 0D 0A 48 6F 73 74 3A  s NT 5.1)..Host:
20 69 64 73 63 2E 65 6D 6F 6A 6F 2E 63 6F 6D 0D   idsc.emojo.com.
0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65  .Connection: Kee
70 2D 41 6C 69 76 65 0D 0A 0D 0A                 p-Alive....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+=+
```

Source of trace:

This traffic was obtained from my cable modem. There is just one machine that runs windows XP professional and Snort.

Detect was generated by:

This detect was generated by Snort version 1.8.3 for windows with a rule set of 3/15/02.
The rule that trigger this alert is listed below:

```
 Rule Header                                              Rule Option
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC musicat
access"; flags:A+; uricontent:"/empower"; nocase; classtype:attempted-recon; sid:1221;
rev:1;)
```

The first part of the rule, Rule Header, alerts the system of any external host connecting to port 80 using TCP. The second part of the rule, Rule Options, inspects the packet for the ACK and any other flags that are set. It also searches the payload for the character string "/empower". This search for the characters is not case sensitive because of the nocase parameter.

Probability the source address was spoofed:

The probability the address is spoofed is low because of the completion of the TCP three way-handshake.

Description of attack:

This attack exploits Muscat Empower. Muscat Empower is a software product that provides knowledge management capability to Internet, Intranet and eCommerce applications. This exploit works by either sending the Muscat Empower product an invalid or non-existing database request, which will display the directory that the CGI program and database reside in.

CVE CAN-2001-0224

BUGTRAQ:20010212 Vulnerability in Muscat Empower which can print path to DB-dir.
URL:http://archives.neohapsis.com/archives/bugtraq/2001-02/0216.html
BID:2374
URL:http://www.securityfocus.com/bid/2374

Attack mechanism:

Is this a stimulus or response?

This traffic is a stimulus. This traffic originated from my network as web surfing traffic. The port scan log confirms that the traffic initiated from my network.

Portlog snip

```
Mar 22 22:00:40 MY.NET.151.11:1254 -> 213.219.48.130:80 SYN
******S*
```

What service is being targeted?

Muscat

Does the service have known vulnerabilities or exposures?
Yes, this service is vulnerable to a request with an invalid or non-existing database is sent to the Muscat Empower product, the product will return the path that the CGI and the Database reside in.

Is this benign, an exploit, denial of service, or reconnaissance?

This is traffic is benign. This traffic is a false positive caused by misconfiguration in Snort. An explanation for this conclusion is in the correlation section.

Correlation:

Examining the logs, particularly; the Snort binary dump, shows why the alert was triggered. The rule meets all of the requirements. The Rule Header was configured to alert any external host connecting to port 80 on web server. But what I found interesting was my address appeared as the external address and the web server. This was caused by a misconfiguration in my snort.conf file for the Home_NET, HTTP and External_NET variables shown below.

Snort.conf snip

```
# or you can specify the variable to be any IP address
# like this:

var HOME_NET any

# Set up the external network addresses as well.
# A good start may be "any"

var EXTERNAL_NET any

# Set up your web servers, or simply configure them
# to HOME_NET

var HTTP_SERVERS $HOME_NET
```

Also when snort searched the payload it found the characters empower which is part of empowermentSmall.gif. This was not an attack but a false positive.

```
03/22-22:00:42.425305 MY.NET.151.11:1254 -> 213.219.48.130:80
TCP TTL:128 TOS:0x0 ID:9239 IpLen:20 DgmLen:387 DF
***AP*** Seq: 0x10526446  Ack: 0x856EF1EC  Win: 0xF950  TcpLen: 20
47 45 54 20 2F 49 63 6F 6E 2F 49 63 6F 6E 73 2F  GET /Icon/Icons/
45 6D 70 6F 77 65 72 6D 65 6E 74 53 6D 61 6C 6C  EmpowermentSmall
2E 67 69 66 20 48 54 54 50 2F 31 2E 31 0D 0A 41  .gif HTTP/1.1..A
63 63 65 70 74 3A 20 2A 2F 2A 0D 0A 52 65 66 65  ccept: */*..Refe
72 65 72 3A 20 68 74 74 70 3A 2F 2F 69 64 73 63  rer: http://idsc
2E 65 6D 6F 6A 6F 2E 63 6F 6D 2F 0D 0A 41 63 63  .emojo.com/..Acc
65 70 74 2D 4C 61 6E 67 75 61 67 65 3A 20 65 6E  ept-Language: en
2D 75 73 0D 0A 41 63 63 65 70 74 2D 45 6E 63 6F  -us..Accept-Enco
64 69 6E 67 3A 20 67 7A 69 70 2C 20 64 65 66 6C  ding: gzip, defl
61 74 65 0D 0A 49 66 2D 4D 6F 64 69 66 69 65 64  ate..If-Modified
```

```
2D 53 69 6E 63 65 3A 20 4D 6F 6E 2C 20 31 32 20  -Since: Mon, 12
4D 61 72 20 32 30 30 31 20 31 35 3A 30 30 3A 33  Mar 2001 15:00:3
30 20 47 4D 54 0D 0A 49 66 2D 4E 6F 6E 65 2D 4D  0 GMT..If-None-M
61 74 63 68 3A 20 22 30 66 62 65 62 32 64 35 61  atch: "0fbeb2d5a
62 63 30 31 3A 61 35 64 22 0D 0A 55 73 65 72 2D  bc01:a5d"..User-
41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 34  Agent: Mozilla/4
2E 30 20 28 63 6F 6D 70 61 74 69 62 6C 65 3B 20  .0 (compatible;
4D 53 49 45 20 36 2E 30 3B 20 57 69 6E 64 6F 77  MSIE 6.0; Window
73 20 4E 54 20 35 2E 31 29 0D 0A 48 6F 73 74 3A  s NT 5.1)..Host:
20 69 64 73 63 2E 65 6D 6F 6A 6F 2E 63 6F 6D 0D   idsc.emojo.com.
0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65  .Connection: Kee
70 2D 41 6C 69 76 65 0D 0A 0D 0A                p-Alive....
```

```
   Rule Header                    Rule Option
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"WEB-MISC musicat
access"; flags:A+; uricontent:"/empower"; nocase; classtype:attempted-recon; sid:1221;
rev:1;)
```

Evidence of active targeting:

There is no evidence of a target attack. This alert is a false postive.

Severity:

(Critical + Lethal) – (system & Network countermeasures)=severity

(2+1)-(5+0)=2

Critical

2 This attack is against a desktop computer

Lethality

1 This attack is very unlikely to succeed because this exploit affects hosts running Muscat, which this host does not.

System Countermeasures

5 This system is running XP professional edition with all the updates installed and the current version of BlackICE Defender.

Network Countermeasure

There are no network defenses.

Defensive recommendation:

If this were a server running the Muscat service I would recommend applying patch to server.

My Recommendation for fixing the false positive is as follow:

- Remark line to setup HTTP_Server
- Disable snort rule that apply to web server

Multiple choice test questions:

What port does Muscat run on?
A) HTTP
B) FTP
C) SSH

Answer is HTTP

Detect #4: Sub Seven

snort:

```
Jan 29 17:34:16 greatwall snort: [1:0:0] TCP to 27374 SubSeven {TCP}
211.169.151.105:2334 -&gt; 12.82.140.157:27374
Jan 29 17:34:37 greatwall last message repeated 3 times
```

ipchains:

```
Jan 29 17:34:16 greatwall kernel: Packet log: input DENY ppp0 PROTO=6
211.169.151.105:2334 12.82.140.157:27374
L=48 S=0x00 I=52423 F=0x4000 T=107 SYN (#64)
Jan 29 17:34:19 greatwall kernel: Packet log: input DENY ppp0 PROTO=6
211.169.151.105:2334 12.82.140.157:27374</p>A
L=48 S=0x00 I=59591 F=0x4000 T=107 SYN (#64)</p>
Jan 29 17:34:25 greatwall kernel: Packet log: input DENY ppp0 PROTO=6
211.169.151.105:2334 12.82.140.157:27374
L=48 S=0x00 I=9672 F=0x4000 T=107 SYN (#64)
Jan 29 17:34:37 greatwall kernel: Packet log: input DENY ppp0 PROTO=6
211.169.151.105:2334 12.82.140.157:27374
L=48 S=0x00 I=44744 F=0x4000 T=107 SYN (#64)
```

p0f: www.stearns.org/p0f/

```
&lt;Tue Jan 29 17:34:16 2002&gt; 211.169.151.105 [22 hops]: Windows 9x (1)
+ 211.169.151.105:2334 -&gt; 12.82.140.157:27374
&lt;Tue Jan 29 17:34:19 2002&gt; 211.169.151.105 [22 hops]: Windows 9x (1)
```

```
+ 211.169.151.105:2334 -&gt; 12.82.140.157:27374
&lt;Tue Jan 29 17:34:25 2002&gt; 211.169.151.105 [22 hops]: Windows 9x (1)
+ 211.169.151.105:2334 -&gt; 12.82.140.157:27374
&lt;Tue Jan 29 17:34:37 2002&gt; 211.169.151.105 [22 hops]: Windows 9x 1)
+ 211.169.151.105:2334 -&gt; 12.82.140.157:27374
```

Source:

This detect was posted to incidents.org on Jan 29,2002 by John Sage.
http://www.incidents.org/archives/intrusions/msg02988.html

Detect was generated by:

This detect was generated by Snort and ipchains, a Linux based firewall. The following rule probably trigger this alert:

```
Alert tcp $EXTERNAL_NET 27374 -> $HOME_NET any (msg:"BACKDOOR subseven
22"; flags: A+; content: "|0d0a5b52504c5d3030320d0a|"; reference:arachnids,485;
reference:url,www.hackfix.org/subseven/; sid:103; classtype:misc-activity; rev:4;)
```

Probability the source address was spoofed:

The probability of the address being spoofed is low. The attacker is trolling for Subseven hosts and this scan would be easier if the IP was not spoof. In order for this scan to work with a spoofed address, the attacker would have to be on the same network and to monitor replies with a sniffer.

Description of attack:

Subseven is a Trojan that allows an attacker to control your computer remotely. Once an attacker has control of your machine they have access to the registry and file system where they could read/modify files. Subseven could be configured to obtain information from a victim by enabling the key logger option which captures keystrokes which would give your username, password or various account information.

Attack mechanism:

Is this a stimulus or response?
This is stimulus because it was initiated by host 211.169.151.105.

What service is being targeted?

This is a scan to port 27374, which is the default port for the Subseven Trojan. Subseven is also found on the following:

Port	
1234 TCP	Sub Seven
6667 TCP	Sub Seven (new icq notification)
6711 TCP	Sub Seven
6712 TCP	Sub Seven
6713 TCP	Sub Seven
6776 TCP	Sub Seven
16959 TCP	Sub Seven DEFCON8 2.1 Backdoor
27374 UDP	Sub 7 2.1
27573 TCP/UDP	Sub 7 2.1

Does the services have known vulnerabilities or exposures?

Is this benign, an exploit, a denial of service, or reconnaissance?

This is an exploit

Sub 7 is sent to a victim as an attachment. The email may claim to be from a well-known company like Microsoft claiming the attachment will protect them from a virus. This was the case with Sub 7 version 2.0 which was sent via email from a Japanese Hotmail account claiming to be from Microsoft Japan service as stated in Symantec.com virus alert. The email attempts to have the user run the attachment that they claimed will protect the machine from the pinkworm virus. Attackers also may try to trick their victims in to installing the Sub 7 trojan by sending the file claiming it a special program or picture file via ICQ. After the software is installed it can notify the attacker of the infected victim.

Correlations:

This attack was unsuccessful because of the rule in the firewall that drops connection to port 27374. I came to this conclusion from the firewall log.

```
Jan 29 17:34:16 greatwall kernel: Packet log: input DENY ppp0 PROTO=6
211.169.151.105:2334 12.82.140.157:27374
```

James C. Slora Jr experience Sub 7 probing on Feb 25, 2002 and posted it on www.incidents.org.

```
2002-02-22     00:55:33       66.108.130.228 xx.xx.xx.xx    Tcp
       1982    27374   SYN
Header: 45 00 00 30 b0 a6 40 00 70 06 4c 73 42 6c 82 e4 xx xx xx xx
Data: 07 be 6a ee 01 cd f7 66 00 00 00 00 70 02 20 00 e9 90 00 00 02
```

```
04 05
b4 01 01 04 02

2002-02-22     00:55:36       66.108.130.228 xx.xx.xx.xx    Tcp
        1982    27374   SYN
Header: 45 00 00 30 bc a6 40 00 70 06 40 73 42 6c 82 e4 xx xx xx xx
Data: 07 be 6a ee 01 cd f7 66 00 00 00 00 70 02 20 00 e9 90 00 00 02
04 05
b4 01 01 04 02

2002-02-22     00:55:42       66.108.130.228 xx.xx.xx.xx    Tcp
        1982    27374   SYN
Header: 45 00 00 30 c9 a6 40 00 70 06 33 73 42 6c 82 e4 xx xx xx xx
Data: 07 be 6a ee 01 cd f7 66 00 00 00 00 70 02 20 00 e9 90 00 00 02
04 05
b4 01 01 04 02

2002-02-22     00:55:54       66.108.130.228 xx.xx.xx.xx    Tcp
        1982    27374   SYN
Header: 45 00 00 30 2d a7 40 00 70 06 cf 72 42 6c 82 e4 xx xx xx xx
Data: 07 be 6a ee 01 cd f7 66 00 00 00 00 70 02 20 00 e9 90 00 00 02
04 05
b4 01 01 04 02
```

Evidence of active targeting:

This is a scan for host infected with the sub seven Trojan.

Severtiy

(2+5) – (0+2)=5

Criticality:

2 = This is a Windows 9X computer.

Lethality:

5 = If this attack would succeed the attacker would gain full control of host.

Sys Countermeasures
0 = I will assume that there system countermeasures.

Network Countermeasure
2 = This site has firewall.

Defense Recommendation

- Setup proper ACL on perimeter devices

- Install an Anti Virus program on clients PCs
- Block email attachment with the exe extension

Question

Subseven is

a. Trojan.
b. Denial of Services attack
c. A Peer to peep program.

Answer is A. Subseven is a Trojan that allows an attacker to remote control a host.

Detect 5: UPNP Malformed advertisement

```
[**] [1:1384:2] MISC UPNP malformed advertisement [**]
[Classification: Misc Attack] [Priority: 2]
03/18/02-03:59:01.157091 0:10:67:0:4E:5B -> 0:D0:9:EC:E1:6D type:0x800 len:0x11D
215.200.155.50:3753 ->MY.NET.47.208:1900 UDP TTL:49 TOS:0x0 ID:10766 IpLen:20
DgmLen:271
Len: 251

[**] [1:1384:2] MISC UPNP malformed advertisement [**]
[Classification: Misc Attack] [Priority: 2]
03/18/02-04:06:42.605946 0:10:67:0:4E:5B -> 0:D0:9:EC:E1:6D type:0x800 len:0x11D
215.200.155.50:3754 -> MY.NET.47.208:1900 UDP TTL:49 TOS:0x0 ID:11905 IpLen:20
DgmLen:271
Len: 251

 [**] [1:1384:2] MISC UPNP malformed advertisement [**]
[Classification: Misc Attack] [Priority: 2]
03/18/02-04:07:05.724679 0:10:67:0:4E:5B -> 0:D0:9:EC:E1:6D type:0x800 len:0x11D
215.200.155.50:3755 ->MY.NET.47.208:1900 UDP TTL:49 TOS:0x0 ID:11975 IpLen:20
DgmLen:271
Len: 251

[**] [1:1384:2] MISC UPNP malformed advertisement [**]
[Classification: Misc Attack] [Priority: 2]
03/18/02-04:07:27.976395 0:10:67:0:4E:5B -> 0:D0:9:EC:E1:6D type:0x800 len:0x11D
215.200.155.50:3756 -> MY.NET.47.208:1900 UDP TTL:49 TOS:0x0 ID:12009 IpLen:20
DgmLen:271
Len: 251
```

Source of trace:

This trace came from my cable connection.

Detect was generated by:

Snort 1.8.3 running on Windows XP professional. The rule that triggered this alert is listed below.

```
alert udp $EXTERNAL_NET any -> $HOME_NET 1900 (msg:"MISC UPNP
malformed advertisement"; content:"NOTIFY * "; nocase; classtype:misc-attack;
reference:cve,CAN-2001-0876; reference:cve,CAN-2001-0877; sid:1384; rev:2;)
```

Probability the source is spoofed:

The probability of the address being spoofed is high since UDP is being used and this attack tries to could cause a buffer overflow or DoS.

Description of attack:

This attack exploits the SSDP discovery service, which allows PCs to discover Universal Plug and Play device on the network.

CVE: CAN-2001-0876
CVE- CVE-2001-0877

Attack Mechanism:

Is this a stimulus or response?

This traffic is most likely a stimulus but it hard to tell since it UDP traffic.

What services is being targeted?

The SSDP discovery service.

Does the service have known vulnerabilities or exposures?

They are currently two ways that this service can be exploited. One method is to specify a server as the host that handles device description in the NOTIFY directive and if enough computers response this would cause a flood of request to server causing a disturbed Denial of service. The second method is to cause a buffer overflow, which could allow an attack access to system in the context of the SSDP service that runs with System privileges.

Is this benign, an exploit, denial of service, or reconnaissance?

This is an exploit of the SSDP service.

Correlations:

James Edwards saw this traffic on 12/21/2001 and posted it on incidents.org
http://www.incidents.org/archives/intrusions/msg02420.html

Steve Wray also saw this traffic and posted it on incidents.org on 12/23/01.
http://www.incidents.org/archives/intrusions/msg02426.html

Evidence of active targeting:

This is an attack targeting this system to cause a DOS. Most likely there was some kind reconnaissance done to know that this system was listening on port 1900. The firewall log does not show any scans to port 1900 but there are scan to port 5000 which is port associated with this attack.

Severity:

(Criticality + Lethality)-(system Countermeasures + network countermeasures)=Severity
(2+4)-(5+0)=1

Criticality
2 = Windows XP professional PC
Lethality
4 = An attacker could cause an DOS or gain access to Host
System countermeasure
5 = System is up to date with all patches and runs BlackIce Defender
Network countermeasure
0 There are no network defensive in place

Defensive Recommendation:

- Apply patch to systems that the run SSDP service.
- Disable service if it is not needed

Multiple-choice question

