Global Information Assurance Certification Paper

SANS GIAC Track 3 – Intrusion Detection In Depth

GCIA Practical Assignment – SANS 2001 San Francisco
Practical Assignment Version 3.0 – 2/18/2002

Colby M. DeRodeff

Table Of Contents

Assignment 1 The State of Intrusion Detection

Got Correlation? Not Without Normalization

Introduction

There have been many attempts by various groups to develop a standard that can take an event from any source and convert it to a standard format. Why would an analyst want something like this? Lets look at what steps an analyst might use to determine if their network may have been compromised. A network Intrusion Detection System (IDS) detects a web exploit targeted at a web server. First the analyst would review the perimeter router logs to see if the router passed the packet that triggered the alert. Based on the nature of this exploit, the probability that the packet was forward through the router is high. This is due to the fact that the exploit uses a standard TCP port (80). Second an analyst would want to review the firewall logs to see if this was blocked by any of the filters which are in place there, since the firewall is statefull it could have blocked something that the router may have passed as acceptable traffic. At this point the analyst is sure that the packets reached the webserver so further investigation is necessary. Since the exploit reached the web server the integrity of that box must be checked.

Third, to check the integrity of the webserver and look at all traffic that originated from the compromised box the analyst would run Tripwire, which is a file integrity checker using MD5 checksums, to see which files if any have been accessed or modified.

Fourth the analyst would look at the Syslog output or the EventLog from that server, as well as pull the tcpdump data off the dedicated tcpdump host for that segment for the time surrounding the attack to see what actually happened. Already the analyst would have accessed four different systems and looked at five different types of logs. That's a lot of work, and it takes time that could be spent securing the network and cleaning the compromised server to make sure that no other systems can be affected. It is preferred to have all the relevant data located in one logging facility allowing the analyst to sort by time to look at the sequence of events as they occurred.

In the art of intrusion detection there are many sources from which we can obtain information that can lead to an explanation, or the conformation of an exploit targeted at one of your network systems. Confirming that you have been the victim of an attack is like putting together a puzzle; the problem is that the pieces are all from different puzzles. When investigating an incident an analyst is dealing with a heterogeneous environment, where each device has a different logging format and reporting mechanism. He will also have logs from remote sites, where security policies and procedures will be different, different types of network devices, host based IDS, network based IDS, and different types of operating system and application logs. That's why the Industry needs normalization and correlation.

Correlation

What is correlation? Correlation is derived from the word correlate that means to be in or bring into mutual relation. That's the dictionary definition, but the "information security world" interprets correlation as having the ability to access, analyze, and relate different

attributes of events from multiple sources to bring something to the attention of an analyst that would have went unnoticed otherwise. Referring to the earlier example, correlating the accepted packet on the perimeter router, the accepted packet on the firewall, the IDS alert that detected a web exploit headed for the webserver, all coming from the same source IP address, along with the results of the integrity check, make it easier to confirm and reinforce the determination that the web server was indeed compromised and further action is necessary. As of now there is no commercially available tool that allows for this capability because the logs from all these devices are stored in different formats, and in different locations. In analysis it would be ideal to access all the logs from the entire enterprise from a single console, and have them stored in one common database. A database would be the most logical central storage facility because of the functionality it would allow for, such as querying and reporting. To accomplish this an analyst would first need to get the logs from all these devices, normalize them, and insert them into the database so they could be stored in a common format. In order to have real correlation we must start with normalization.

Normalization

How does normalization, meaning conforming to an accepted standard or norm apply to hunting down hackers and examining log files? Picture a typical enterprise environment, it consists of many different types of network devices ranging from border routers, VPN devices, to firewalls, to authentication servers, along with an even wider range of application servers like webservers, email servers, and always-critical database servers. All these different devices generate logs that are critical to an analyst who's responsible for the security of the site. It is seldom if ever that two different manufactures or vendors will use the same logging mechanism, format their logs differently. For example a Cisco PIX will not report an accepted packet the same as a Checkpoint firewall or even the same as a Cisco Router. The fact that the formats are all different makes it virtually impossible to store the log data in a common location such as a database without normalizing the events first.

The following are logs from different network devices all reporting on the exact same packet traveling across the network. These logs represent a remote printer buffer overflow that connects to IIS servers over port 80.

CheckPoint:

```
"14"  "21Dec2001"  "12:10:29"  "eth-s1p4c0"  "ip.of.firewall"  "log"
"accept"  "www-http"  "65.65.65.65"  "10.10.10.10"  "tcp"  "4"  "1355"
""  ""  ""  ""  ""  ""  ""  ""  ""  "firewall"  " len 68"
```

Cisco Router:

```
Dec 21 12:10:27: %SEC-6-IPACCESSLOGP: list 102 permitted tcp
65.65.65.65(1355) -> 10.10.10.10(80), 1 packet
```

Cisco PIX:

```
Dec 21 2001 12:10:28: %PIX-6-302001: Built inbound TCP connection
125891 for faddr 65.65.65.65/1355 gaddr 10.10.10.10/80 laddr
10.0.111.22/80
```

Snort:

```
[**] [1:971:1] WEB-IIS ISAPI .printer access [**]
[Classification: Attempted Information Leak] [Priority: 3]
```

```
12/21-12:10:29.100000 65.65.65.65:1355 -> 10.10.10.10:80
TCP TTL:63 TOS:0x0 ID:5752 IpLen:20 DgmLen:1234 DF
***AP*** Seq: 0xB13810DC  Ack: 0xC5D2E066  Win: 0x7D78  TcpLen: 32
TCP Options (3) => NOP NOP TS: 493412860 0
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0241]
[Xref => http://www.whitehats.com/info/IDS533]
```

All these formats are different and would be impossible to store in a database with out normalizing them first. Looking at the checkpoint record it contains the following fields: event id, date, time, firewall interface, IP address of the firewall interface, logging facility, action, service, source IP, target IP, protocol, source port, some checkpoint specific fields and then the size of the datagram. This is the most obscure format and it's especially hard to read with all the empty fields that are represented by double quotes. Now the Cisco router has a different format the fields it populates are date, time, logging facility, event name, source IP, source port, target address, target port, and number of packets. The Cisco PIX, which one would expect to have the same format as the Cisco router since they are made by the same company, uses date, time, event name, source IP, source port, translated address or target address, target port, local address, and local port. The final record is the Snort alert that claims this traffic was malicious. The fields Snort populates are exploit or event name, classification, priority, date, time, source IP, source port, target IP, target port, protocol, TTL (Time to Live), type of service, ID, IP length, datagram length, tcp flags, sequence number, acknowledgement number, window size, and tcp length. Snort also includes additional data such as references to investigate this exploit.

So how could these events possibly be stored in a common format in a database? It must first be decided which fields are interesting and develop a schema to accommodate the different fields that are populated by these devices. Choosing the fields must be content driven not based on semantic differences between what Checkpoint may call target address and what Cisco calls destination address. Next a parser must be coded to pull out those values from the event and populate the corresponding fields in the database. So pretend that the following table is from a database containing these alerts after they have been normalized.

Date	Time	Event_Name	Src_IP	Src_Port	Tgt_IP	Tgt_Port	Device_Type	Additional_data
21-Dec-01	12:10:29	accept	65.65.65.65	1355	10.10.10.10	80	CheckPoint	
21-Dec-01	12:10:27	list 102 permitted tcp	65.65.65.65	1355	10.10.10.10	80	Cisco Router	
21-Dec-01	12:10:28	Built inbound TCP connection	65.65.65.65	1355	10.10.10.10	80	Cisco PIX	
21-Dec-01	12:10:29	WEB-IIS ISAPI .printer access	65.65.65.65	1355	10.10.10.10	80	Snort	TCP TTL:63 TOS:0x0 ID:5752 IpLen:20 DgmLen:1234 DF ***AP*** Seq: 0xB13810DC Ack: 0xC5D2E066 Win: 0x7D78 TcpLen: 32 TCP Options (3) => NOP NOP TS: 493412860 0

These are the same four events we looked at earlier except they have been normalized. This would be ideal for an analyst investigating an incident. With the data organized like this one could pull all records containing a value that's of interest or sort by any field that may be relevant. This would make it extremely more efficient to investigate what

occurred during the course of an attempted exploit and whether or not the attack was indeed successful. The problem is that just putting this data into a spreadsheet manually was easy but to get a program to do it would be much more difficult. For instance the checkpoint firewall reports target port as www-http not 80 like most devices. Therefore there must be a lookup mechanism to ensure that www-http gets translated into port 80 otherwise this value would be useless in correlation. Another complication would be converting the date/timestamps. Since the devices all use a different format the program couldn't just parse out the time stamp reported by the device it would also need to convert it to a common format such as GMT.

Developing a Standard

What is needed industry wide is a standard that supports interoperability. There have been several groups of engineers that have tried to accomplish this task. The first group recognized by Security Professionals is the CIDF working group, which was sponsored by DARPA (Defense Advanced Research Projects Agency). CIDF stands for Common Intrusion Detection Framework. The goal of the group was to provide common message formats and exchange procedures for interoperability and a common understanding between intrusion detection systems. They discovered that it was necessary to express the information in a format that all of the systems could understand and interpret. CIDF seems to have phased out but it has provided a framework and a set of guidelines that have been partially adopted by another group. The Network Intrusion Detection An Analyst's Handbook says "The effort (CIDF) did a great service to the community, however, by trying to establish a vocabulary to discuss intrusions." So the work was not in vein it just never became accepted as the industry standard. According to the SANS course material, "The current status of CIDF is unclear, though some of it's efforts may have been overtaken by the Intrusion Detection Working Group (IDWG)."

The IDWG is another group of engineers who are working towards developing a standard data format called IDMEF. IDMEF is Intrusion Detection Message Exchange Format. Some of the benefits that IDMEF could provide in future implementations range from a single database containing logs from different security products, to the foundation for an event correlation system which could accomplish cross vendor and cross platform correlation. IDMEF is based on an object orientated data model that allows for flexibility. Different alerts will have different needs; some will offer much more information than others requiring additional objects to be added to the model. They chose an object oriented model because of the ability to subclass which allows them to extend the model, meaning that one system may not know what all the objects in the alert mean but they will still be able to interpret the values which are of concern to them. In their white paper the IDWG states, " The goal of the data model is to provide a standard representation of the information that an intrusion-detection analyzer detected an occurrence of some unusual activity. These alerts may be simple or complex, depending on the capabilities of the analyzer that created them." There has yet to be any large-scale implementation of IDMEF in the commercial market but in the open source world a company called Silicon Defense (www.silicondefense.org) has implemented an IDMEF compliant plugin for the Snort NIDS. The biggest problem is getting the commercial IDS vendors to see a value in interoperability. They don't want people to be able to mix and

match their expensive commercial products with open source products that you can get for free. From their business perspective it is not high on the priority list, they would much rather force you to stick with their product line whether it solves your problem or not.

Conclusion

In the ever-changing world of intrusion detection there is a definite need for data normalization. Looking at logs in twenty different formats and on four different consoles, as well as trying to find all the events across the network that may pertain to the attack being investigated is one of the hardest parts of any analyst's job. There is no way to visualize the sequence of events when they are all stored in different locations, and visualization is one of the keys to deciphering a network attack. The ability to relate and analyze events from a multitude of vendors, from a variety of intrusion detection devices, and from all the event generating devices that make up the common enterprise would make every analyst's puzzle a little easier to solve.

References

1) D. Curry, H. Debar, M. Huang "IDMEF Data Model and XML DTD" December 05, 2000 http://www.oasis-open.org/cover/IDMEF-provisional-draft-ietf-idwg-idmef-xml-02.html
2) Brian Tung, Dan Schnackenberg "The Common Intrusion Detection Framework" March 02, 1999 http://www.isi.edu/~brian/cidf/papers/cidf-isw.txt
3) Jeffrey Posluns "Security Monitoring and Incident Response: Definitions Of An Optimal Solution" October 11,2001 http://www.secureops.com/en/resources/white_papers/Security_Monitoring.pdf
4) Stephen Northcutt, Judy Novak "Network Intrusion Detection An Analyst's Handbook Second Edition" September 2000
5) StephenNorthcutt "IDS Signatures and Analysis, Parts 1 and 2" Version 4.1 SANS class material San Francisco 12-2000

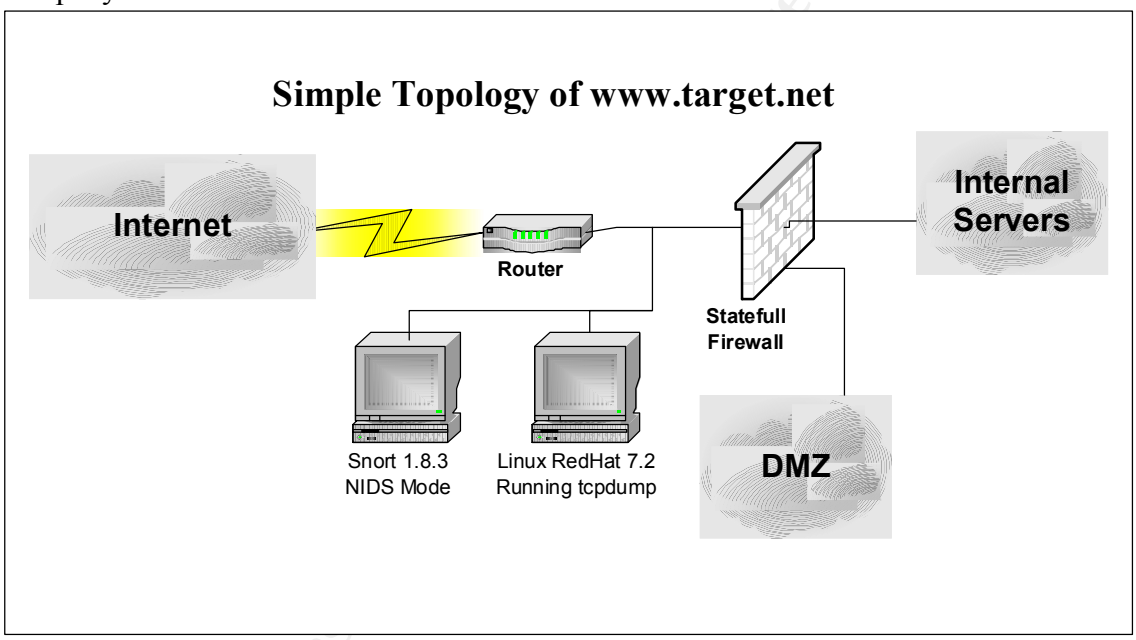
Security Focus, Nathan Einwechter **"**An Introduction To Distributed Intrusion Detection Systems" January 8, 2001 http://securityfocus.com/infocus/1532

Assignment 2 Network Detects

Introduction

I collected the following detects from a network that I have access to as part of my job. Please note that the security policies of target.net are not under my control nor do I have any influence as to getting them to change their policies. I do not agree with most of the security measures they have in place.