```
215.200.155.50:3756 -> MY.NET.47.208:1900 UDP TTL:49 TOS:0x0 ID:12009 IpLen:20
DgmLen:271
Len: 251
```

What port does SSDP run on?

a. 3756
b. 1900
c. 3201

Answer b SSDP runs on port 1900.

Assignment 3- Analyze This

Overview:

The purpose of the analysis of the logs collected from March 25 – 29 will attempt to identify compromised systems, network configuration problems and/or malicious activity within the university network.

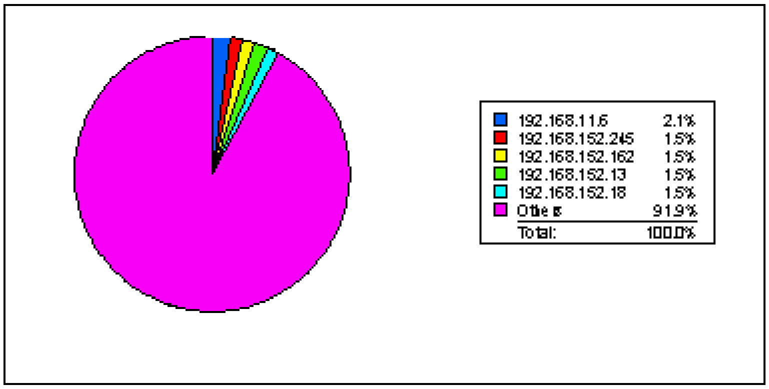
My analysis was on the following files:

Alerts	OOS	Scans
Alert.020325.txt	Oos_Mar.25.2002	Scans.020325.txt
Alert.020326.txt	Oos_Mar.26.2002	Scans.020326.txt
Alert.020327.txt	Oos_Mar.27.2002	Scans.020327.txt
Alert.020328.txt	Oos_mar.28.2002	Scans.020328.txt
Alert.020329.txt	Oos_mar.29.2002	Scans.020329.txt

Each of the files was opened with Microsoft Word were I replaced MY.NET with 192.168, to help in the parsing the files. All alerts were process with SnortSnarf. Scan files were imported into Microsoft Access, and then were queried by Crystal Reports. The "Oos" files were imported into Microsoft Excel and were analyzed manually.

This analysis required the analyst to parse through and identify malicious traffic in 216mb of data consisting of 279,740 network alerts. There were 1,993,603 scans on the network. The majority of the scans, originated from within the MY.NET network, and destined to other hosts in the network. The top 5 source and target hosts are shown in the charts below:

Top 5 Targets



Note: Others consist of hosts with less than 1.5% of activity.

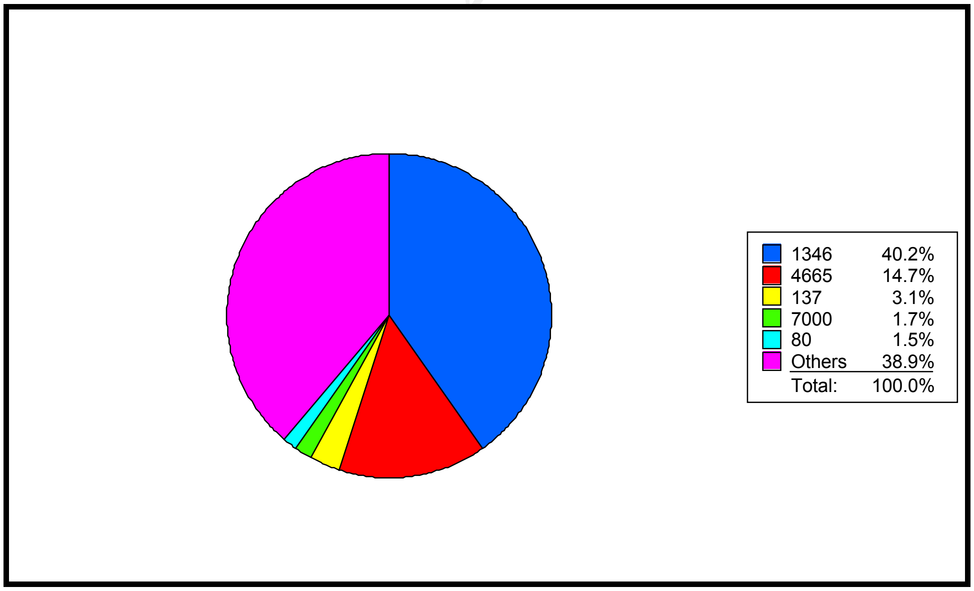
Top 5 Source IP Addresses

Source IP	# of Alerts triggered
MY.NET.153.197	21640
MY.NET.70.177	20292
MY.NET.11.8	15833
MY.NET.152.13	10805
MY.NET.11.7	8351

Top Destination Ports

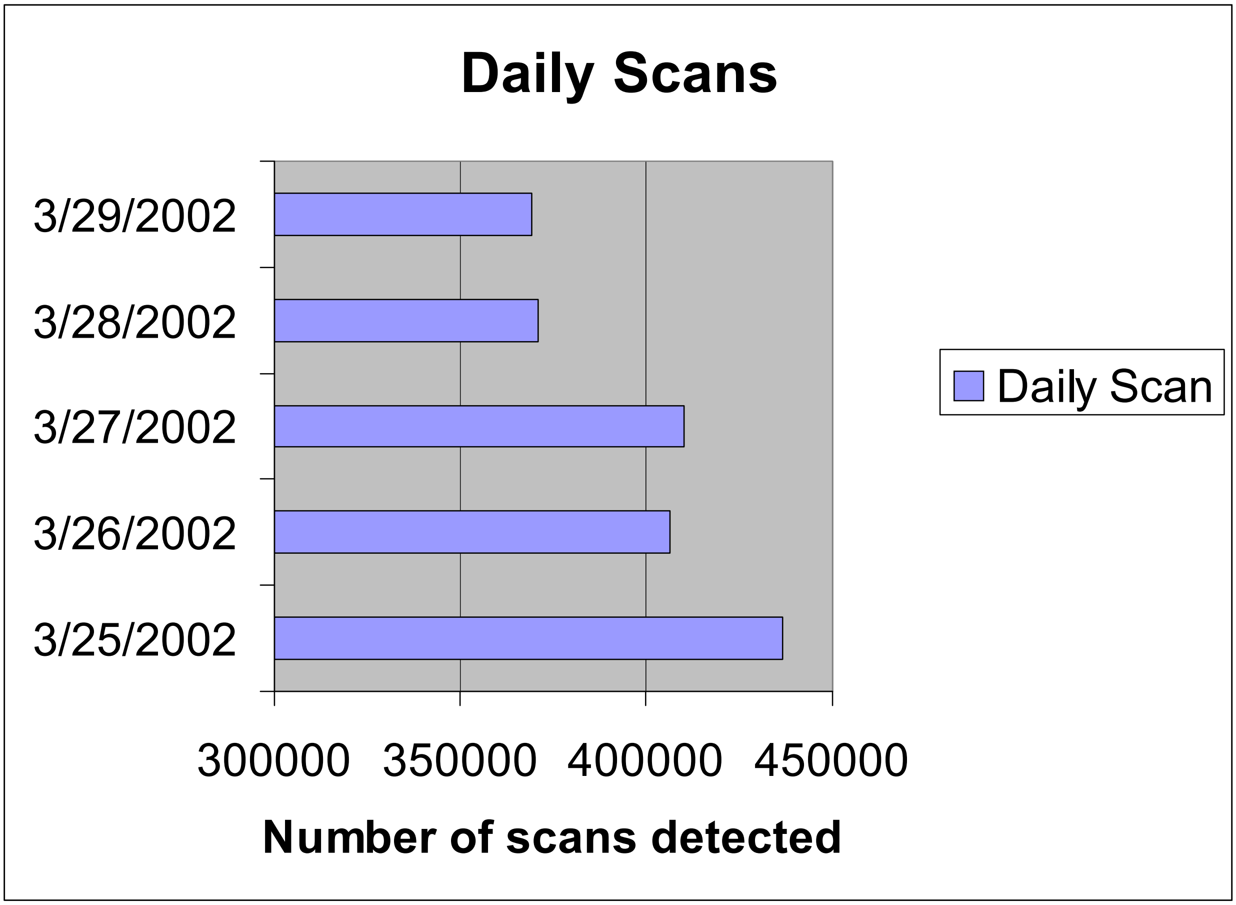
During the analysis phase the top destination ports were identified and separated from the traffic. This can be a form of statistical analysis and possibly identify malicious traffic to a specific port. In addition, this would allow the analyst to identify the types of traffic normally occurring on this network. When armed with this information the analyst will be better able to block specific ports on a firewall or ACL when one is implemented or planned. Last, when this traffic is identified this will better establish baseline information for the analyst to better identify malicious traffic especially if it occurs on a large-scale basis. The graph below identifies the Top 5 destination Ports.

Top 5 destination ports



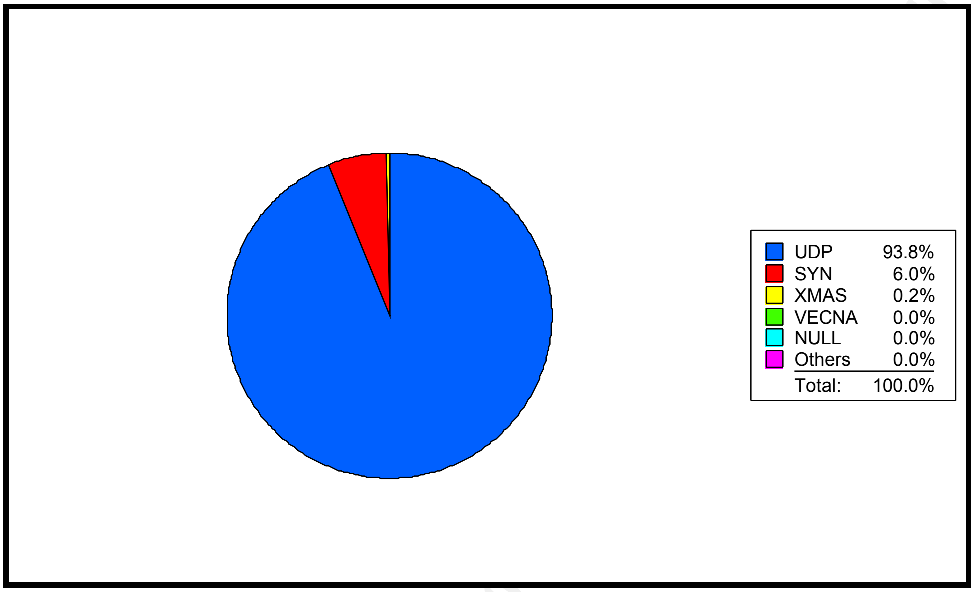
Daily Scans

Analysis of daily scan numbers can assist the analyst in identifying a jump in traffic from day to day. This can identify malicious activity that occurs on large scale. This method should not be considered reliable for identifying specific systems that have been compromised as this scan is normally used on a high level rather than “down in the trenches”. As indicated in the graph below you can see there was a huge decrease in traffic from 3/25/2002 to 3/29/2002. This can be signs of network problems that occurred, or something less malicious like students or faculty going home early for the weekend or taking extended weekend. As seen in the graph below there were approximately 1 million scans that were accounted for.



Top Protocols Used

Identified in the graph below is the top protocols that were used on this network. The majority of network activity was UDP traffic with **275,437** occurrences. The majority of UDP Traffic that triggered alerts was the MISC Large UDP Packet rule, the High port 65535 UDP – possible Red Worm – traffic rule, and TFTP - External UDP connection to internal TFTP server rule. These rules all should trigger a response from the analyst as this traffic appears at first glance highly suspicious.



Alert Activity

The types and classes of alerts triggered on this network tend to identify that this university is a heterogeneous network consisting of windows and UNIX hosts. The following is a description for each of the top ten alerts by occurrence.

Top 10 Alerts by Occurrences

In order to better classify and identify some of the traffic we have broken it down into the top 10 alerts and based this on number of occurrences. This allows the analyst to better see what is causing the majority of traffic on the network and whether those alerts are suspicious or benign.

Alert Description	Occurrences
connect to 515 from inside	56202
spp_http_decode: IIS Unicode attack detected	53950
SMB Name Wildcard	51480
SNMP public access	40069
ICMP Echo Request L3retriever Ping	25531
MISC Large UDP Packet	21982

INFO MSN IM Chat data	7689
ICMP Echo Request Nmap or HPING2	3875
INFO Inbound GNUTella Connect request	2822
Watchlist 000220 IL-ISDNNET-990517	2237

Connect to 515 from inside

Connections directed to TCP port 515 is the LPR service. As specified from the rule it appears the connections are from inside the network. This traffic can be classified as benign since connections to print servers are very common on internal networks. If the connections were being derived from external a cause for concern should be noted as TCP port 515 does have vulnerabilities on certain operating systems. Such as in the case of BSD and Linux distributions where attackers can exploit the syslog() function. This exploit if successful, would allow an attacker root access to both a local and remote host. Additionally there is an exploit affecting Windows NT 4.0 and 2000 TCP/IP Print service. An attacker could cause a DOS by sending a malformed packet to port 515, resulting in an interruption in the service and other services like, FTPSvc, and DHCPServer that are running on the host.

Top Ten Talkers for this Alert:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.203	5994	6154	1	15
MY.NET.153.125	5358	6950	1	29
MY.NET.153.119	4890	4893	1	2
MY.NET.153.118	4283	4283	1	1
MY.NET.153.109	3560	3944	1	19
MY.NET.153.120	3419	3532	1	14
MY.NET.153.115	2115	5619	1	60
MY.NET.153.108	1885	1887	1	2
MY.NET.153.113	1815	2200	1	7
MY.NET.153.148	1721	1980	1	21

Correlation

The following analyst listed below experienced the similar traffic:

Roland Gerlach http://www.sans.org/y2k/practical/Roland_Gerlach_GCIA.html#assign3

Lorraine Weaver http://www.giac.org/practical/Lorraine_Weaver_GCIA.zip

Defense Recommendation

- Patch any systems running vulnerable versions of LPRng version.
- Block access to traffic coming in and going out of your company at perimeter defenses.

IIS Unicode attack

The IIS Unicode attack, also known as the Web Server Folder traversal, allows an attacker access to view files that are outside of the web root. This is accomplished by sending an abnormal URL request like http://victim/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir to an IIS 4.0 or 5.0 server. When IIS processes the request, it is done inaccurately resulting in access to the file system in the context of the configured service account.

During the period of March 25 – 29, the network encountered a considerable amount of IIS Unicode scans.

This alert some times generates false positive when visited by sites that use multi-btye characters such Simple Chinese characters as stated in John Berkers response on Neohapsis.com message board. Four external hosts connected from the Asia-Pacific region from China, Korea, and Thailand.

Here is a list of IP's from the Asia-Pacific region that triggered alert:

IP	Country
61.186.37.39	China
202.44.10.80	Thailand
203.229.99.61	Korea
61.186.37.70	China

Some of the internal hosts that were targets in this alert caused some concerns because of other malicious web alerts that were triggered on them. It could be cause by an external host attempting to exploit one of the many IIS vulnerabilities. These host listed should be scanned with Microsoft Hfnetchk, a hot fix checker tool, to determine if all patches on hosts are current.

MY.NET.5.79	MY.NET.150.195	MY.NET.150.226	MY.NET.2.244	MY.NET.5.96
MY.NET.150.83	MY.NET.5.95	MY.NET.150.16	MY.NET.150.147	MY.NET.150.139
MY.NET.153.219	MY.NET.152.220	MY.NET.150.107	MY.NET.150.133	MY.NET.150.197

Reference:

John Berkers messages posted on neohapsis message board.
http://archives.neohapsis.com/archives/snort/20001-08/0075.html

The following is a list of the top ten talkers of this alert:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.153.197	20657	21640	88	97
MY.NET.153.115	3504	5619	59	60
MY.NET.153.171	2552	2679	18	19
MY.NET.153.124	2547	3046	51	52
MY.NET.152.19	2525	3657	20	24
MY.NET.153.168	1706	1726	24	27
MY.NET.153.162	1659	2394	45	50
MY.NET.153.125	1590	6950	28	29
MY.NET.153.190	1424	1545	25	27
MY.NET.153.154	1379	1563	43	46

Correlation

Keven Murphy saw this alert in his analysis.

Recommendation

- Apply the following patch to associated IIS version running

  Microsoft IIS 4.0 alpha:

  Microsoft Patch Q269862
  http://download.microsoft.com/download/winntsp/Patch/q269862/NT4ALPHA/EN-US/prmcan4a.exe

  Microsoft Patch Q269862
  http://download.microsoft.com/download/winntsp/Patch/q269862/NT4ALPHA/EN-US/prmcan4as.exe

  Microsoft IIS 4.0:

  Microsoft Patch Q269862
  http://download.microsoft.com/download/winntsp/Patch/q269862/NT4ALPHA/EN-US/prmcan4i.exe

  Microsoft Patch Q269862
  http://download.microsoft.com/download/winntsp/Patch/q269862/NT4ALPHA/EN-US/prmcan4is.exe

  Microsoft Personal Web Server 4.0:

  David Raitzer Patch pws_patch.zip
  http://www.geocities.com/p_w_server/pws_patch/index.htm

Microsoft IIS 5.0:

Microsoft Patch Q269862
http://download.microsoft.com/download/win2000platform/Patch/q269862/NT5/EN-US/Q269862_W2K_SP2_x86_en.EXE

SMB Name Wildcard

The SMB Name Wildcard alert is triggered when a host sends a packet from port 137 to port 137 on a remote host with certain characters string in the payload. An attacker attempting this exploit is trying to gain information on NetBIOS resources to launch future attacks. This NETBIOS traffic is common when there are windows PCs or Samba servers on a network, which I believe is the case with machines involved in network activity to hosts MY.NET.11.6 and MY.NET.11.7 who were the two top source and destination IP in SnortSnarf that triggered this alert. 51480 alerts were generated for this traffic out of which MY.NET.11.6 triggered 15833 alerts and MY.NET.11.7 triggered 8351 alerts. The traffic seen below illustrates NetBIOS Name service activities that are used for logon sequence, Windows NT 4.0 trusts, Windows NT 4.0 secure channel, pass through validation, browsing, and printing in a Windows environment.