I setup the IDS devices on their network so that I have snort running outside the firewall with a passive interface. I have also disabled ARP so that there is no way to gain access to the snort box from any external source. I also have another workstation running tcpdump for packet analysis, configured in the same manner. I have included a simple topology of the network that generated these detects. For the purpose of this paper I have labeled it www.target.net. The name is entirely fictitious and makes no reference to a real company.



Detect1 - FTP Exploit Wu-Ftpd 2.6.0 site exec format string overflow Linux

Snort Alerts

```
[**] [1:344:2] FTP EXPLOIT wu-ftpd 2.6.0 site exec format string
overflow Linux [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
01/29-00:38:13.637286 attacker.net.6.123:3499 -> target.net.106.232:21
TCP TTL:46 TOS:0x0 ID:19210 IpLen:20 DgmLen:448 DF
***AP*** Seq: 0x5467BB43  Ack: 0xD9365ABF  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 14086300 708445
[Xref => http://www.securityfocus.com/bid/1387]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0573]
[Xref => http://www.whitehats.com/info/IDS287]

[**] [1:361:2] FTP site exec [**]
```

```
[Classification: Potentially Bad Traffic] [Priority: 2]
01/29-00:38:14.484539 attacker.net.6.123:3499 -> target.net.106.232:21
TCP TTL:46 TOS:0x0 ID:19213 IpLen:20 DgmLen:82 DF
***AP*** Seq: 0x5467BCCF  Ack: 0xD9365D5C  Win: 0x19D3  TcpLen: 32
TCP Options (3) => NOP NOP TS: 14086386 708464
[Xref => http://www.securityfocus.com/bid/2241]
[Xref => http://www.whitehats.com/info/IDS317]

[**] [1:361:2] FTP site exec [**]
[Classification: Potentially Bad Traffic] [Priority: 2]
01/29-00:38:19.996423 attacker.net.6.123:3499 -> target.net.106.232:21
TCP TTL:46 TOS:0x0 ID:19226 IpLen:20 DgmLen:552 DF
***AP*** Seq: 0x5467C362  Ack: 0xD93669D8  Win: 0x4704  TcpLen: 32
TCP Options (3) => NOP NOP TS: 14086936 709017
```
[Xref => http://www.securityfocus.com/bid/2241][Xref => http://www.whitehats.com/info/IDS317]

TCPDUMP Output From These Alerts
Alert1

I ran the tcpdump output through ethereal and generated a text file because it is much easier to read. What these packets show is the initial connection to port 21 on my FTP server from the attacker. Frame 9 contains the exploit that triggers the first snort alert.

```
Frame 6 (75 on wire, 75 captured)
    Arrival Time: Jan 29, 2002 00:38:13.4780
    Time delta from previous packet: 0.000993 seconds
    Time relative to first packet: 0.775609 seconds
    Frame Number: 6
    Packet Length: 75 bytes
    Capture Length: 75 bytes
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 61
    Identification: 0x4b09
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 46
    Protocol: TCP (0x06)
    Header checksum: 0xb049 (correct)
    Source: attacker.net.6.123 (attacker.net.6.123)
    Destination: target.net.106.232 (target.net.106.232)
Transmission Control Protocol, Src Port: 3499 (3499), Dst Port: 21
(21), Seq: 1416084282, Ack: 3644218047
    Source port: 3499 (3499)
    Destination port: 21 (21)
    Sequence number: 1416084282
    Next sequence number: 1416084291
    Acknowledgement number: 3644218047
```

```
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0x82a3 (correct)
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 14086285, tsecr 708429
File Transfer Protocol (FTP)
    Request: user
    Request Arg: ftp

   0  0000 0000 0001 0000 0c1f 385b 0800 4500   ..........8[..E.
  10  003d 4b09 4000 2e06 b049 xxxx xxxx xxxx   .=K.@....IA!c.Aw
  20  xxxx 0dab 0015 5467 bb3a d936 5abf 8018   j.....Tg.:.6Z...
  30  16d0 82a3 0000 0101 080a 00d6 f08d 000a   ................
  40  cf4d 7573 6572 2066 7470 0a               .Muser ftp.

Frame 7 (66 on wire, 66 captured)
    Arrival Time: Jan 29, 2002 00:38:13.4780
    Time delta from previous packet: 0.000019 seconds
    Time relative to first packet: 0.775628 seconds
    Frame Number: 7
    Packet Length: 66 bytes
    Capture Length: 66 bytes
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP; ECN:
0x00)
        0001 00.. = Differentiated Services Codepoint: Unknown (0x04)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 52
    Identification: 0x37df
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0xb16c (correct)
    Source: target.net.106.232 (target.net.106.232)
    Destination: attacker.net.6.123 (attacker.net.6.123)
Transmission Control Protocol, Src Port: 21 (21), Dst Port: 3499
(3499), Seq: 3644218047, Ack: 1416084291
    Source port: 21 (21)
    Destination port: 3499 (3499)
    Sequence number: 3644218047
```

```
    Acknowledgement number: 1416084291
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 32120
    Checksum: 0x95af (correct)
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 708445, tsecr 14086285

   0  0002 b319 e4c6 0000 0000 0000 0800 4510   ..............E.
  10  0034 37df 4000 4006 b16c xxxx xxxx xxxx   .47.@.@..lAwj.A!
  20  xxxx 0015 0dab d936 5abf 5467 bb43 8010   c......6Z.Tg.C..
  30  7d78 95af 0000 0101 080a 000a cf5d 00d6   }x...........]..
  40  f08d                                      ..

Frame 8 (134 on wire, 134 captured)
    Arrival Time: Jan 29, 2002 00:38:13.4790
    Time delta from previous packet: 0.000958 seconds
    Time relative to first packet: 0.776586 seconds
    Frame Number: 8
    Packet Length: 134 bytes
    Capture Length: 134 bytes
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP; ECN:
0x00)
        0001 00.. = Differentiated Services Codepoint: Unknown (0x04)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 120
    Identification: 0x37e0
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0xb127 (correct)
    Source: target.net.106.232 (target.net.106.232)
    Destination: attacker.net.6.123 (attacker.net.6.123)
Transmission Control Protocol, Src Port: 21 (21), Dst Port: 3499
(3499), Seq: 3644218047, Ack: 1416084291
    Source port: 21 (21)
    Destination port: 3499 (3499)
    Sequence number: 3644218047
    Next sequence number: 3644218115
    Acknowledgement number: 1416084291
```

```
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 32120
    Checksum: 0xabbe (correct)
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 708445, tsecr 14086285
File Transfer Protocol (FTP)
    Response: 331
    Response Arg: Guest login ok, send your complete e-mail address as
password.

   0  0002 b319 e4c6 0000 0000 0000 0800 4510   ..............E.
  10  0078 37e0 4000 4006 b127 xxxx xxxx xxxx   .x7.@.@..'Awj.A!
  20  xxxx 0015 0dab d936 5abf 5467 bb43 8018   c......6Z.Tg.C..
  30  7d78 abbe 0000 0101 080a 000a cf5d 00d6   }x...........]..
  40  f08d 3333 3120 4775 6573 7420 6c6f 6769   ..331 Guest logi
  50  6e20 6f6b 2c20 7365 6e64 2079 6f75 7220   n ok, send your
  60  636f 6d70 6c65 7465 2065 2d6d 6169 6c20   complete e-mail
  70  6164 6472 6573 7320 6173 2070 6173 7377   address as passw
  80  6f72 642e 0d0a                            ord...

Frame 9 (462 on wire, 144 captured)
    Arrival Time: Jan 29, 2002 00:38:13.6372
    Time delta from previous packet: 0.158246 seconds
    Time relative to first packet: 0.934832 seconds
    Frame Number: 9
    Packet Length: 462 bytes
    Capture Length: 144 bytes
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 448
    Identification: 0x4b0a
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 46
    Protocol: TCP (0x06)
    Header checksum: 0xaec5 (correct)
    Source: attacker.net.6.123 (attacker.net.6.123)
    Destination: target.net.106.232 (target.net.106.232)
```

```
Transmission Control Protocol, Src Port: 3499 (3499), Dst Port: 21
(21), Seq: 1416084291, Ack: 3644218047
    Source port: 3499 (3499)
    Destination port: 21 (21)
    Sequence number: 1416084291
    Next sequence number: 1416084687
    Acknowledgement number: 3644218047
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0x3dee
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 14086300, tsecr 708445
File Transfer Protocol (FTP)
    Request: pass
    Request Arg:
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\22
0\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\2
20\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\
220\220\220\2

   0  0000 0000 0001 0000 0c1f 385b 0800 4500   ..........8[..E.
  10  01c0 4b0a 4000 2e06 aec5 xxxx xxxx xxxx   ..K.@.....A!c.Aw
  20  xxxx 0dab 0015 5467 bb43 d936 5abf 8018   j.....Tg.C.6Z...
  30  16d0 3dee 0000 0101 080a 00d6 f09c 000a   ..=.............
  40  cf5d 7061 7373 2090 9090 9090 9090 9090   .]pass .........
  50  9090 9090 9090 9090 9090 9090 9090 9090   ................
  60  9090 9090 9090 9090 9090 9090 9090 9090   ................
  70  9090 9090 9090 9090 9090 9090 9090 9090   ................
  80  9090 9090 9090 9090 9090 9090 9090 9090   ................
```