```
Mar 25 00:06:28 MY.NET.152.21:137 -> MY.NET.11.6:137 UDP
```

The traffic below followed the traffic above in the scan logs. This represent a connection to some kind of NetBIOS session like File Sharing, Printing, Logon Sequence, Windows NT 4.0 Trusts, Windows NT 4.0 Directory Replication, Windows NT 4.0 Secure Channel, Pass Through Validation, and Windows NT 4.0 Administration Tools.

```
Mar 25 00:06:28 MY.NET.152.21:2259 -> MY.NET.11.6:139 SYN ******S*
```

Also there were 60 instances that this alert was triggered by an external IP, host 24.188.117.164. This host caused concern because this could indicate a script-kiddy attempting to gain information about a host using NBTSTAT or network.vbs worm as stated in Bryce Alexander's intrusion detection FAQ on port 137 scan.

This host also caused concerns by the alerts it triggered.

4 different signatures are present for *24.188.117.164* as a source

- 4 instances of *ICMP Echo Request Windows*
- 39 instances of *ICMP Echo Request L3retriever Ping*
- 43 instances of *suspicious host traffic*
- 60 instances of *SMB Name Wildcard*

Reference:

Bryce Alexander’s intrusion detection FAQ

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

The top talker were:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.11.6	15833	15833	50	50
MY.NET.11.7	8351	8351	45	45
MY.NET.152.160	867	1792	3	3
MY.NET.11.5	707	707	55	55
MY.NET.152.161	553	1182	2	3
MY.NET.152.15	548	1193	3	8
MY.NET.152.21	545	1508	3	175
MY.NET.152.171	540	1260	3	7
MY.NET.152.251	540	1174	3	10
MY.NET.152.19	525	3657	2	24

Correlation

The following analysts also saw SMB wild card traffic in their analysis:

Lorraine Weaver http://www.giac.org/practical/Lorraine_Weaver_GCIA.zip

Mark Evans http://www.giac.org/practical/Mark_Evans_GCIA.zip

David Singer http://www.giac.org/practical/David_Singer_GIAC.doc

Byrce Alexander also experienced the same traffic which is listed below, additionally he wrote about the same traffic in FAQ IDS question.

```
Apr 21 00:17:29 myhost snort: SMB Name Wildcard: 192.168.0.1:137 -> my.ip.addr:137
Apr 21 00:17:29 myhost snort: SMB Name Wildcard: 24.28.135.131:137 -> my.ip.addr:137
Apr 21 00:17:31 myhost snort: SMB Name Wildcard: 24.28.135.131:137 -> my.ip.addr:137
Apr 21 00:17:31 myhost snort: SMB Name Wildcard: 192.168.0.1:137 -> my.ip.addr:137
```

Defense Recommendation

- Disable file sharing if it is not needed
- Deploy a corporate personal firewall that can block access to unauthorized hosts and can establish a trust with authorized hosts.
- Block access at perimeter defense for inbound and outbound traffic.
- Scan hosts with anti-virus software with current scan files for virus.

SNMP public access

The SNMP Public access alert is triggered by an attempt to use an SNMP community string name of public or private to access systems and/or change system information. The default community strings are public for read access and private for write access. It is recommended that you change the community string, which would prevent attacks that may exploit the use of these default community strings. This traffic was not sent from any external machines, all source IP's triggering this alert originated from within the MY.NET network and were destined to other hosts within the network. This traffic could be normal but there is a concern with recent vulnerabilities in SNMP affecting many hardware platforms, products, and operating systems, that the possibility of a compromised host may have generated some of this traffic.

The following hosts are polling MY.NET.150.195 every 10 minutes:

1. MY.NET.88.159
2. MY.NET.88.136
3. MY.NET.88.145
4. MY.NET.88.203
5. MY.NET.88.207
6. MY.NET.88.212

These machines should be check to determine if this is a misconfiguration, or possible compromised system.

The following rules will detect this traffic:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP public access udp"; content:"public"; reference:cve,CAN-2002-0012; reference:cve,CAN-2002-0013; sid:1411; rev:1; classtype:attempted-recon;)

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

alert udp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP private access udp"; content:"private"; reference:cve,CAN-2002-0012; reference:cve,CAN-2002-0013; sid:1413; rev:1; classtype:attempted-recon;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 161 (msg:"SNMP private access tcp"; content:"private"; reference:cve,CAN-2002-0012; reference:cve,CAN-2002-0013; sid:1414; rev:1; classtype:attempted-recon;)
```

The top talkers were:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.70.177	20271	20292	32	32
MY.NET.150.198	5069	5069	102	102

MY.NET.153.220	2493	2493	1	1
MY.NET.88.203	1342	1425	1	4
MY.NET.88.159	1336	1376	1	3
MY.NET.88.145	1286	1287	1	2
MY.NET.88.207	1275	1370	1	5
MY.NET.150.245	1124	1124	1	1
MY.NET.153.178	883	1490	1	401
MY.NET.153.191	868	954	1	31

Correlation

Jeff Stutzman recorded this traffic on February 21, 2000, which he posted on incidents.org.

http://www.incidents.org/archives/y2k/022100-1130.htm

```
[**] SNMP access, public [**]
02/20-23:31:44.692432 216.80.8.56:1679 -> x.x.x.y:161
UDP TTL:113 TOS:0x0 ID:22711
Len: 246
30 81 EB 02 01 00 04 06 70 75 62 6C 69 63 A1 81 0.......public..
DD 02 02 54 68 02 01 00 02 01 00 30 81 D0 30 0B ...Th......0..0.
06 07 2B 06 01 02 01 01 01 05 00 30 0B 06 07 2B ..+........0...+
06 01 02 01 01 03 05 00 30 0B 06 07 2B 06 01 02 ........0...+...
01 01 05 05 00 30 0D 06 09 2B 06 01 02 01 02 02 .....0...+......
01 06 05 00 30 0D 06 09 2B 06 01 02 01 04 14 01 ....0...+.......
01 05 00 30 0E 06 0A 2B 06 01 02 01 19 03 02 01 ...0...+........
03 05 00 30 10 06 0C 2B 06 01 04 01 0B 02 03 09 ...0...+........
01 01 07 05 00 30 10 06 0C 2B 06 01 04 01 0B 02 .....0...+......
03 09 05 01 03 05 00 30 10 06 0C 2B 06 01 04 01 .......0...+....
0B 02 04 03 08 03 02 05 00 30 10 06 0C 2B 06 01 .........0...+..
04 01 0B 02 04 03 08 03 03 05 00 30 0F 06 0B 2B ...........0...+
06 01 04 01 0B 02 04 03 0A 07 05 00 30 0F 06 0B ............0...
2B 06 01 04 01 0B 02 04 03 0A 0D 05 00 30 0F 06 +............0..
0B 2B 06 01 04 01 0B 02 04 03 0D 01 05 00 .+............

[**] SNMP access, public [**]
02/20-23:31:44.818701 216.80.8.56:1679 -> x.x.x.y:161
UDP TTL:113 TOS:0x0 ID:22713
Len: 120
30 6E 02 01 00 04 06 70 75 62 6C 69 63 A1 61 02 0n.....public.a.
02 54 6A 02 01 00 02 01 00 30 55 30 0B 06 07 2B .Tj......0U0...+
06 01 02 01 01 01 05 00 30 0B 06 07 2B 06 01 02 ........0...+...
01 01 03 05 00 30 0B 06 07 2B 06 01 02 01 01 05 .....0...+......
05 00 30 0D 06 09 2B 06 01 02 01 02 02 01 06 05 ..0...+.........
00 30 0D 06 09 2B 06 01 02 01 04 14 01 01 05 00 .0...+..........
30 0E 06 0A 2B 06 01 02 01 19 03 02 01 03 05 00 0...+...........


[**] SNMP access, public [**]
02/20-23:31:45.035339 216.80.8.56:1674 -> x.x.x.y:161
UDP TTL:113 TOS:0x0 ID:22714
Len: 58
30 30 02 01 00 04 06 70 75 62 6C 69 63 A1 23 02 00.....public.#.
```

```
02 54 69 02 01 00 02 01 00 30 17 30 15 06 11 2B .Ti......0.0...+
06 01 02 01 03 01 01 02 02 01 81 46 52 81 77 61 ...........FR.wa
05 00 ..

[**] SNMP access, public [**]
02/20-23:31:45.131411 216.80.8.56:1674 -> x.x.x.y:161
UDP TTL:113 TOS:0x0 ID:22715
Len: 58
30 30 02 01 00 04 06 70 75 62 6C 69 63 A1 23 02 00.....public.#.
02 54 6B 02 01 00 02 01 00 30 17 30 15 06 11 2B .Tk......0.0...+
06 01 02 01 03 01 01 02 02 01 81 46 52 81 77 62 ...........FR.wb
05 00 ..

[**] SNMP access, public [**]
02/20-23:31:45.393930 216.80.8.56:1674 -> x.x.x.y:161
UDP TTL:113 TOS:0x0 ID:22716
Len: 58
30 30 02 01 00 04 06 70 75 62 6C 69 63 A1 23 02 00.....public.#.
02 54 6B 02 01 00 02 01 00 30 17 30 15 06 11 2B .Tk......0.0...+
06 01 02 01 03 01 01 02 02 01 81 46 52 81 77 62 ...........FR.wb
05 00 ..

[**] SNMP access, public [**]
02/20-23:31:45.489403 216.80.8.56:1674 -> x.x.x.y:161
UDP TTL:113 TOS:0x0 ID:22717
Len: 57
30 2F 02 01 00 04 06 70 75 62 6C 69 63 A1 22 02 0/.....public.".
02 54 6C 02 01 00 02 01 00 30 16 30 14 06 10 2B .Tl......0.0...+
06 01 02 01 03 01 01 02 02 01 81 60 00 00 00 05 ...........`....
00 .
```

Recommendation

- Install SNMP version 3 to protect the community strings when transmitted.
- Patch any hardware and operating systems that may be vulnerable to SNMP attacks.
- Block port 161 and 162 at the perimeter defense for inbound and outbound traffic.
- If there are eternal systems that needs managing, then setup a rule on your perimeter firewall/router that allows SNMP access only to host that need managing.