Alerts 2,3

```
Frame 14 (96 on wire, 96 captured)
    Arrival Time: Jan 29, 2002 00:38:14.4845
    Time delta from previous packet: 0.606363 seconds
    Time relative to first packet: 1.782085 seconds
    Frame Number: 14
    Packet Length: 96 bytes
    Capture Length: 96 bytes
Ethernet II
    Destination: 00:00:00:00:00:01 (00:00:00:00:00:01)
    Source: 00:00:0c:1f:38:5b (00:00:0c:1f:38:5b)
    Type: IP (0x0800)
Internet Protocol
    Version: 4
    Header length: 20 bytes
```

```
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 82
    Identification: 0x4b0d
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 46
    Protocol: TCP (0x06)
    Header checksum: 0xb030 (correct)
    Source: 129.79.6.123 (129.79.6.123)
    Destination: 65.119.106.232 (65.119.106.232)
Transmission Control Protocol, Src Port: 3499 (3499), Dst Port: 21
(21), Seq: 1416084687, Ack: 3644218716
    Source port: 3499 (3499)
    Destination port: 21 (21)
    Sequence number: 1416084687
    Next sequence number: 1416084717
    Acknowledgement number: 3644218716
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 6611
    Checksum: 0x7e63 (correct)
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 14086386, tsecr 708464
File Transfer Protocol (FTP)
    Request: SITE
    Request Arg: EXEC %x %x %x %x +%x |%x

   0  0000 0000 0001 0000 0c1f 385b 0800 4500   ..........8[..E.
  10  0052 4b0d 4000 2e06 b030 xxxx xxxx xxxx   .RK.@....0A!c.Aw
  20  xxxx 0dab 0015 5467 bccf d936 5d5c 8018   j.....Tg...6]\..
  30  19d3 7e63 0000 0101 080a 00d6 f0f2 000a   ..~c............
  40  cf70 5349 5445 2045 5845 4320 2578 2025   .pSITE EXEC %x %
  50  7820 2578 2025 7820 2b25 7820 7c25 780a   x %x %x +%x |%x.

Frame 36 (566 on wire, 144 captured)
    Arrival Time: Jan 29, 2002 00:38:19.9964
    Time delta from previous packet: 0.601567 seconds
    Time relative to first packet: 7.293969 seconds
    Frame Number: 36
    Packet Length: 566 bytes
    Capture Length: 144 bytes
Ethernet II
```

```
    Destination: 00:00:00:00:00:01 (00:00:00:00:00:01)
    Source: 00:00:0c:1f:38:5b (00:00:0c:1f:38:5b)
    Type: IP (0x0800)
Internet Protocol
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 552
    Identification: 0x4b1a
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 46
    Protocol: TCP (0x06)
    Header checksum: 0xae4d (correct)
    Source: 129.79.6.123 (129.79.6.123)
    Destination: 65.119.106.232 (65.119.106.232)
Transmission Control Protocol, Src Port: 3499 (3499), Dst Port: 21
(21), Seq: 1416086370, Ack: 3644221912
    Source port: 3499 (3499)
    Destination port: 21 (21)
    Sequence number: 1416086370
    Next sequence number: 1416086870
    Acknowledgement number: 3644221912
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 18180
    Checksum: 0x2bef
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 14086936, tsecr 709017
File Transfer Protocol (FTP)
    Request: SITE
    Request Arg: EXEC
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaabbbbtÐÿÿ¿%.f%.f%.f%.f%.f%.f%.f%.f%.

   0  0000 0000 0001 0000 0c1f 385b 0800 4500   ..........8[..E.
  10  0228 4b1a 4000 2e06 ae4d xxxx xxxx xxxx   .(K.@....MA!c.Aw
  20  xxxx 0dab 0015 5467 c362 d936 69d8 8018   j.....Tg.b.6i...
  30  4704 2bef 0000 0101 080a 00d6 f318 000a   G.+.............
  40  d199 5349 5445 2045 5845 4320 6161 6161   ..SITE EXEC aaaa
  50  6161 6161 6161 6161 6161 6161 6161 6161   aaaaaaaaaaaaaaaa
  60  6161 6161 6161 6161 6161 6262 6262 74d0   aaaaaaaaaabbbbt.
  70  ffff bf25 2e66 252e 6625 2e66 252e 6625   ...%.f%.f%.f%.f%
```

```
80   2e66 252e 6625 2e66 252e 6625 2e66 252e    .f%.f%.f%.f%.f%.
```

Source of Trace
This trace came from a network that I have access to.

Detect was generated by
These alerts were generated by snort version 1.8.3 running with the full rule set available at www.snort.org. These alerts were generated by the following snort rules.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP EXPLOIT wu-ftpd
2.6.0 site exec format string overflow Linux"; content: "|31c031db
31c9b046 cd80 31c031db|"; flags: A+; reference:bugtraq,1387;
reference:cve,CAN-2000-0573; reference:arachnids,287;
classtype:attempted-admin; sid:344; rev:2;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP site exec"; cont
ent: "site exec"; nocase; flags: A+; reference:bugtraq,2241;
reference:arachnids
,317; classtype:bad-unknown; sid:361; rev:2;)
```

The second of these snort rules should be modified please refer to the defensive recommendations section as to why and how.

The raw packet data was gathered from a host outside the firewall running tcpdump and logging to a repository. I grabbed the tcpdump output after I found these alerts and wrote several filters and then loaded that into ethereal and saved the output to a text file because it is easier to read.

Probability the source address was spoofed
I don't believe that the source addresses were spoofed because in order for this attack to be successful a three-way handshake must take place.

Description of attack
This attack is targeted at FTP servers running Wu-Ftpd 2.6.0. Wu-Ftpd is a very common version of ftp that is shipped with many Linux distributions, and was developed by Washington University. Because of insufficient input string validation an attacker can execute arbitrary commands on the remote host as root. There are CVE, (CVE # 2000-0573) and cert advisories located at the following links.
http://www.cert.org/advisories/CA-2000-13.html
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0573

In this case the targeted host was running a vulnerable version of Wu-Ftpd and this attack was successful. The ftp server also allowed for anonymous login, which is the default setting when you build an ftp server. The attacker issued system commands on the host and actually copied the shadow password file off of the system. I know that this attack was indeed successful because of a packet I saw later in the dump that showed the

transfer of the shadow password file. This traffic is indicated in the following packets. If you look at the payload of the first packet you will see the command being issued: cat /etc/shadow. The second packet contains the contents of the shadow file including the user name root. Indicating that the command executed successfully. I have highlighted both of these payloads to make them easy to find.

```
Frame 57 (82 on wire, 82 captured)
    Arrival Time: Jan 29, 2002 00:39:37.0881
    Time delta from previous packet: 16.612387 seconds
    Time relative to first packet: 84.385699 seconds
    Frame Number: 57
    Packet Length: 82 bytes
    Capture Length: 82 bytes

File Transfer Protocol (FTP)
    Request: cat
    Request Arg: /etc/shadow

   0  0000 0000 0001 0000 0c1f 385b 0800 4500   ..........8[..E.
  10  0044 4b25 4000 2e06 b026 xxxx xxxx xxxx   .DK%@....&A!c.Aw
  20  xxxx xxxx 0015 5467 c55a d936 8a21 8018   j.....Tg.Z.6.!..
  30  87c0 2879 0000 0101 080a 00d7 1136 000a   ..(y.........6..
  40  e978 6361 7420 2f65 7463 2f73 6861 646f   .xcat /etc/shado
  50  770a                                     w.

Frame 58 (578 on wire, 144 captured)
    Arrival Time: Jan 29, 2002 00:39:37.0903
File Transfer Protocol (FTP)
    Response: root:$1$NIvhyxFd$uLKB.WM7t6.AYn5GOzC2M.:11715:0:99999:7:-
1:-1:134539268
    bin:*:

   0  0002 b319 e4c6 0000 0000 0000 0800 4510   ..............E.
  10  0234 3b36 4000 4006 ac15 xxxx xxxx xxxx   .4;6@.@...Awj.A!
  20  xxxx 0015 0dab d936 8a21 5467 c56a 8018   c......6.!Tg.j..
  30  7d78 5731 0000 0101 080a 000a f006 00d7   }xW1............
  40  1136 726f 6f74 3a24 3124 4e49 7668 7978   .6root:$1$NIvhyx
  50  4664 2475 4c4b 422e 574d 3774 362e 4159   Fd$uLKB.WM7t6.AY
  60  6e35 474f 7a43 324d 2e3a 3131 3731 353a   n5GOzC2M.:11715:
  70  303a 3939 3939 393a 373a 2d31 3a2d 313a   0:99999:7:-1:-1:
  80  3133 3435 3339 3236 380a 6269 6e3a 2a3a   134539268.bin:*:
```

Attack mechanism

With in the FTP service there is functionality called site exec that allows logged in users to execute a restricted subset of commands on the ftp server. The following explanation is from the cert advisory regarding this exploit.

> "The wu-ftpd "site exec" vulnerability is the result of missing character-formatting argument in several function calls that implement the "site exec" command functionality. Normally if "site exec" is enabled, a user logged into an ftp server (including the 'ftp' or 'anonymous' user) may execute a restricted subset of quoted commands on the server itself. However, if a malicious user can pass character format strings consisting of carefully constructed *printf() conversion

characters (%f, %p, %n, etc) while executing a "site exec" command, the ftp daemon may be tricked into executing arbitrary code as root."
http://www.cert.org/advisories/CA-2000-13.html

If you look at several of the packets that I have included you will see the use of the %f character as well as %x.

```
40  d199 5349 5445 2045 5845 4320 6161 6161   ..SITE EXEC aaaa
50  6161 6161 6161 6161 6161 6161 6161 6161   aaaaaaaaaaaaaaaa
60  6161 6161 6161 6161 6161 6262 6262 74d0   aaaaaaaaaabbbbt.
70  ffff bf25 2e66 252e 6625 2e66 252e 6625   ...%.f%.f%.f%.f%
80  2e66 252e 6625 2e66 252e 6625 2e66 252e   .f%.f%.f%.f%.f%.
40  cf70 5349 5445 2045 5845 4320 2578 2025   .pSITE EXEC %x %
50  7820 2578 2025 7820 2b25 7820 7c25 780a   x %x %x +%x |%x.
```

There is a very detailed description of this exploit containing the exact source code that creates the vulnerability is located at the following link.
http://www.securityfocus.com/archive/1/66544
In this trace the attacker has utilized this vulnerability to gain access to the /etc/shadow file. The compromised host was running RedHat 6.2 and was used as an FTP server for the company for which it belonged.
Why would an attacker want to use this exploit to gain access to the shadow file. Well there are many reasons first of which it will give him a list of all the user names on that system. With that information he has already solved half the problem into breaking into other systems. Most of the time there will be user names in that file still containing the default passwords. This would be the case when you have an admin who makes everyone a user account and tells him or her to be sure to change his or her passwords. Now how many times have you heard that and left your password as the default 'password'. Hopefully never, but there are many users who are not security aware and they find it easier to remember default or welcome than a difficult password including special characters and numbers. Just having a list of valid user names gives an attacker just that much more of an advantage. They can be used to brute force logon attempts, not a very stealthy way but some time effective.
The shadow file also contains the encrypted password for each user. Depending on the length and the complexity of the password they can be cracked within a reasonable amount of time.

Correlations

The honey net group detected the following alerts. These alerts were most likely triggered by similar activity.
http://project.honeynet.org/scans/scan19/scan/som6/timeline.xls

00:55:58.209849	[ALERT] FTP site exec	207.35.251.172:2243	192.168.1.102:21
00:55:58.372588	[ALERT] FTP site exec	207.35.251.172:2243	192.168.1.102:21

There is a brief discussion of this exploit at the following link.
http://www.sans.org/y2k/072100.htm
They are basically discussing how this vulnerability was around for a while, at least eight months before it was ever posted to bugtraq or CVE. The reason that they know this is

because the author of the code states, "WuFTPD: Providing *remote* root since at least1994"

Evidence of active targeting
This was definitely active targeting. This is an FTP exploit targeted at an FTP server. The attacker may have scanned the network prior to these alerts in order to determine which hosts were running ftp services.

Severity
Criticality = 5 This server is a part of the network infrastructure for target.net and is required for business purposes.

Lethality = 5 Any attack that gives an attacker root on one of my network devices is considered to be extremely lethal. On this case the attack was successful and the attacker gained information and maybe more.

System Countermeasures = 0 This system had no counter measures as it was exploited. The server was running a vulnerable version of ftpd and the attacker took advantage of that.

Network Countermeasure = 0 This server is in the DMZ and is allowed to be accessed from the internet. There are no ACL's in place or rules on the fire wall that would prevent this attack.

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

(5 + 5) – (0 + 0) = 10

This attack is extremely severe such proves this formula. There was a successful attack and if you look at the severity it is a 10. A 10 represents the highest severity possible.

Defensive recommendation
First off I would recommend rebuilding this server as it has been compromised you don't know what else may have happened to it. Unfortunately tripwire was not installed at the time so a file comparison couldn't be done to see what other files may have been accessed or modified. I would recommend in the future having tripwire installed on all production servers. Tripwire is available at www.tripwire.org I would also recommend upgrading the version of Wu-Ftpd to the latest version. Upgrades for the version of RedHat are available at the following link. ftp://updates.redhat.com/6.2/i386/wu-ftpd-2.6.0-14.6x.i386.rpm
Another thing to consider would be implementing access controls on the firewall to only allow specified IP addresses to access this FTP server. Since this attack is contingent upon being logged into the ftp server I would recommend disabling anonymous access.

As I stated earlier the second snort rule that is looking for the "site exec" string in the payload should be modified. The reason for this is due to the fact that that the attacker could put two spaces in the command and therefore bypass the IDS. I would recommend building a dynamic rule. By dynamic I mean using the regex option. "The regex option allows content options to specify wildcard options. The wildcards behave more like shell globbing than Perl-type regular expressions. A '*' in the content string, along with the regex modifier is interpreted to mean "any character, any number of times." http://www.snort.org/docs/writing_rules/chap2.html#tth_chAp2 This is from the Snort Users Guide written by Marty Roesch.
The following rule has been modified and I would recommend replacing the current snort rule with this one.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP site exec"; cont
ent: "site*exec"; regex; nocase; flags: A+; reference:bugtraq,2241;
reference:arachnids
,317; classtype:bad-unknown; sid:361; rev:2;)
```

Finally, ftp access can be restricted by using TCP wrappers.

Multiple Choice Question

The * symbol can be used with the regex option when writing a Snort rule to specify a range of characters. What symbol can be used to represent a single wildcard character?

A) %
B) +
C) ?
D) $

The answer is C.

Detect 2 – ShellCode x86 NOOP

Alert

Event Name	Detect Time	Target Address	Target Port	Source Address	Source Port
SHELLCODE x86 NOOP	2/1/02 00:32:34 PST	Target.net.106.227	514	Attacker.net.12.107	58289

Source of Trace
This trace came from a network that I have access to.

Detect was generated by
These alerts were generated by snort version 1.8.3 running with the full rule set available at www.snort.org. I didn't have access to the alert file at the time this alert was generated so I pulled it out of the mysql database that snort also logs to. This alert was generated by the following snort rule.
alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"SHELLCODE x86 NOOP"; content: "|90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; depth: 128; refer

ence:arachnids,181; classtype:shellcode-detect; sid:648; rev:4;)

I didn't see the string "90 90 90 90 90 90 90 90 90 90 90 90 90 90" in the payload of the packet that generated this alert because tcpdump wasn't logging the full payload. By default tcpdump only logs the first 68 bytes. In the future I will consider recommending increasing the snaplen but it will take up more storage space.
I didn't include the tcpdump output for this alert because I didn't have enough of the payload to make worthwhile. Basically what I saw were connection attempts to the targeted host on port 514 and the host responded with resets because it is not listening on the targeted port.

Probability the source address was spoofed
In this case the source address was probably not spoofed because the attacker is probably trying to execute commands on the target system that will allow him to gain access. In order for that to work he will need to complete the three way handshake which will not be possible if the source address is spoofed.

Description of the Attack
At first I thought this attack was targeted at the syslog port but if you recall syslog is UDP 514. In this case the attack is using the TCP protocol. After further research I discovered that there is an RPC Backdoor associated with TCP port 514. TCP port 514 is also associated with remote shell. I found this information using the ports database at www.snort.org
I did find a CVE reference to what may be the intent of this attack.
CAN-2001-0707
"** CANDIDATE (under review) ** Denicomp RSHD 2.18 and earlier allows a remote attacker to cause a denial of service (crash) via a long string to port 514."
http://cve.mitre.org/cgibin/cvekey.cgi?keyword=port+514
I do not believe this attack to have been successful because the targeted host was not listening on tcp 514 and sent a reset back to the attacker.

Attack Mechanism
This alert is triggering off the NOOP or no operation padding 9090 9090 bytes usually found in buffer overflows. Attackers use NOOPs to fill up the memory allocated to a certain application and when it fills up, the stack crashes and arbitrary code can be fed to the processor. In this case the attack is targeted at TCP port 514 so I researched vulnerabilities related to services that run on that port. There were not many references to exploits, especially buffer overflows, dealing with TCP port 514.
I would typically expect to see a payload similar to the following one where you can clearly see the padding and the shell code near the bottom of the payload. In this example the buffer overflow is attempting to execute /bin/sh after the stack is crashed.

```
  0  0060 0846 d018 0000 c577 9ab4 0800 4500   .`.F.....w....E.
 10  05a0 1633 0000 3011 59d9 d1b4 7198 0af2   ...3..0.Y...q...
 20  c702 0407 00b1 058c b87c 0001 0004 057d   .........|.....}
 30  0578 7f00 0001 0000 0000 0000 0000 9090   .x..............
 40  9090 9090 9090 9090 9090 9090 9090 9090   ................
 50  9090 95f7 ffbf 9090 9090 9090 9090 9090   ................
 60  9090 9090 9090 9090 9090 9090 9090 9090   ................
```

```
  70   9090 9090 9090 9090 9090 9090 9090 9090    ................
  80   9090 9090 9090 9090 9090 9090 9090 9090    ................
{SNIP}
 4f0   9090 9090 9090 9090 9090 9090 9090 9090    ................
 500   9090 9090 9090 9090 9090 9090 9090 9090    ................
 510   9090 9090 9090 9090 9090 9090 9090 9090    ................
 520   9090 9090 9090 9090 9089 e531 d2b2 6689    ...........1..f.
 530   d031 c989 cb43 895d f843 895d f44b 894d    .1...C.].C.].K.M
 540   fc8d 4df4 cd80 31c9 8945 f443 6689 5dec    ..M...1..E.Cf.].
 550   66c7 45ee 0f27 894d f08d 45ec 8945 f8c6    f.E..'.M..E..E..
 560   45fc 1089 d08d 4df4 cd80 89d0 4343 cd80    E.....M.....CC..
 570   89d0 43cd 8089 c331 c9b2 3f89 d0cd 8089    ..C....1..?.....
 580   d041 cd80 eb18 5e89 7508 31c0 8846 0789    .A....^.u.1..F..
 590   450c b00b 89f3 8d4d 088d 550c cd80 e8e3    E......M..U.....
       5a0  ffff ff2f 6269 6e2f 7368 0000 0000
            .../bin/sh....
```

The ports Database on www.Snort.org returned that the target port in this case is associated with an RPC backdoor but I didn't find any references to that on Google, Cert.org, or the SANS site.
I found the following information by searching Google and various other search engines.

I found that TCP rsh (remote shell) can send a command to a shell on the remote machine and receives the stderr and stdout from it. I also found the following explanation of a weakness in 4.2BSD.

> "4.2BSD provides a remote execution "server", which listens for TCP connection requests on port 514. When such a request arrives at a machine, the server checks that the originating host is "trusted" by comparing the source host ID in the IP header to a list of trusted computers. If the source host is OK, the server reads a user id and a command to execute from the virtual circuit TCP provides. The weakness in this scheme is that the source host itself fills in the IP source host id, and there is no provision in 4.2BSD or TCP/IP to discover the true origin of a packet."

I don't think that this is a valid explanation of what I have seen here. I have yet to find a buffer overflow related to the remote shell service.
I found the following warning on Xforce regarding rsh running on windows servers.

> "The Rsh service was detected as running. A version of rsh ships with the Windows NT Resource Kit, which executes all commands, regardless of user, under the system account. The system account is the most powerful account on a Windows NT computer, and we recommend not running this service under any circumstances. If this service is detected, use the instsrv tool, which also ships with the Windows NT Resource Kit, to remove the rsh service."
> http://www.iss.net/security_center/static/114.php

Correlations
I didn't find any correlations for traffic matching this alert. There are many references to the ShellCode x86 NOOP alert but not targeted at TCP port 514. Some of the references I found to the alert name are listed bellow.
http://lists.insecure.org/incidents/2001/Oct/0018.html

The traffic here clearly shows the padding in the payload of the packet. Some of the payload has been removed to save space.

```
44 24 18 2B F3 8B 08 03 CE 89 08 B8 01 00 00 00 D$.+............
5F 5E 5D 5B C3 90 90 90 90 90 90 90 90 90 90 90 _^][............
90 8B 44 24 04 8B 0D E0 41 44 00 3B C1 73 3F 8B ..D$....AD.;.s?.
C8 8B D0 C1 F9 05 83 E2 1F 8B 0C 8D E0 40 44 00 .............@D.
F6 44 D1 04 01 74 27 50 E8 54 2F 00 00 83 C4 04 .D...t'P.T/.....
50 FF 15 8C 65 44 00 85 C0 75 08 FF 15 F0 64 44 P...eD...u....dD
00 EB 02 33 C0 85 C0 74 12 A3 B4 26 44 00 C7 05 ...3...t...&D...
```

Sans also had reports of similar traffic.

```
Feb  5 15:33:56 hostka snort[23477]: IDS362 - MISC - Shellcode X86
NOPS-UDP:
  207.238.5.67:733 -> a.b.c.225:32772
```

http://www.sans.org/y2k/020901-1200.htm

Evidence of active Targeting
I can't tell if this was active targeting or part of a random attack. The targeted server was not listening on tcp port 514 or UDP port 514 so if this were active targeting it wouldn't make much sense.

Severity
Criticality = 5 This server is a part of the network infrastructure for target.net and is required for business purposes.

Lethality = 3 since this attack did not allow access to the system and at most could crash the rshell service I wouldn't consider this attack to be extremely lethal.

System Countermeasures = 5 This host was not listening on the targeted port therefore the attack could not have been successful.

Network Countermeasure = 1 This server is in the DMZ and is allowed to be accessed from the internet. There are no ACL's in place or rules on the firewall that would prevent this attack.

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

(5 + 3) – (5 + 1) = 2

Defensive Recommendations
I would recommend blocking access to port 514 at the perimeter firewall. Port 514 is associated with remote shell which will allow trusted IP's to execute commands on the

host running that service. Someone could spoof a trusted IP have access to that host. If this host were running the rshell service I would recommend turning it off. I would also recommend blocking all unnecessary ports at the firewall to help protect hosts on the internal and DMZ networks. The best way to do this is have a deny al rule and then allow only the necessary ports to be open.

Multiple Choice Question

Syslogd runs on UDP/514 what service typically runs on TCP/514
A) krshd
B) rshell
C) The tcp implementation of syslogd
D) Secure Syslog

The answer is B rshell or remote shell.

Detect 3 – RPC EXPLOIT statdx

Snort Alerts

```
[**] [1:600:1] RPC EXPLOIT statdx [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1]
01/27-21:25:13.487554 attacker.net.99.232:702 -> target.net.233.44:934
TCP TTL:46 TOS:0x0 ID:3503 IpLen:20 DgmLen:1132 DF
***AP*** Seq: 0x40BA711D  Ack: 0xC2C095A4  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 4291783 1863521
[Xref => http://www.whitehats.com/info/IDS442]
```

TCPDUMP Output For this alert

I ran the tcpdump output through ethereal and generated a text file because it is much easier to read. The following packet triggered the snort alert.

```
Frame 60 (1146 on wire, 144 captured)
    Arrival Time: Jan 27, 2002 21:25:13.487554000
    Time delta from previous packet: 0.030430000 seconds
    Time relative to first packet: 2023.984450000 seconds
    Frame Number: 60
    Packet Length: 1146 bytes
    Capture Length: 144 bytes
Internet Protocol, Src Addr: attacker.net.99.232 (attacker.net.99.232),
Dst Addr: target.net.233.44 (target.net.233.44)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 1132
    Identification: 0x0daf
    Flags: 0x04
        .1.. = Don't fragment: Set
```

```
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 46
    Protocol: TCP (0x06)
    Header checksum: 0xe974 (correct)
    Source: attacker.net.99.232 (attacker.net.99.232)
    Destination: target.net.233.44 (target.net.233.44)
Transmission Control Protocol, Src Port: 702 (702), Dst Port: 934
(934), Seq: 1085960477, Ack: 3267401124
    Source port: 702 (702)
    Destination port: 934 (934)
    Sequence number: 1085960477
    Next sequence number: 1085961557
    Acknowledgement number: 3267401124
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 5840
    Checksum: 0x024a
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 4291783, tsecr 1863521
Remote Procedure Call
    Last Fragment: Yes
    Fragment Length: 1076
    XID: 0x77dec70 (125693040)
    Message Type: Call (0)
    RPC Version: 2
    Program: STAT (100024)
    Program Version: 1
    Procedure: STAT (1)
    Credentials
        Flavor: AUTH_UNIX (1)
        Length: 32
        Stamp: 0x3c5429df
        Machine Name: localhost
            length: 9
            contents: localhost
            fill bytes: opaque data
        UID: 0
        GID: 0
        Auxiliary GIDs
    Verifier
        Flavor: AUTH_NULL (0)
        Length: 0
Network Status Monitor Protocol
    Program Version: 1
    Procedure: STAT (1)
[Short Frame: STAT]
```

```
00  00 00 00 00 00 01 00 00 0c 1f 38 5b 08 00 45 00   ..........8[..E.
10  04 6c 0d af 40 00 2e 06 e9 74 xx xx xx xx xx xx   .l..@....tA!c.Aw
20  6a e8 02 be 03 a6 40 ba 71 1d c2 c0 95 a4 80 18   j.....@.q.......
30  16 d0 02 4a 00 00 01 01 08 0a 00 41 7c c7 00 1c   ...J.......A|...
40  6f 61 80 00 04 34 07 7d ec 70 00 00 00 00 00 00   oa...4.}.p......
50  00 02 00 01 86 b8 00 00 00 01 00 00 00 01 00 00   ................
60  00 01 00 00 00 20 3c 54 29 df 00 00 00 09 6c 6f   ..... <T).....lo
70  63 61 6c 68 6f 73 74 00 00 00 00 00 00 00 00 00   calhost.........
80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Source of Trace
This trace came from a network that I have access to.

Detect was generated by
These alerts were generated by snort version 1.8.3 running with the full rule set available at www.snort.org. This alert was generated by the following snort rule.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"RPC EXPLOIT statdx";
flags: A+; content: "/bin|c74604|/sh";reference:arachnids,442;
classtype:attempted-admin; sid:600; rev:1;)
```

I didn't see the string `"/bin|c74604|/sh"` in the payload of the packet that generated this alert because tcpdump wasn't logging the full payload. By default tcpdump only logs the first 68 bytes. In the future I will consider recommending increasing the snaplen but it will take up more storage space.
The raw packet data was gathered from a host outside the firewall running tcpdump and logging to a repository. I grabbed the tcpdump output after I found these alerts and wrote several filters and then loaded that into ethereal and saved the output to a text file because it is easier to read.

Probability the source address was spoofed
I don't believe that the source addresses were spoofed because in order for this attack to be successful a three-way handshake must take place. The attacker is also looking to gain access to the system which would be impossible if he spoofed his source address.