ICMP Echo Request L3retriever Ping

The rule that triggered this alert was:

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP L3retriever Ping"; content:
"ABCDEFGHIJKLMNOPQRSTUVWABCDEFGHI"; itype: 8; icode: 0; depth: 32; reference:arachnids,311;
classtype:attempted-recon; sid:466; rev:1;)
```

This rule detects ICMP packet sent from outside the university network to internal hosts meeting the follow meeting requirements:

Content: "ABCDEFGHIJKLMNOPQRSTUVWABCDEFGHI"
This parameter searches the payload for the character string "ABCDEFGHIJKLMNOPQRSTUVWABCDEFGHI"

Itype: 8
This parameter looks for an ICMP type field of 8, which is an echo request.

Icode: 0
This parameter examines the packet for ICMP code value.

Depth: 32
The parameter tells snort to search 32 bytes starting from the beginning of the payload.

This traffic is classified as a recon. The ICMP Echo Request L3retriever ping alert is the result of a security assessment tool called retriever 1.5. It has been reported that a win2k client talking to a win2k Domain Controller triggers a false positive alert and I believe that this is the case with the traffic sent to MY.NET.11.6 and MY.NET.11.7. I came to the conclusion that those hosts may be Windows 2000 domain controllers running Active Directory by the ports that were being connected to in the port scan log. Also these host were the top 2 source and destination for the SMB Wild card alert which also generates false positive in a windows environment.

Below are snips of the Port scan log.

```
Mar 25 14:04:50 MY.NET.152.166:2289 -> MY.NET.11.6:389 UDP
Mar 25 14:04:50 MY.NET.152.166:2292 -> MY.NET.11.6:389 SYN ******S*
Mar 25 14:26:13 MY.NET.152.162:3471 -> MY.NET.11.7:389 UDP
Mar 25 14:26:13 MY.NET.152.162:3473 -> MY.NET.11.7:389 SYN ******S*
Mar 25 18:35:28 MY.NET.152.14:3884 -> MY.NET.11.6:88 UDP
Mar 25 18:35:26 MY.NET.152.14:3882 -> MY.NET.11.6:389 SYN ******S*
Mar 25 18:37:15 MY.NET.152.252:2842 -> MY.NET.11.7:88 UDP
Mar 25 18:37:13 MY.NET.152.252:2840 -> MY.NET.11.7:389 SYN ******S*
```

Port 389 UDP/TCP is associated with Lightweight Directory access Protocol (LDAP). LDAP is an Internet standard for accessing directory services. A Windows 2000 domain running active directory uses LDAP to perform directory lookups, directory synchronization, and directory management.

Port 88 UDP/TCP is associated with Kerberos. Kerberos, a protocol used to provide mutual authentication between a client, such as a user, computer, or service, and a server. The Kerberos Key Distribution Center (KDC), which runs on every domain controller as part of Active Directory, is an intermediary between the clients.

Additional information can be found at the following site:

http://www.activeworx.com/arachnids/IDS311/event.html

http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0523

The top ten active talkers were:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.152.160	868	1792	3	3
MY.NET.152.161	561	1182	2	3
MY.NET.152.21	554	1508	3	175
MY.NET.152.251	545	1174	3	10
MY.NET.152.15	543	1193	3	8
MY.NET.152.171	538	1260	3	7
MY.NET.152.19	528	3657	2	24
MY.NET.152.173	526	1118	3	3
MY.NET.152.178	524	1132	3	7
MY.NET.152.174	523	1218	2	6

Correlation
Keven Murphy also experienced this traffic in his analysis.

Recommendation

- Configure snort.conf and rules to reduce the amount of false positives that may occur in relation to this attack. If this is not possible then it might be necessary to remove this alert all together or better yet remove from the sensors that are triggering this alert.

- Create a Network Scanning policy for conducting assessment/vulnerability scans. In it, identify what tools are allowed, who are allowed to perform scan and when scans can be performed.

  Note: Some scanners can crash servers, so schedule scanning after hours.

- State in your Acceptable use Policy what kind of computer activities is authorized.

- Designate one host to perform scans from.
- Allow only users with Administrator/root Privilege to run security tool or create a group with permissions to run.

MISC Large UDP Packet

The following rule triggered this alert:

```
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"MISC Large UDP Packet"; dsize: >4000;
reference:arachnids,247; classtype:bad-unknown; sid:521; rev:1;)
```

This rule is triggered when an UDP packet greater than 4000 bytes is sent from an external host and detected by the sensor. This kind of traffic can be associated with denial of service attack. Host 140.140.8.72 is suspected of attempting such an attack. The frequency of packets being transmitted would indicate a possible denial of service attack.

This kind of alert can also be triggered by games like Unreal and Quake, which allows players to compete against other players on the network or via the Internet.

Below is a list of games that can be played over the network and Internet:

GAME	PORT
Half Life	UDP 6003, UDP 7002,UDP 27010, UDP 27015, UDP 27025
MSN Game zone	TCP 6667, TCP 28800 - 29000
Quake	UDP 27910, UDP 27660
Starcraft	UDP 6112
Subspace	UDP 2001, 2010,2011,2020,2021,2030,2031
Unreal Tournament server	UDP 7777, UDP 7778

Neither of the ports for Unreal and Quake was seen in the scan logs but it is possible to configure the games to use different ports.

Some of this traffic was triggered by host MY.NET.153.153 that connected to a Windows media server. The Microsoft knowledge base article number Q189416 explains this traffic.

Excerpt from the Microsoft knowledge base article number Q189416:

When using UDP streams, the client first makes a connection to the Windows Media server using TCP port 1755. After this connection is established, the client and the server choose the UDP port that will be used by the server to stream the Windows Media content down to the client.

This network activity is illustrated in the traffic below.

```
Mar 26 12:26:42 MY.NET.153.153:3949 -> 66.28.104.154:1755 SYN ******S*
Mar 26 12:26:45 MY.NET.153.153:3858 -> 66.28.104.154:1755 UDP
```

Reference:
Excerpt from Microsoft Knowledge Base Article Q189416

The top ten talkers were:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
66.28.104.154	10805	10805	1	1
140.142.8.72	6203	6203	1	1
202.101.232.110	2314	2314	1	1
202.101.235.110	1132	1132	1	1
211.206.125.14	368	369	1	1
61.78.35.42	225	235	1	1
61.78.53.74	166	166	1	1
211.62.59.30	164	164	1	1
216.247.76.251	158	158	1	1
203.170.198.14	152	161	1	1

All of the sources addresses came from outside of the network. Here is the registration information for five of the top ten talkers of this traffic.
Search results for: 66.28.104.154

Cogent Communications (NETBLK-COGENT-NB-0000)
1015 31st Street, NW
Washington, DC 20007
US

Netname: COGENT-NB-0000
Netblock: 66.28.0.0 - 66.28.255.255
Maintainer: COGC

Coordinator:
Cogent Communications (ZC108-ARIN) noc@cogentco.com
+1-877-875-4311

Domain System inverse mapping provided by:

AUTH1.DNS.COGENTCO.COM 66.28.0.14
AUTH2.DNS.COGENTCO.COM 66.28.0.30

```
   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE
  Reassignment information for this block can be found at
  rwhois.cogentco.com 4321

  Record last updated on 05-Dec-2001.
  Database last updated on  13-Apr-2002 19:57:55 EDT.
```

Search results for: 140.142.8.72

```
NorthWestNet Network Operations Center (NET-UW-SEA)
  Academic Computing Center
  3737 Brooklyn NE
  Seattle, WA 98105
  US

  Netname: UW-SEA
  Netblock: 140.142.0.0 - 140.142.255.255
  Maintainer: UWND

  Coordinator:
    University, Of Washington  (OWU2-ARIN)  noc@CAC.WASHINGTON.EDU
    206-543-5128

  Domain System inverse mapping provided by:

  HANNA.CAC.WASHINGTON.EDU          140.142.5.5
  MARGE.CAC.WASHINGTON.EDU          140.142.5.13
  NS.UNET.UMN.EDU              128.101.101.101

  Record last updated on 17-Mar-2000.
  Database last updated on  13-Apr-2002 19:57:55 EDT.
```

Search results for: 202.101.232.110

```
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 3858 3100

  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

  Regional Internet Registry for the Asia-Pacific Region.