Description of attack
The target of this attack was running a version of Linux called Slackware version 8. I do not believe that it is vulnerable to this exploit. In this case at least it did not appear that the attack was successful. The RPC statd exploit is a very common exploit and has even been used as part of larger worms and Trojans such as the Ramen worm that would deface websites with a picture of Ramen Noodles. It targets a known vulnerability in an RPC daemon called StatD. The purpose of statd is to implement the Network Status Monitor RPC protocol to provide reboot notification for other services such as the NFS service. This exploit was reported to Bugtraq on August 5, 2000, so it is now almost 2 years old. The CVE reference for this exploit is located at the following link.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0666

CVE states, “rpc.statd in the nfs-utils package in various Linux distributions does not properly cleanse untrusted format strings, which allows remote attackers to gain root privileges”

Attack Mechanism

One of the processes associated with rpc.statd passes logging information using the syslog() function. The format string that is passed is user supplied data and there is no bounds checking. Since there is no bounds checking this buffer can be overflowed, which in turn would place executable code into the process address space and overwrite the process return address, forcing the execution of what could and typically is malicious code.

As I stated earlier the packet that I collected that triggered this alert didn’t contain enough of the payload for me to show exactly what was happening. A friend of mine had a similar trace and he didn’t have the same storage restrictions that I have therefore he was able to retain the entire packet. I have included that here.

```
00   4500 0450 0171 0000 4011 cad9 xxxx xxxx      E..P.q..@.......
10   xxxx xxxx 039f 03a3 043c cc27 23fe 6f11      .........<.'#.o.
20   0000 0000 0000 0002 0001 86b8 0000 0001      ................
30   0000 0001 0000 0001 0000 0020 3999 8092      ............9...
40   0000 0009 6c6f 6361 6c68 6f73 7400 0000      ....localhost...
50   0000 0000 0000 0000 0000 0000 0000 0000      ................
60   0000 0000 0000 03e7 18f7 ffbf 18f7 ffbf      ................
70   19f7 ffbf 19f7 ffbf 1af7 ffbf 1af7 ffbf      ................
80   1bf7 ffbf 1bf7 ffbf 2538 7825 3878 2538      ........%8x%8x%8
90   7825 3878 2538 7825 3878 2538 7825 3878      x%8x%8x%8x%8x%8x
a0   2538 7825 3233 3678 256e 2531 3337 7825      %8x%236x%n%137x%
b0   6e25 3130 7825 6e25 3139 3278 256e 9090      n%10x%n%192x%n..
c0   9090 9090 9090 9090 9090 9090 9090 9090      ................
0d0   9090 9090 9090 9090 9090 9090 9090 9090       ................
0e0   9090 9090 9090 9090 9090 9090 9090 9090       ................
<SNIP>
380   9090 9090 9090 9090 9090 9090 9090 9090       ................
390   9090 9090 9090 9090 9090 9090 9090 9090       ................
3a0   9090 9090 9090 9090 9090 9090 9090 9090       ................
3b0   9090 9090 9090 9090 9090 9090 9090 9090       ................
3c0   9090 9090 9090 9090 9090 31c0 eb7c 5989       ..........1..|Y.
3d0   4110 8941 08fe c089 4104 89c3 fec0 8901       A..A....A.......
3e0   b066 cd80 b302 8959 0cc6 410e 99c6 4108       .f.....Y..A...A.
3f0   1089 4904 8041 040c 8801 b066 cd80 b304       ..I..A.....f....
400   b066 cd80 b305 30c0 8841 04b0 66cd 8089       .f....0..A..f...
410   ce88 c331 c9b0 3fcd 80fe c1b0 3fcd 80fe       ...1..?.....?...
420   c1b0 3fcd 80c7 062f 6269 6ec7 4604 2f73       ..?..../bin.F./s
430   6841 30c0 8846 0789 760c 8d56 108d 4e0c       hA0..F..v..V..N.
440   89f3 b00b cd80 b001 cd80 e87f ffff ff00       ................
```

If you look at the sections of the payload that have been bolded you will see that the beginning of the payload is very similar to the packet I received. Near the end of the payload you will see bin/sh. If you notice the 9090 bytes, those are used to fill up the buffer. In other words those bytes do nothing but fill up the buffer so that the stack can be smashed and the malicious code i.e. bin/sh, can be executed. 9090 represents a NOOP or

a no operation, it indicates no operation to be executed. This means that they are not instructing the processor to do anything.

Correlations
I found many correlations for traffic similar to this. Once this exploit was posted to Bugtraq there were increased scans looking for systems with port 111 or rpcinfo listening so the attacker could find out what port statd was running on.
Laurie Zirkle reported the following traffic to incidents.org

```
Dec  6 05:21:36 hosty snort: [ID 702911 local0.alert] [1:583:2] RPC
portmap request rstatd [Classification: Decode of an RPC Query]
[Priority: 2]: {UDP} 194.251.105.187:854 -> z.y.x.34:111
Dec  6 05:21:36 hostj snort: RPC portmap request rstatd
[Classification: Attempted Information Leak   Priority: 3]:
194.251.105.187:855 -> z.y.x.66:111
Dec  6 05:21:36 hostmi snort: [ID 702911 auth.alert] [1:1282:1] RPC
EXPLOIT statdx [Classification: Attempted Administrator Privilege Gain]
[Priority: 1]: {UDP} 194.251.105.187:857 -> z.y.x.98:32777
Dec  6 05:21:36 hostmi snort: [ID 702911 auth.alert] [1:583:2] RPC
portmap request rstatd [Classification: Decode of an RPC Query]
[Priority: 2]: {UDP} 194.251.105.187:856 -> z.y.x.98:111
```
http://www.incidents.org/archives/intrusions/msg02798.html

That wasn't the only report to incident.org. I also found the following traffic again reported by Laurie Zirkle.

```
Dec 13 11:30:25 hosty snort: [ID 702911 local0.alert] [1:583:2] RPC
portmap request rstatd [Classification: Decode of an RPC Query]
[Priority: 2]: {UDP} 150.254.230.137:979 -> z.y.x.34:111
Dec 13 11:30:26 hosty snort: [ID 702911 local0.alert] [1:1282:1] RPC
EXPLOIT statdx [Classification: Attempted Administrator Privilege Gain]
[Priority: 1]: {UDP} 150.254.230.137:980 -> z.y.x.34:32777
Dec 13 11:31:16 hoste portsentry[22361]: attackalert: Connect from
host: 150.254.230.137/150.254.230.137 to TCP port: 111
Dec 13 17:23:15 150.254.230.137:2109 -> a.b.c.27:111 SYN ******S*
Dec 13 17:23:15 150.254.230.137:2115 -> a.b.c.33:111 SYN ******S*
Dec 13 17:23:18 150.254.230.137:2133 -> a.b.c.51:111 SYN ******S*
Dec 13 17:23:18 150.254.230.137:2144 -> a.b.c.62:111 SYN ******S*
Dec 13 17:23:21 150.254.230.137:685 -> a.b.c.62:111 UDP
Dec 13 17:23:18 150.254.230.137:2153 -> a.b.c.71:111 SYN ******S*
Dec 13 17:23:18 150.254.230.137:2164 -> a.b.c.82:111 SYN ******S*
Dec 13 17:23:18 150.254.230.137:2183 -> a.b.c.101:111 SYN ******S*
Dec 13 17:23:18 150.254.230.137:2188 -> a.b.c.106:111 SYN ******S*
Dec 13 17:23:18 150.254.230.137:2193 -> a.b.c.111:111 SYN ******S*
Dec 13 17:23:18 150.254.230.137:2210 -> a.b.c.128:111 SYN ******S*
Dec 13 17:23:18 150.254.230.137:2220 -> a.b.c.138:111 SYN ******S*
Dec 13 17:23:18 150.254.230.137:2259 -> a.b.c.177:111 SYN ******S*
```
http://www.incidents.org/archives/intrusions/msg02909.html

Evidence of Active targeting
This alert shows many signs of active targeting. I say this because after looking into my snort logs for other events from this source address or targeted at this destination address I found the following alerts.

```
[**] [1:596:2] RPC portmap listing [**]
[Classification: Decode of an RPC Query] [Priority: 2]
01/27-20:55:24.312486 attacker.net.99.232:939 -> target.net.233.44:111
TCP TTL:46 TOS:0x0 ID:42709 IpLen:20 DgmLen:96 DF
***AP*** Seq: 0xD00567AF  Ack: 0x51C032C3  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 4112884 1684510
[Xref => http://www.whitehats.com/info/IDS429]
```

This alert shows the attacker requesting rpc port map info on my host. He would have done this in order to find out what port the rpc.statd service was running on. Then an hour later he decided to run a statd buffer flow targeted at a host he knew was running the exploitable service.

Severity
Criticality = 5 This server is a part of the network infrastructure for target.net and is required for business purposes.

Lethality = 5 Any attack that gives an attacker root on one of my network devices is considered to be extremely lethal.

System Countermeasures = 5 The version of Slackware that was running on the target of this attack is not vulnerable to this exploit.

Network Countermeasure = 1 This server is in the DMZ and is allowed to be accessed from the internet. There are no ACL's in place or rules on the firewall that would prevent this attack.

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

(5 + 5) – (5 + 1) = 4

Defensive Recommendations
I would recommend not running rpc services on hosts that are accessible from the Internet. Since portmapper-managed ports are dynamically assigned, it is difficult to firewall individual ports and may be more feasible to "deny all unless specifically allowed", at least on ports less than 1024. I would also recommend that any servers running rpc services are using a version that is not vulnerable to rpc exploits. I would like to refer you to an excellent paper that describes the uses, the dangers, and how to protect your self from vulnerabilities related to RPC. I would recommend implementing the suggestions made in the What To Do section of this paper.
http://www.sans.org/newlook/resources/IDFAQ/blocking.htm

Multiple Choice Question
What is the default snaplen of TCPDUMP?
A) 48

B) 72
C) 32
D) 68

The answer is D. 68 "`Snarf snaplen bytes of data from each packet rather` than the default of 68" http://www.tcpdump.org/tcpdump_man.html

Detect 4 – EXPLOIT LPRng overflow

Alert

Event Name	Protocol	Detect Time	Target Address	Target Port	Source Address	Source Port
EXPLOIT LPRng overflow	TCP	1/29/02 23:58:43 PST	target.net.106.217	515	Attacker.net.12.33	50418

Source of Trace
This trace came from a network that I have access to.

Detect was generated by
These alerts were generated by snort version 1.8.3 running with the full rule set available at www.snort.org. This alert was generated by the following snort rule.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 515 (msg:"EXPLOIT LPRng o
verflow"; flags: A+; content: "|43 07 89 5B 08 8D 4B 08 89 43 0C B0 0B
CD 80 31 C0 FE C0 CD 80 E8 94 FF FF FF 2F 62 69 6E 2F 73 68 0A|";
reference:bugtraq,1712;
 classtype:attempted-admin; sid:301; rev:1;)
```

I didn't include the tcpdump output for this alert because I didn't have enough of the payload to make worthwhile. Basically what I saw were connection attempts to the targeted host on port 515 and the host responded with resets because it was not listening on the targeted port.

Probability the source address was spoofed
I don't believe that the source addresses were spoofed the packets contained no signs of spoofing. A blind spoof would defeat the purpose of this attack.

Description of attack
This attack is looking to exploit a vulnerability found in the LPrng service of some Linux distributions. The LPrng service is a printer daemon that runs on port 515 TCP/UDP. LPRng is a print spooling system that was designed to mimic the BSD line printer service. LPrng will print a document with little or no knowledge of its contents and no special processing is required to print on a local machine or in a distributed printing environment. Within the code there is a format string vulnerability that could allow an attacker to execute arbitrary code.
The CVE reference is listed bellow.

CVE-2000-0917 "Format string vulnerability in use_syslog() function in LPRng 3.6.24 allows remote attackers to execute arbitrary commands." http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0917

Attack Mechanism

According to The cert advisory http://www.kb.cert.org/vuls/id/382365 LPrng

> "has a missing format string argument in at least two calls to the syslog() function. Missing format strings in function calls which allow user-supplied arguments to be passed to a susceptible *snprintf() function call may allow remote users with access to the printer port (port 515/tcp) to pass format-string parameters that can overwrite arbitrary addresses in the printing service's address space. Such overwriting can cause segmentation violations leading to denial of printing services or lead to the execution of arbitrary code injected through other means into the memory segments of the printer service."

The attack mechanism is a classic buffer overflow exploit. This attack works by sending packets padded with NOOPs in order to "smash the stack" or overflow the memory buffer. Once the buffer is overflowed, /bin/sh is passed to the processor and executed as if it were part of the LPrng service. There is sample code available from http://www.rdcrew.com.ar/files/rdC-LPRng.c If we examine a snip of the exploit we will see the use of shell code.

```
char shellcode[]= // not
mine"\x31\xc0\x31\xdb\x31\xc9\xb3\x07\xeb\x67\x5f\x8d\
x4f"
"\x07\x8d\x51\x0c\x89\x51\x04\x8d\x51\x1c\x89\x51\x08"
"\x89\x41\x1c\x31\xd2\x89\x11\x31\xc0\xc6\x41\x1c\x10"
"\xb0\x66\xcd\x80\xfe\xc0\x80\x79\x0c\x02\x75\x04\x3c"
"\x01\x74\x0d\xfe\xc2\x80\xfa\x01\x7d\xe1\x31\xc0\xfe"
"\xc0\xcd\x80\x89\xd3\x31\xc9\x31\xc0\xb0\x3f\xcd\x80"
"\xfe\xc1\x80\xf9\x03\x75\xf3\x89\xfb\x31\xc0\x31\xd2"
"\x88\x43\x07\x89\x5b\x08\x8d\x4b\x08\x89\x43\x0c\xb0"
"\x0b\xcd\x80\x31\xc0\xfe\xc0\xcd\x80\xe8\x94\xff\xff"
"\xff\x2f\x62\x69\x6e\x2f\x73\x68";
```

The following piece of code is where the vulnerability lies in LPrng.