  *** Use whois -h whois.apnic.net [object]               ***
  *** or see http://www.apnic.net/db/ for database assistance   ***


  Record last updated on 18-Jun-1999.
  Database last updated on  13-Apr-2002 19:57:55 EDT.
```

Search results for: 202.101.235.110

```
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 3858 3100

  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

  Regional Internet Registry for the Asia-Pacific Region.

  *** Use whois -h whois.apnic.net [object]               ***
  *** or see http://www.apnic.net/db/ for database assistance   ***


  Record last updated on 18-Jun-1999.
  Database last updated on  13-Apr-2002 19:57:55 EDT.
```

Search results for: 211.206.125.14

```
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.
  AU

  Netname: APNIC-CIDR-BLK2
  Netblock: 210.0.0.0 - 211.255.255.255
  Coordinator:
        Administrator, System  (SA90-ARIN)  [No mailbox]
 +61 7 3858 3100

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.

*** Use whois -h whois.apnic.net [object]                 ***

  *** or see http://www.apnic.net/db/ for database     assistance   ***


Record last updated on 03-May-2000.
Database last updated on  13-Apr-2002 19:57:55 EDT.
```

Correlation

This traffic has been seen at the following sites:

http://online.securityfocus.com/archive/75/249598

http://www.incidents.org/archives/intrusions/msg01295.html

http://www.incidents.org/archives/intrusions/msg03699.html

Recommendation

- Review Acceptable use policy and verify that streaming video is allowed.
- If not allowed block ports as specified in Knowledge Base Article

INFO MSN IM Chat data

This traffic is triggered when an Instant messenger session was established. The following rule triggered the alert:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 1863 (msg:"INFO MSN IM Chat
data";flags:A+;
content:"|746578742F706C61696E|";  depth:100; classtype:not-suspicious; sid:540;
rev:1;)
```

This alert was triggered by a TCP packets originating from the university network destine to external hosts. Also, there were requirements in the rule option that needed to be meet for the packet to be triggered.

Below is a description of the Rule Option requirements:

flags:A+
This parameter examines the packet if the ACK and any other TCP flags are set.

content:"|746578742F706C61696E|";
This Parameter examines the payload for the character string for "|746578742F706C61696E|".

depth:100
The parameter tells snort to search the 100 bytes starting from the beginning of the payload.

This traffic is classified as not suspicious but you should confirm if it is allowed in the Acceptable use policy if available.

Additional information can be found at the following site:

Susan Willner
August 19, 2001
http://rr.sans.org/threats/IM2.php

Dan Frase
January 31, 2002
http://rr.sans.org/threats/IM_menace.php

Top ten talkers:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
64.4.12.178	526	526	5	5
MY.NET.150.165	490	906	9	31
MY.NET.153.177	443	443	32	32
64.4.12.158	290	290	8	8
MY.NET.150.242	265	320	11	12
MY.NET.153.113	263	2200	3	7
MY.NET.153.127	239	451	6	7
MY.NET.88.151	227	1573	15	59
64.4.12.156	226	226	4	4
MY.NET.153.199	208	573	6	14

Correlation

The following analyst saw this traffic:

Stan Hoffman
http://www.giac.org/practical/stan_hoffman_GCIA.doc

Recommendation

- If Instant messenger must be used I recommend using one that provides encryption like the following products:

  Kunani IM
  http://www.kunani.com/kim

  Imici messenger
  http://www.imici.com/

- Block traffic at perimeter defense if Instant messenger is not allowed in Acceptable use policy.

ICMP Echo Request Nmap or HPING2

This traffic is associated with network mapping. An Echo request generated by the NMAP or HPING2 is often used for reconnaissance probes of hosts or networks. There were 3875 alerts generated from 63 hosts destined to 296 hosts in the five days seen in analysis. The majority of alerts triggering this traffic was generated by hosts within MY.NET.152 network segment destined to hosts MY.NET.11.6 and MY.NET.11.7. When examining this traffic I notice that all of the hosts within MY.NET.152 networks activities where similar. Each host triggered ICMP Echo Request Nmap or HPING2, ICMP Echo Request L3retriever Ping AND SMB Name Wildcard. This may be happening due to the way the systems are configured. I recommend examining the host to confirm if it is a configuration problem or normal traffic.

When Nmap sends an ICMP Echo Request before starting to scan for the selected ports it sends an identical ICMP packet as HPING2 does by default. That will say no data at all. Since the ID change with every packet you can't tell if it's Nmap or HPING2 sending the ICMP Echo Request.

Host MY.NET.253.10 should be examined. This host is scanning other host within the university for port 6112. Port 6112 TCP/UDP is associated with Diablo, a popular multiplayer game that can be played via a local area network or Internet. This port is also used for Unix-based Common Desktop Environment (CDE) which runs typically as root, is vulnerable to a buffer overflow that would allow execution of arbitrary code with root privileges. This traffic also concerned me because in the same order that this alert was triggered the TCP Nmap ping was triggered indicating a scan of host on the network. Also there were Xmas scans, a stealth portscan technique, detected with these port scan.

Snip of traffic captured of host MY.NET.253.10:

```
Mar 28 15:06:57 MY.NET.253.10:48276 -> MY.NET.5.79:6112
SYN ******S*
Mar 28 15:06:57 MY.NET.253.10:48278 -> MY.NET.5.79:6112 XMAS **U*P**F
Mar 28 15:06:57 MY.NET.253.10:48265 -> MY.NET.5.79:6112 UDP
Mar 28 15:07:00 MY.NET.253.10:48276 -> MY.NET.5.79:6112
SYN ******S*
Mar 28 15:07:00 MY.NET.253.10:48278 -> MY.NET.5.79:6112
XMAS **U*P**F
Mar 28 15:07:00 MY.NET.253.10:48265 -> MY.NET.5.79:6112 UDP
Mar 28 15:07:12 MY.NET.253.10:48276 -> MY.NET.5.83:6112
SYN ******S*
Mar 28 15:07:12 MY.NET.253.10:48278 -> MY.NET.5.83:6112 XMAS **U*P**F
Mar 28 15:07:10 MY.NET.253.10:48265 -> MY.NET.5.83:6112 UDP
Mar 28 15:07:15 MY.NET.253.10:48276 -> MY.NET.5.87:6112
SYN ******S*
Mar 28 15:07:15 MY.NET.253.10:48278 -> MY.NET.5.87:6112
XMAS **U*P**F
Mar 28 15:07:15 MY.NET.253.10:48265 -> MY.NET.5.87:6112 UDP
```

Top ten talker were:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.253.10	311	1100	291	320
MY.NET.152.174	83	1218	1	6
MY.NET.152.21	79	1508	2	175
MY.NET.152.171	75	1260	1	7
MY.NET.152.19	74	3657	1	24
MY.NET.152.178	73	1132	2	7
MY.NET.152.251	73	1174	1	10
MY.NET.152.169	72	1071	1	2
MY.NET.152.15	72	1193	1	8
MY.NET.152.173	71	1118	2	3

Correlation

Keven Murphy also experienced this traffic in his analysis.
http://www.giac.org/practical/Keven_Murphy_GCIA.zip

Recommendation

- Remove Nmap and Hing2 off of hosts that have been identified.
- Inform users of the Acceptable use policy.
- Allow only users designated to run programs in the Scan policy access to programs.
- Designate a machine that network scans can be performed from.

INFO Inbound GNUTella Connect request

The following rule triggered the alert:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any (msg:"INFO Inbound GNUTella client request";
flags:A+; content:"GNUTELLA OK"; depth:40; classtype:misc-activity; sid:557; rev:3;)

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"INFO Inbound GNUTella client request";
flags:A+; content:"GNUTELLA CONNECT"; depth:40; classtype:misc-activity; sid:559; rev:3;)
```

These rules detect the Gnutella traffic. Gnutella is a peer-to-peer file sharing program. This program allows users to share files on their computers with other Gnutella users. This can be a problem because there are no logs of what information is being disturbed. Also, virus can be installed liked the mandragore worm, which affected gnutella clients due to the ability not to detect files transferred between users.

Top ten talkers:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
167.159.1.2	10	10	1	1
24.214.74.234	10	10	1	1
194.77.100.2	8	8	1	1
134.58.253.196	8	8	1	1
193.1.206.62	7	7	1	1
213.23.66.172	7	7	1	1
204.203.52.213	6	6	1	1
217.135.130.145	5	5	1	1
24.141.172.34	5	5	1	1
217.84.81.254	5	5	1	1

Correlation

Keven Murphy also experienced this traffic in his analysis.

Recommendation

- It is recommended that this peer to peer be blocked at the perimeter router and /or firewall.
- Continue monitoring P2P traffic because communication ports can be changed.
- Create an Acceptable use policy and define what traffic is acceptable.
- Enable Flexpres on Snort to break connection when a Gnutella Session is detected. You can do this with the following rule.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"INFO Inbound
GNUTella client request"; flags:A+; content:"GNUTELLA CONNECT";
depth:40; resp:rst_snd classtype:misc-activity; sid:559; rev:3;)
```

Watchlist 000220 IL-ISDNNET-990517

The purpose of a Watchlist rule is to monitor traffic coming from a network. There were a total of 23 hosts that triggered this alert all coming from the 212.179. network which is part of the IP block 212.0.0.0 – 212.255.255.255 that are assigned to different companies oversees in Europe.