```
LPRng-3.6.24/src/common/errormsg.c, use_syslog()
---
static void use_syslog(int kind, char *msg)
[...]
# ifdef HAVE_OPENLOG
        /* use the openlog facility */
        openlog(Name, LOG_PID | LOG_NOWAIT, SYSLOG_FACILITY );
        syslog(kind, msg);
        closelog();

# else
    (void) syslog(SYSLOG_FACILITY | kind, msg);
# endif                                                  /*
```

```
HAVE_OPENLOG */
[...]
```

There is also an excellent article which discuses this vulnerability on the SANS site. http://rr.sans.org/malicious/ramen.php The paper is discussing the Ramen worm but it discusses this vulnerability as well since it is one of the vulnerabilities Ramen looks to exploit.

Correlations

Kathy Bergsma reported scans for port 515 to insecure.org at the handler notes are available at the following link.
http://lists.insecure.org/incidents/2000/Dec/0006.html

Some Port 515 activity can be attributed to the Ramen worm as described by Matt Fearnow at the following SANS link. The Ramen worm will try to exploit remote systems using this LPrng vulnerability.
http://www.sans.org/y2k/012201.htm

Evidence of Active Targeting

If this host were running LPrng printer services I would think that this would be an example of active targeting. In this case the server was not and therefore I wouldn't expect this exploit to be targeted at this host. I have also never seen any activity from the source address of this attack targeted at target.net. If an attacker had scanned the network before they would have known that this server was not running LPrng and would probably not have chosen this as an attack method.

Defensive Recommendations

First off I would recommend not allowing access to internal servers from the Internet. I don't believe that there is any reason that would justify having port 515 accessible from the net even though this box is in the DMZ. If this server were running LPrng services I would recommend upgrading to version 3.6.24-2 or later for RedHat. There is a complete list for other operating systems at the following link. http://www.cert.org/advisories/CA-2000-22.html

Severity

Criticality = 5 This server is a part of the network infrastructure for target.net and is required for business purposes.

Lethality = 5 Any attack that could potentially give an attacker root is considered to be extremely lethal.

System Countermeasures = 5 LPrng was not running on the targeted system.

Network Countermeasure = 1 This server is in the DMZ and is allowed to be accessed from the internet. There are no ACL's in place or rules on the firewall that would prevent this attack.

(Criticality + Lethality) –

(System Countermeasures + Network Countermeasures) = Severity

(5 + 5) – (5 + 1) = 4

Multiple Choice Question
The following logs showing a scan for the LPrng port were generated by?

```
Jan 29 17:50:54 host1 snort: [1:0:0] TCP to 515 lpr {TCP}
 X.X.186.238:1837 -> X.X.140.157:515
Jan 29 17:50:54 host1 snort: [1:0:0] TCP to 515 lpr {TCP}
 X.X.186.238:1837 -> X.X.140.157:515
```

A) IP Tables
B) TcpDump
C) Gauntlet
D) IP Chains

The answer id D. IP Chains

Detect 5 - EXPLOIT ssh CRC32 overflow

Snort Alerts

```
[**] [1:1327:1] EXPLOIT ssh CRC32 overflow [**]
[Classification: Executable code was detected] [Priority: 1]
01/21-14:53:54.236191 attacker.net.55.4:852 -> target.net.106.224:22
TCP TTL:46 TOS:0x0 ID:13491 IpLen:20 DgmLen:1500 DF
***AP*** Seq: 0xEF8A007B  Ack: 0x8A3AEB88  Win: 0x7D78  TcpLen: 32
TCP Options (3) => NOP NOP TS: 131591045 173629903
[Xref => http://www.securityfocus.com/bid/2347]
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0144]
```

TCPDUMP Output from these alerts
The following packet generated this alert. I have also included the response from the targeted system.

```
Frame 13 (1514 on wire, 1400 captured)
    Arrival Time: Jan 21, 2002 14:53:54.236191000
    Time delta from previous packet: 0.040135000 seconds
    Time relative to first packet: 4.473671000 seconds
    Frame Number: 13
    Packet Length: 1514 bytes
    Capture Length: 1400 bytes
Internet Protocol, Src Addr: attacker.net.55.4 (attacker.net.55.4), Dst
Addr: target.net.106.224 (target.net.106.224)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
```

```
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 1500
    Identification: 0x34b3
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 46
    Protocol: TCP (0x06)
    Header checksum: 0x1627 (correct)
    Source: attacker.net.55.4 (attacker.net.55.4)
    Destination: target.net.106.224 (target.net.106.224)
Transmission Control Protocol, Src Port: 852 (852), Dst Port: 22 (22),
Seq: 4018798715, Ack: 2319117192
    Source port: 852 (852)
    Destination port: 22 (22)
    Sequence number: 4018798715
    Next sequence number: 4018800163
    Acknowledgement number: 2319117192
    Header length: 32 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 32120
    Checksum: 0x609d
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 131591045, tsecr 173629903
iSCSI (NOP Out)
    Opcode: NOP Out (0x00)
    0... .... = X: Not retry
    .0.. .... = I: Queued delivery
    Flags: 0x01
        0... .... = P: No poll requested
    DataSegmentLength: 0x00001800
    LUN: FFFFFFFF00001801
    InitiatorTaskTag: 0xffffffff
    TargetTransferTag: 0x00001804
    CmdSN: 0xffffffff
    ExpStatSN: 0x00001805
    BufferOffset: 0xffffffff
    Payload: FFFFFFFF0000180CFFFFFFFF0000180D...

00  00 00 00 00 00 01 00 00 0c 1f 38 5b 08 00 45 00   ..........8[..E.
10  05 dc 34 b3 40 00 2e 06 16 27 xx xx xx xx xx xx   ..4.@....'..7.Aw
20  xx xx 03 54 00 16 ef 8a 00 7b 8a 3a eb 88 80 18   j..T.....{.:....
30  7d 78 60 9d 00 00 01 01 08 0a 07 d7 eb 85 0a 59   }x`............Y
40  61 cf 00 01 57 00 00 00 18 00 ff ff ff ff 00 00   a...W...........
50  18 01 ff ff ff ff 00 00 18 04 ff ff ff ff 00 00   ................
```

```
60  18 05 ff ff ff ff 00 00 18 08 ff ff ff ff 00 00   ................
70  18 09 ff ff ff ff 00 00 18 0c ff ff ff ff 00 00   ................
80  18 0d ff ff ff ff 00 00 18 10 ff ff ff ff 00 00   ................
90  18 11 ff ff ff ff 00 00 18 14 ff ff ff ff 00 00   ................
a0  18 15 ff ff ff ff 00 00 18 18 ff ff ff ff 00 00   ................
b0  18 19 ff ff ff ff 00 00 18 1c ff ff ff ff 00 00   ................
c0  18 1d ff ff ff ff 00 00 18 20 ff ff ff ff 00 00   ..........  ....
d0  18 21 ff ff ff ff 00 00 18 24 ff ff ff ff 00 00   .!.......$......
e0  18 25 ff ff ff ff 00 00 18 28 ff ff ff ff 00 00   .%.......(......
f0  18 29 ff ff ff ff 00 00 18 2c ff ff ff ff 00 00   .).......,......
100  18 2d ff ff ff ff 00 00 18 30 ff ff ff ff 00 00   .-.......0......
110  18 31 ff ff ff ff 00 00 18 34 ff ff ff ff 00 00   .1.......4......
120  18 35 ff ff ff ff 00 00 18 38 ff ff ff ff 00 00   .5.......8......
130  18 39 ff ff ff ff 00 00 18 3c ff ff ff ff 00 00   .9.......<......
140  18 3d ff ff ff ff 00 00 18 40 ff ff ff ff 00 00   .=.......@......
150  18 41 ff ff ff ff 00 00 18 44 ff ff ff ff 00 00   .A.......D......
160  18 45 ff ff ff ff 00 00 18 48 ff ff ff ff 00 00   .E.......H......
170  18 49 ff ff ff ff 00 00 18 4c ff ff ff ff 00 00   .I.......L......
180  18 4d ff ff ff ff 00 00 18 50 ff ff ff ff 00 00   .M.......P......
190  18 51 ff ff ff ff 00 00 18 54 ff ff ff ff 00 00   .Q.......T......
1a0  18 55 ff ff ff ff 00 00 18 58 ff ff ff ff 00 00   .U.......X......
1b0  18 59 ff ff ff ff 00 00 18 5c ff ff ff ff 00 00   .Y.......\......
1c0  18 5d ff ff ff ff 00 00 18 60 ff ff ff ff 00 00   .].......`......
1d0  18 61 ff ff ff ff 00 00 18 64 ff ff ff ff 00 00   .a.......d......
1e0  18 65 ff ff ff ff 00 00 18 68 ff ff ff ff 00 00   .e.......h......
1f0  18 69 ff ff ff ff 00 00 18 6c ff ff ff ff 00 00   .i.......l......
200  18 6d ff ff ff ff 00 00 18 70 ff ff ff ff 00 00   .m.......p......
210  18 71 ff ff ff ff 00 00 18 74 ff ff ff ff 00 00   .q.......t......
220  18 75 ff ff ff ff 00 00 18 78 ff ff ff ff 00 00   .u.......x......
230  18 79 ff ff ff ff 00 00 18 7c ff ff ff ff 00 00   .y.......|......
240  18 7d ff ff ff ff 00 00 18 80 ff ff ff ff 00 00   .}..............
250  18 81 ff ff ff ff 00 00 18 84 ff ff ff ff 00 00   ................
260  18 85 ff ff ff ff 00 00 18 88 ff ff ff ff 00 00   ................
270  18 89 ff ff ff ff 00 00 18 8c ff ff ff ff 00 00   ................
280  18 8d ff ff ff ff 00 00 18 90 ff ff ff ff 00 00   ................
290  18 91 ff ff ff ff 00 00 18 94 ff ff ff ff 00 00   ................
2a0  18 95 ff ff ff ff 00 00 18 98 ff ff ff ff 00 00   ................
2b0  18 99 ff ff ff ff 00 00 18 9c ff ff ff ff 00 00   ................
2c0  18 9d ff ff ff ff 00 00 18 a0 ff ff ff ff 00 00   ................
2d0  18 a1 ff ff ff ff 00 00 18 a4 ff ff ff ff 00 00   ................
2e0  18 a5 ff ff ff ff 00 00 18 a8 ff ff ff ff 00 00   ................
2f0  18 a9 ff ff ff ff 00 00 18 ac ff ff ff ff 00 00   ................
300  18 ad ff ff ff ff 00 00 18 b0 ff ff ff ff 00 00   ................
310  18 b1 ff ff ff ff 00 00 18 b4 ff ff ff ff 00 00   ................
320  18 b5 ff ff ff ff 00 00 18 b8 ff ff ff ff 00 00   ................
330  18 b9 ff ff ff ff 00 00 18 bc ff ff ff ff 00 00   ................
340  18 bd ff ff ff ff 00 00 18 c0 ff ff ff ff 00 00   ................
350  18 c1 ff ff ff ff 00 00 18 c4 ff ff ff ff 00 00   ................
360  18 c5 ff ff ff ff 00 00 18 c8 ff ff ff ff 00 00   ................
370  18 c9 ff ff ff ff 00 00 18 cc ff ff ff ff 00 00   ................
380  18 cd ff ff ff ff 00 00 18 d0 ff ff ff ff 00 00   ................
390  18 d1 ff ff ff ff 00 00 18 d4 ff ff ff ff 00 00   ................
3a0  18 d5 ff ff ff ff 00 00 18 d8 ff ff ff ff 00 00   ................
3b0  18 d9 ff ff ff ff 00 00 18 dc ff ff ff ff 00 00   ................
3c0  18 dd ff ff ff ff 00 00 18 e0 ff ff ff ff 00 00   ................
3d0  18 e1 ff ff ff ff 00 00 18 e4 ff ff ff ff 00 00   ................
3e0  18 e5 ff ff ff ff 00 00 18 e8 ff ff ff ff 00 00   ................
```