Top ten talkers:

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
212.179.35.118	1554	1554	4	4
212.179.27.176	288	288	4	4
212.179.35.8	112	112	2	2
212.179.35.119	76	76	5	5
212.179.45.203	30	30	2	2
212.179.18.20	29	29	2	2
212.179.35.121	24	24	4	4
212.179.27.6	22	22	1	1
212.179.7.58	21	21	1	1
212.179.44.2	13	13	1	1

The majority of this traffic was inbound HTTP traffic from web servers. There was also a lot of Kazaa traffic. Kazaa is a Napster-like file-sharing Peer-to-Peer program. These programs tend to affect network bandwidth with the constant downloading of files.

The following talkers connected to port 1214:

212.179.27.6
212.179.45.203
212.179.18.20
212.179.7.58
212.179.44.2

Host 212.179.35.119 was the only source with a source port of 1214 indicating that the connection was made from with MY.NET.

Correlation

John Green saw this traffic on the 19th and 20th of May 2000 and posted it on incidents.org.

http://www.incidents.org/archives/y2k/051900.htm

http://www.incidents.org/archives/y2k/052000.htm

Recommendation

- Review Acceptable use policy to confirm that file sharing programs is allowed.

- If traffic does not cohere to the policy block access at perimeter defense for default ports and continue to monitor for traffic because ports can be changed.

Computers to investigate for possible comprise or suspicious activity

- Host MY.NET.153.46 should be inspected for the nimda/Code Red virus. These alerts were triggered from host 64.124.157.32 and caused concern.

```
03/27-12:46:57.050177 [**] High port 65535 udp - possible Red Worm - traffic [**]
64.124.157.32:65535 -> 192.168.153.46:65535
03/27-12:48:33.824144 [**] TFTP - External UDP connection to internal tftp server [**]
64.124.157.32:256 -> 192.168.153.46:69
```

- Also the TFTP – Internal UDP connection to external tftp server rule needs to be configured so it can detect internal traffic destined to external tftp servers.

```
03/27-09:34:55.166905 [**] TFTP - Internal UDP connection to external tftp server [**] 64.124.157.32:69 -
> 192.168.153.46:54461
```

- There were 104 alerts triggered for the possible Trojan server activity alert by 15 hosts. Out of the 15 host there were 4 external host identified initiating traffic from port 27374, which is the default port for Sub seven Trojan .

```
03/29-16:46:47.897857 [**] Possible trojan server activity [**] 202.145.95.22:27374 ->
192.168.150.143:4662
03/29-16:46:47.898081 [**] Possible trojan server activity [**] 192.168.150.143:4662 ->
202.145.95.22:27374

03/28-07:28:40.792216 [**] Possible trojan server activity [**] 61.220.153.49:27374 ->
192.168.150.220:4662
03/28-01:31:32.233854 [**] Possible trojan server activity [**] 192.168.150.220:4662 ->
210.63.108.34:27374

03/25-16:23:45.364926 [**] Possible trojan server activity [**] 64.12.96.7:27374 ->
192.168.5.96:80
03/25-16:23:45.364995 [**] Possible trojan server activity [**] 192.168.5.96:80 ->
64.12.96.7:27374

03/28-01:31:31.842051 [**] Possible trojan server activity [**] 210.63.108.34:27374 ->
192.168.150.220:4662
03/28-01:31:32.233854 [**] Possible trojan server activity [**] 192.168.150.220:4662 ->
210.63.108.34:27374

03/27-20:38:14.837094 [**] Possible trojan server activity [**] 62.30.220.235:27374 ->
192.168.150.113:1214
```

```
03/27-20:38:14.839676 [**] Possible trojan server activity [**] 192.168.150.113:1214 ->
62.30.220.235:27374
```

Some of this traffic may be normal like network activity between 64.12.96.7:27374 -> 192.168.5.96:80. The client seems to have initiated communication using ephemeral port 27374 to MY.NET.5.96 on port 80 illustrating normal http traffic.

All other host traffic triggering this alert originated within the unviresity network with MY.NET hosts connecting to other MY.NET host on port 27374.

The following hosts need to be scanned and cleaned with an anti virus program for the Sub seven Trojan:

Host	**Comment**
MY.NET.5.50	Possible Sub Seven infections
MY.NET.5.88	Possible Sub Seven infections
MY.NET.5.29	Possible Sub Seven infections
MY.NET.5.92	Possible Sub Seven infections
MY.NET.5.44	Possible Sub Seven infections
MY.NET.5.83	Appears to be scanning internal and Internet host for infected machines
MY.NET.150.220	Appears to be scanning internal and Internet host for infected machines
MY.NET.150.113	Possible Sub Seven infections. Also host 62.30.220.235 was able to from the internet.

- Inspect hosts' MY.NET.5.44 and MY.NET.5.67 server logs for any abnormal activities. Both of these machines received traffic that triggered the suspicious host traffic alert. MY.NET.5.44 received traffic from 24.188.117.164, 68.10.172.34 and 198.26.130.37 triggering 117 alerts. They attempted to access port 135, 80 and 9127. When I examined the port scan and oos files I did not see any of these hosts. Hosts MY.NET.5.67 received suspicious traffic from 134.67.6.3 and 207.202.185.114. These hosts were also not seen in the portscan and oos files.

```
03/28-20:40:09.476327 [**] suspicious host traffic [**] 24.188.117.164:4407 -> 192.168.5.44:135
03/28-20:40:09.932530 [**] suspicious host traffic [**] 68.10.172.34:1119 -> 192.168.5.44:80
03/28-20:46:58.299680 [**] suspicious host traffic [**] 198.26.130.37:53673 -> 192.168.5.44:80
03/28-20:46:59.307626 [**] suspicious host traffic [**] 198.26.130.37:53673 -> 192.168.5.44:80
```

03/27-16:16:52.206158 [**] suspicious host traffic [**] 134.67.6.3:38138 -> 192.168.5.67:80
03/27-16:56:00.736109 [**] suspicious host traffic [**] 207.202.185.114:51005 -> 192.168.5.67:80

Analysis of OOS Files

The majority of traffic found in the oos file triggered the queso fingerprinting rule. This traffic is associated with host running queso, a program used for identifying remote PCs. It works by sending a host a malformed packet and then determines what the remote OS is by the way it responds to the packet. Below are snips of the scan log of queso fingerprinting traffic that was captured by host identified in the oos logs.

```
Mar 25 13:14:10 61.198.200.52:10073 -> MY.NET.153.45:6346 SYN 12****S*
RESERVEDBITS

Mar 25 14:58:18 80.145.117.134:3156 -> MY.NET.153.45:6346 SYN 12****S*
RESERVEDBITS

Mar 27 15:31:54 217.82.123.75:46197 -> MY.NET.152.21:6346 SYN 12****S*
RESERVEDBITS

Mar 27 17:56:26 80.144.189.160:63486 -> MY.NET.153.196:6346 SYN 12****S*
RESERVEDBITS

Mar 27 21:22:29 140.110.30.59:32862 -> MY.NET.150.220:4662 SYN 12****S*
RESERVEDBITS

Mar 28 06:44:31 80.133.124.114:3621 -> MY.NET.150.113:1214 SYN 12****S*
RESERVEDBITS

Mar 27 21:22:29 140.110.30.59:32862 -> MY.NET.150.220:4662 SYN 12****S*
RESERVEDBITS

Mar 28 06:44:31 80.133.124.114:3621 -> MY.NET.150.113:1214 SYN 12****S*
RESERVEDBITS
```

There were a few detections of traffic generated by NMAP in the oos file. NMAP is a security tool that is used to assess a host or network. Some features of NMAP is it's ability to port scan a host/network for ports that are listening and detect an OS by sending certain packets. This traffic looks like one of the seven packets that are used to identify an OS. Below are snips of the scan and oos logs.

Scan log snip

```
Mar 27 15:24:24 213.169.245.41:3800 -> MY.NET.152.21:6346 NMAPID *2U*P*SF RESERVEDBITS
```

OOS log snip

```
03/27-15:24:28.649944 213.169.245.41:3800 -> MY.NET.152.21:6346
TCP TTL:110 TOS:0x0 ID:408  DF
2*SF*P*U Seq: 0x3F7473   Ack: 0x20736D61   Win: 0x6564
```

```
68 65 61 64 20 77 69 74 68 20                 head with

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
03/27-15:24:29.845479 213.169.245.41:3800 -> MY.NET.152.21:6346
TCP TTL:110 TOS:0x0 ID:5784  DF
2*SF*P*U Seq: 0x3F7473   Ack: 0x20736D61   Win: 0x6564
68 65 61 64 20 77 69 74 68 20                 head with
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

There was 2 two attempts of hosts trying stealth scans. The stealth scanning methods used were SYN/FIN and XMAS shown below.

```
Mar 25 20:19:18 128.214.182.22:4242 -> MY.NET.150.113:4662 SYNFIN *2****SF
RESERVEDBITS

Mar 28 23:09:44 61.216.83.124:64835 -> MY.NET.150.220:4662 FULLXMAS *2UAPRSF
RESERVEDBITS
```

I recommend monitoring the hosts in the oos files to track further activities.