```
3f0  18 e9 ff ff ff ff 00 00 18 ec ff ff ff ff 00 00   ................
400  18 ed ff ff ff ff 00 00 18 f0 ff ff ff ff 00 00   ................
410  18 f1 ff ff ff ff 00 00 18 f4 ff ff ff ff 00 00   ................
420  18 f5 ff ff ff ff 00 00 18 f8 ff ff ff ff 00 00   ................
430  18 f9 ff ff ff ff 00 00 18 fc ff ff ff ff 00 00   ................
440  18 fd ff ff ff ff 00 00 19 00 ff ff ff ff 00 00   ................
450  19 01 ff ff ff ff 00 00 19 04 ff ff ff ff 00 00   ................
460  19 05 ff ff ff ff 00 00 19 08 ff ff ff ff 00 00   ................
470  19 09 ff ff ff ff 00 00 19 0c ff ff ff ff 00 00   ................
480  19 0d ff ff ff ff 00 00 19 10 ff ff ff ff 00 00   ................
490  19 11 ff ff ff ff 00 00 19 14 ff ff ff ff 00 00   ................
4a0  19 15 ff ff ff ff 00 00 19 18 ff ff ff ff 00 00   ................
4b0  19 19 ff ff ff ff 00 00 19 1c ff ff ff ff 00 00   ................
4c0  19 1d ff ff ff ff 00 00 19 20 ff ff ff ff 00 00   ......... ......
4d0  19 21 ff ff ff ff 00 00 19 24 ff ff ff ff 00 00   .!.......$......
4e0  19 25 ff ff ff ff 00 00 19 28 ff ff ff ff 00 00   .%.......(......
4f0  19 29 ff ff ff ff 00 00 19 2c ff ff ff ff 00 00   .).......,......
500  19 2d ff ff ff ff 00 00 19 30 ff ff ff ff 00 00   .-.......0......
510  19 31 ff ff ff ff 00 00 19 34 ff ff ff ff 00 00   .1.......4......
520  19 35 ff ff ff ff 00 00 19 38 ff ff ff ff 00 00   .5.......8......
530  19 39 ff ff ff ff 00 00 19 3c ff ff ff ff 00 00   .9.......<......
540  19 3d ff ff ff ff 00 00 19 40 ff ff ff ff 00 00   .=.......@......
550  19 41 ff ff ff ff 00 00 19 44 ff ff ff ff 00 00   .A.......D......
560  19 45 ff ff ff ff 00 00 19 48 ff ff ff ff 00 00   .E.......H......
570  19 49 ff ff ff ff 00 00                           .I......
```

The Response

```
Frame 14 (66 on wire, 66 captured)
    Arrival Time: Jan 21, 2002 14:53:54.236785000
    Time delta from previous packet: 0.000594000 seconds
    Time relative to first packet: 4.474265000 seconds
    Frame Number: 14
    Packet Length: 66 bytes
    Capture Length: 66 bytes
Internet Protocol, Src Addr: target.net.106.224 (target.net.106.224),
Dst Addr: attacker.net.55.4 (attacker.net.55.4)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 52
    Identification: 0x0000
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0x3e82 (correct)
    Source: target.net.106.226 (target.net.106.226)
    Destination: attacker.net.55.4 (attacker.net.55.4)
Transmission Control Protocol, Src Port: 22 (22), Dst Port: 852 (852),
Seq: 2319117192, Ack: 4018800163
    Source port: 22 (22)
```

```
    Destination port: 852 (852)
    Sequence number: 2319117192
    Acknowledgement number: 4018800163
    Header length: 32 bytes
    Flags: 0x0010 (ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 0... = Push: Not set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 8688
    Checksum: 0x89b5 (correct)
    Options: (12 bytes)
        NOP
        NOP
        Time stamp: tsval 173630020, tsecr 131591045

00  00 60 08 16 93 25 00 00 00 00 00 00 08 00 45 00   .`...%........E.
10  00 34 00 00 40 00 40 06 3e 82 xx xx 6a e2 xx xx   .4..@.@.>.Awj...
20  37 04 00 16 03 54 8a 3a eb 88 ef 8a 06 23 80 10   7....T.:.....#..
30  21 f0 89 b5 00 00 01 01 08 0a 0a 59 62 44 07 d7   !..........YbD..
   40  eb 85
```

The following logs were found in /var/log/messages of the attacked host. Note the time difference is because the system clocks were not in sync. NTP would be very useful for this purpose because when doing analysis detect times are very critical.

```
Jan 21 15:04:20 Natasha PAM_pwdb[2538]: (sshd) session opened for user
root by (uid=0)
Jan 21 15:05:00 Natasha sshd[2510]: fatal: Timeout before
authentication for 24.229.55.4.
Jan 21 15:10:40 Natasha sshd[2526]: fatal: Timeout before
authentication for 24.229.55.4.
Jan 21 15:39:41 Natasha sshd[565]: Generating new 768 bit RSA key.
Jan 21 15:39:42 Natasha sshd[565]: RSA key generation complete.
Jan 21 15:44:20 Natasha sshd[2699]: Disconnecting: Corrupted check
bytes on input.
Jan 21 15:46:26 Natasha sshd[2700]: Disconnecting: Corrupted check
bytes on input.
Jan 21 16:39:42 Natasha sshd[565]: Generating new 768 bit RSA key.
Jan 21 16:39:43 Natasha sshd[565]: RSA key generation complete.
Jan 21 16:40:34 Natasha sshd[2882]: Disconnecting: crc32 compensation
attack: network attack detected
Jan 21 16:41:05 Natasha sshd[2883]: Disconnecting: crc32 compensation
attack: network attack detected
Jan 21 16:42:52 Natasha sshd[2890]: Disconnecting: crc32 compensation
attack: network attack detected
Jan 21 16:43:30 Natasha sshd[2893]: Disconnecting: crc32 compensation
attack: network attack detected
<SNIP>
Jan 21 16:54:23 Natasha sshd[2924]: Disconnecting: crc32 compensation
attack: network attack detected
```

```
Jan 21 16:54:53 Natasha sshd[2925]: Disconnecting: crc32 compensation
attack: network attack detected
Jan 21 16:56:38 Natasha sshd[2927]: Disconnecting: crc32 compensation
attack: network attack detected
Jan 21 16:57:27 Natasha sshd[2932]: Disconnecting: crc32 compensation
attack: network attack detected
Jan 21 16:57:49 Natasha sshd[2933]: Disconnecting: crc32 compensation
attack: network attack detected
Jan 21 16:59:00 Natasha sshd[2934]: Disconnecting: crc32 compensation
attack: network attack detected
Jan 21 17:02:18 Natasha sshd[2940]: Disconnecting: Corrupted check
bytes on input.
Jan 21 17:02:47 Natasha sshd[2937]: Disconnecting: crc32 compensation
attack: network attack detected
```

Source of Trace
This trace came from a network that I have access to.

Detect was generated by
These alerts were generated by snort version 1.8.3 running with the full rule set available at www.snort.org. This alert was generated by the following snort rule.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"EXPLOIT ssh CRC3
2 overflow"; flags:A+; content:"|00 01 57 00 00 00 18|"; offset:0;
depth:7; cont
ent:"|FF FF FF FF 00 00|"; offset:8; depth:14; reference:bugtraq,2347;
reference
:cve,CVE-2001-0144; classtype:shellcode-detect; sid:1327; rev:1;)
```

The raw packet data was gathered from a host outside the firewall running tcpdump and logging to a repository. I grabbed the tcpdump output after I found these alerts and wrote several filters and then loaded that into ethereal and saved the output to a text file because it is easier to read.

Probability the source address was spoofed
I don't believe that the source addresses were spoofed because in order for this attack to be successful a three-way handshake must take place. Another reason the source address is most likely not spoofed is that the attacker is attempting to issue commands on the system that will allow him access. An ideal command would be bin/sh that could enable the attacker to gain a remote shell. If he spoofed his source address the shell would never be returned to him.

Attack Description
This attack was target at a sever in the DMZ that was running a vulnerable version of SSH. I believe that this attack may have been successful because of the messages I saw in /var/messages. The following cert advisory states, "In reports received by the CERT/CC, systems compromised via this vulnerability have exhibited the following pattern in system log messages:

```
hostname sshd[xxx]: Disconnecting: Corrupted check bytes on
input.
hostname sshd[xxx]: Disconnecting: crc32 compensation attack:
network attack detected
hostname sshd[xxx]: Disconnecting: crc32 compensation attack:
network attack detected"
```
http://www.cert.org/advisories/CA-2001-35.html

These logs look very similar to the logs I found on my host.

```
Jan 21 15:44:20 Natasha sshd[2699]: Disconnecting: Corrupted
check bytes on input.
Jan 21 15:46:26 Natasha sshd[2700]: Disconnecting: Corrupted
check bytes on input.
Jan 21 16:39:42 Natasha sshd[565]: Generating new 768 bit RSA
key.
Jan 21 16:39:43 Natasha sshd[565]: RSA key generation complete.
Jan 21 16:40:34 Natasha sshd[2882]: Disconnecting: crc32
compensation attack: network attack detected
      Jan 21 16:41:05 Natasha sshd[2883]: Disconnecting: crc32
      compensation
```

The CVE reference for this vulnerability is CVE-2001-0144 and is available at http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=EXPLOIT+ssh+CRC32+overflow+ CVE states "CORE SDI SSH1 CRC-32 compensation attack detector allows remote attackers to execute arbitrary commands on an SSH server or client via an integer overflow."

Attack Mechanism

SSH is a widely used client-server application for authentication and encryption of network communications it is a more secure replacement for applications such as telnet and rlogin. In 1998 a design flaw in the SSH1 protocol was discovered that could lead an attacker to inject malicious packets into an SSH encrypted stream that would allow execution of arbitrary commands on either client or server. They couldn't fix the problem with out breaking the protocol so they introduced a way to detect the attack. They introduced a file called deattack.c but a vulnerability existed in this file.

> "There is a remote integer overflow vulnerability in several implementations of the SSH1 protocol. This vulnerability is located in a segment of code that was introduced to defend against exploitation of CRC32 weaknesses in the SSH1 protocol (see VU#13877). The attack detection function (detect_attack, located in deattack.c) makes use of a dynamically allocated hash table to store connection information that is then examined to detect and respond to CRC32 attacks. By sending a crafted SSH1 packet to an affected host, an attacker can cause the SSH daemon to create a hash table with a size of zero. When the detection function then attempts to hash values into the null-sized hash table, these values can be used to modify the return address of the function call, thus causing the program to execute arbitrary code with the privileges of the SSH daemon, typically root."
>
> This description of the SSH vulnerability is from the cert advisory located at the following link. http://www.kb.cert.org/vuls/id/945216

There is a great description of this vulnerability and a walk through of the vulnerable code located at the following link.
http://www.securiteam.com/securitynews/5LP042K3FY.html
I would also recommend visiting the security focus archive that also gives a good description of the vulnerable code.
http://www.securityfocus.com/archive/1/161444

Correlations
The following traffic was reported to Vicki Iriwin who was the handler on duty at Incidents.org. They received many alerts in their logs for port 22 which is associated with SSH. http://www.incidents.org/diary/november01/111201.php

```
Nov  1 18:47:47 victim sshd[9531]: log: Connection from 10.10.10.10
port 33318
Nov  1 18:47:47 victim sshd[9532]: log: Connection from 10.10.10.10
port 33319
Nov  1 18:47:48 victim sshd[9533]: log: Connection from 10.10.10.10
port 33320
Nov  1 18:47:48 victim sshd[9534]: log: Connection from 10.10.10.10
port 33321
Nov  1 18:47:48 victim sshd[9535]: log: Connection from 10.10.10.10
port 33322
Nov  1 18:47:49 victim sshd[9536]: log: Connection from 10.10.10.10
port 33323
Nov  1 18:47:49 victim sshd[9537]: log: Connection from 10.10.10.10
port 33324
Nov  1 18:47:50 victim sshd[9538]: log: Connection from 10.10.10.10
port 33325
Nov  1 18:47:50 victim sshd[9539]: log: Connection from 10.10.10.10
port 33326
Nov  1 18:47:50 victim sshd[9540]: log: Connection from 10.10.10.10
port 33327
Nov  1 18:47:51 victim sshd[9541]: log: Connection from 10.10.10.10
port 33328
```

Nov 1 18:47:51 victim sshd[9542]: log: Connection from 10.10.10.10 port 33329

There was also the payload of an SSH attack which included in the shell code to execute bin/sh. You will see that string near the middle of the payload.

```
10.10.10.10:32957 -> 10.10.10.3:2222 [A]
  ................................................................
  ................................................................
  ................................................................
  ................................................................
  ................................................................
  ................................................................
  ................................................................
  ................................................................
  ..................1..f..1...C.].C.].K.M..M...1..E.Cf.].f.E.09.M..E.
  .E..E.....M.....CC....C....1..?......A....^.u.1..F..E......M..U.......
  ./bin/sh.h0h0h0, 7350, zip/TESO!................................
  ................................................................
  ................................................................
  ................................................................
  ................................................................
  ................................................................
  ................................................................
```

```
.....................................................................
.....................................................................
.....................................................................
.....................................1...p}.@
```

I also found log messages similar to mine reported by Johan Augustsson at the following link. http://archives.unixtech.be/arch055/3451.html

> > > Nov 25 11:37:40 ns sshd[10994]: Disconnecting: crc32 compensation
> attack:
> > > network attack detected
> > > Nov 25 11:37:48 ns sshd[11006]: Disconnecting: Corrupted check bytes on
> > > input.
> > > Nov 25 11:37:53 ns sshd[11013]: Disconnecting: Corrupted check bytes on
> > > input.
> > > Nov 25 11:37:54 ns sshd[11014]: Disconnecting: Corrupted check bytes on
> > > input.

You may also want to look at the following link containing a write up this attack by Dave Dittrich. dittrich@cac.washington.edu
http://lists.bikkel.org/archive/whitehat/Week-of-Mon-20011105/000215.html

Evidence of Active Targeting
This was more that likely active targeting. The fact that the attacker knew that particular server was running SSH was not a coincidence. They used an SSH exploit which infers that they knew the host was running SSH and that the host was accessible. I would like to go back and review the scan logs to see if any scans came from this attacker prior to the attack. A scan originating from this attacker in the weeks prior would be concrete evidence of active targeting.

Severity
Criticality = 4 This server is used as part of the business infrastructure and causes many interruptions when it is down. It is not a mail server or DNS, but from this system an attacker may be able to access other systems with information he could gather.

Lethality = 5 Any attack that gives an attacker root on one of my network devices is considered to be extremely lethal.

System Countermeasures = 0 The targeted host was running a version of OpenSSH prior to version 2.3.0 where the problem was addressed.

Network Countermeasure = 0 This server is in the DMZ and is allowed to be accessed from the internet. There are no ACL's in place or rules on the firewall that would prevent this attack.

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

(4 + 5) – (0 + 0) = 9

Defensive Recommendations
First off I will recommend rebuilding this server and wiping the disks. If the attacker gained root access to the system there is no telling what sort of Trojan, or rootkit software he may have downloaded and installed. With out a program such as tripwire it is extremely difficult to track down system modifications. Installing Tripwire is a must. www.tripwire.org
Any box that has been compromised should be rebuilt. I would recommend that upon rebuilding the server that the version of SSH is greater than 2.3. Open ssh is available at http://www.openssh.com Furthermore hardening of the firewall would be in order. Since SSH into this server is a necessity considerations should be made as to which IP addresses or groups of IP addresses are allowed to access it and be enforced on the firewall using access controls. For future network design decisions I would remove the necessity of SSH in the DMZ. If access to the DMZ is necessary from offsite I would recommend the use of a VPN device which supports authentication and encryption.

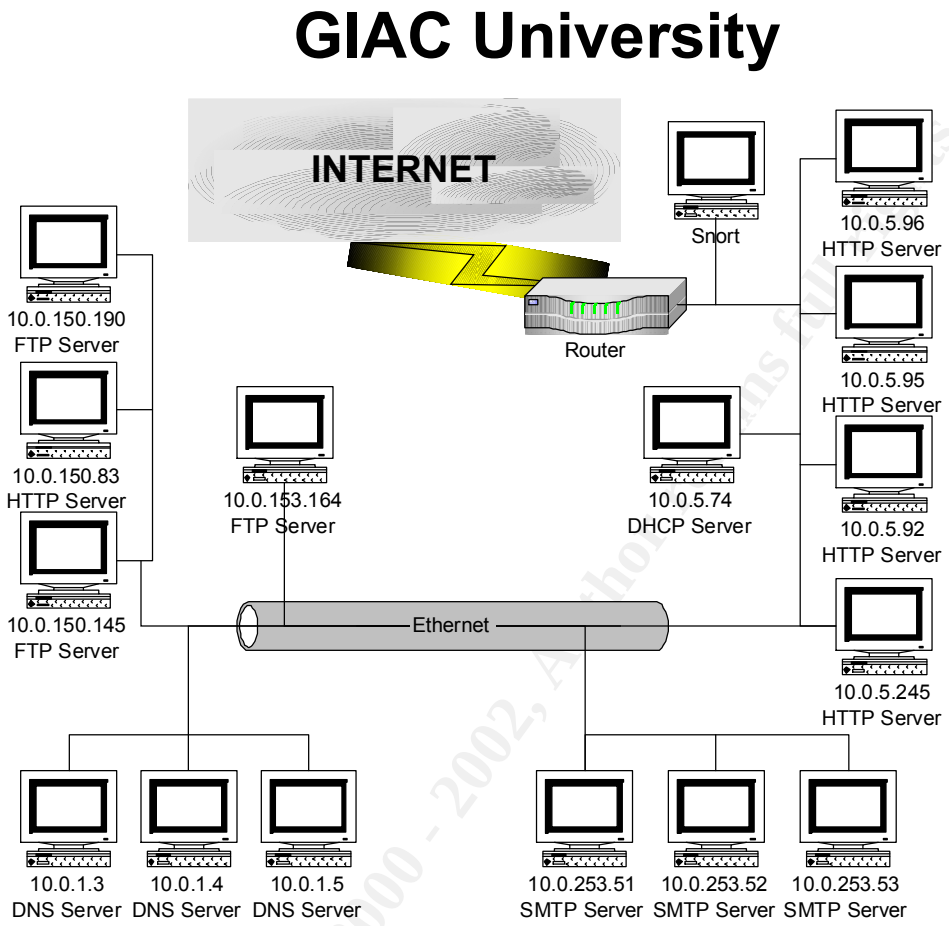
Multiple Choice Question
Buffer overflows fill up memory with NOOPS (no operation), which causes the stack to crash and allows for the execution of arbitrary code. Which of the following strings would you look for in a snort rule to find packets with NOOP's?

A) %F
B) %%
C) 90
D) 09

The answer is c 90.
You will often see packets filled with "90" which may indicate padding for a buffer flow attack.
See the following snort rule for an example of how you could detect NOOP's in a packet.

alert tcp $NET any -> $MY_NET any (msg:"**Possible BufferFlow NOOPs FOUND**"; flags:A+; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90|";

Assignment 3 "Analyze This"



This is the assumed topology of the GIAC University Network. All assumptions here were based on traffic that was seen in the supplied Alert, Scan, and OOS data files. If there are services that I have represented here which are actually not running on these servers than further investigation is necessary.

Executive Summary

I have been hired to conduct a security audit for the GIAC University Network. I was supplied with five days worth of data ranging from Snort Alerts to Scan Logs and OOS events. The date range I analyzed was from 1-4-02 through 1-8-02. I received the following files:

Snort Alert	OOS	SCAN
alert.020104.gz	oos_Jan.4.2002.gz	scans.020104.gz
alert.020105.gz	oos_Jan.5.2002.gz	scans.020105.gz
alert.020106.gz	oos_Jan.6.2002.gz	scans.020106.gz
alert.020107.gz	oos_Jan.7.2002.gz	scans.020107.gz
alert.020108.gz	oos_Jan.8.2002.gz	scans.020108.gz

For this analysis I have converted all the MY.NET addresses to the 10.0. address space. I needed to do this conversion for my queries and reports to work properly. From this point

on I will not reference MY.NET and all references will be made to 10.0.X.X addresses. Included in my report you will find analysis of the top 20 alerts reported for the specified time range, including a description of each and defensive recommendations. You will also find a top talkers summary for each of the data sets along with who is information for those hosts. In the last section I have included a link graph that displays all traffic to and from a host that I believe to have been compromised. I have also indicated through out the report any hosts that I feel should be investigated further to confirm whether or not they have been compromised.

Alert Summary

This Report represents the Unique Alerts that were reported from snort. I have excluded the Scans and the ICMP traffic that I didn't find to be an indication of any compromised system. Following this report I have included a description of the top 20 alerts and defensive recommendations for each. Followed by any external correlations I could find on either the external source or target addresses.

Count of Distinct Alerts Excluding Scans and ICMP

Event Name	count of Event Name	Distinct SRC	Distinct TGT
connect to 515 from inside	19538	38	2
spp_http_decode: IIS Unicode attack detected	17315	64	259
SNMP public access	11554	14	136
MISC Large UDP Packet	8660	9	7
INFO - ICQ Access	1910	2	42
INFO MSN IM Chat data	1827	47	49
High port 65535 udp - possible Red Worm - traffic	1209	21	76
SMB Name Wildcard	908	32	25
Watchlist 000220 IL-ISDNNET-990517	307	4	6
FTP DoS ftpd globbing	189	3	2
WEB-MISC Attempt to execute cmd	153	12	5
WEB-CGI scriptalias access	96	1	1
spp_http_decode: CGI Null Byte attack detected	96	1	1
EXPLOIT x86 NOOP	82	3	3
OOS Event	64	14	6
Possible trojan server activity	48	7	7
TCP SRC and DST outside network	37	9	5
INFO FTP anonymous FTP	21	6	2
Incomplete Packet Fragments Discarded	17	3	3
High port 65535 tcp - possible Red Worm - traffic	13	4	4
WEB-MISC 403 Forbidden	13	1	10
FTP passwd attempt	12	1	2
WEB-IIS _vti_inf access	11	7	1
WEB-FRONTPAGE _vti_rpc access	10	6	1
EXPLOIT x86 setuid 0	9	6	6
INFO Inbound GNUTella Connect accept	9	2	9
WEB-IIS view source via translate header	9	2	1

INFO Possible IRC Access	7	3	3
IDS552/web-iis_IIS ISAPI Overflow ida nosize	6	6	5
Back Orifice	5	3	4
INFO Inbound GNUTella Connect request	5	4	2
WEB-CGI formmail access	5	5	1
EXPLOIT x86 setgid 0	5	4	5
EXPLOIT NTPDX buffer overflow	4	2	2
NMAP TCP ping!	4	2	2
Port 55850 udp - Possible myserver activity - ref. 010313-1	4	1	2
Watchlist 000222 NET-NCFC	4	1	1
Tiny Fragments - Possible Hostile Activity	4	2	1
Attempted Sun RPC high port access	3	2	3
EXPLOIT x86 stealth noop	3	2	2
WEB-IIS Unauthorized IP Access Attempt	3	1	2
INFO Napster Client Data	3	1	2
MISC traceroute	3	2	2
INFO Outbound GNUTella Connect accept	2	2	2
WEB-MISC compaq nsight directory traversal	2	1	1
SUNRPC highport access!	2	2	2
FTP CWD - possible warez site	1	1	1
TFTP - External UDP connection to internal tftp server	1	1	1
x86 NOOP - unicode BUFFER OVERFLOW ATTACK	1	1	1
X11 outgoing	1	1	1
WEB-CGI redirect access	1	1	1
TFTP - Internal UDP connection to external tftp server	1	1	1
Queso fingerprint	1	1	1

Connect to 515 from Inside

Event Name	count of Event Name	Distinct SRC	Distinct TGT
Connect to 515 from inside	19538	38	2

➢ **Brief Alert Description**

Port 515 is LPRng service which is basically a printer service. There have been several vulnerabilities published which target particular versions of this service. The vulnerability is explained in the following quote from a SANS security advisory. "The vulnerability is due to incorrect usage of the syslog (3) function. Local and remote users can send string-formatting operators to the printer daemon to corrupt the daemon's execution, potentially gaining root access."
http://www.sans.org/newlook/alerts/port515.htm
Since port 515 is potentially vulnerable I decided to look at the distinct sources from which this event was triggered. They are listed bellow.

Distinct Sources

10.0.152.166	10.0.153.120	10.0.153.154	10.0.153.195
10.0.152.167	10.0.153.121	10.0.153.163	10.0.153.209

10.0.152.170	10.0.153.122	10.0.153.164	10.0.253.10
10.0.152.182	10.0.153.125	10.0.153.165	10.0.88.148
10.0.153.106	10.0.153.126	10.0.153.172	10.0.88.181
10.0.153.108	10.0.153.127	10.0.153.173	10.0.153.119
10.0.153.109	10.0.153.137	10.0.153.174	10.0.153.150
10.0.153.114	10.0.153.146	10.0.153.178	10.0.153.193
10.0.153.115	10.0.153.148	10.0.153.186	
10.0.153.117	10.0.153.149	10.0.153.189	

Since all of these source addresses are internal GIAC university addresses I would assume that this traffic is legitimate network traffic and that the two target addresses 10.0.150.198 and 10.0.153.111 are actually running some sort of network printer services and this traffic should not be reason for alert.

- **Defensive Recommendations**

In this case I have made the assumption that the two servers that are targeted by this alert are running print services and this traffic should not be investigated further. I would however be suspicious if I saw external connections to either of these servers on port 515 so as to help ensure that these services are not accessible from the Internet, I would block that port at the perimeter firewall. As well as blocking this service from being accesses by external sources, I would ensure that the servers are up to the latest patch level and should run LPRng version 3.6.25 as a minimum.

- **Correlations**

Many sites have seen increased number of scans looking for port 515. These sites include the following:
http://www1.dshield.org/port_report.php?port=515
http://www.sans.org/newlook/alerts/port515.htm
Since none of the Source IP addresses were external I was unable to find any source IP address correlations.

spp_http_decode: IIS Unicode attack detected

Event Name	count of Event Name	Distinct SRC	Distinct TGT
spp_http_decode: IIS Unicode attack detected	17315	64	259

- **Brief Alert Description**

The IIS Unicode attack is an attack that targets windows servers running IIS web services, if successful this exploit allows attackers to execute commands by issuing crafted http queries. This exploit grew from an older exploit called dot dot, which allowed for directory traversal by the attacker. The Unicode vulnerability is exploited by substituting a standard "/" with the Unicode translation which is "\" The following quote is from the Xforce advisory regarding IIS Unicode attacks. "By appending the '..' and a Unicode slash or backslash after a virtual directory with execute permissions, it is possible for an attacker to execute arbitrary commands."
http://xforce.iss.net/alerts/advise68.php

Since this is a potentially serious vulnerability I decided it requires further investigation. Bellow are the internal target IP's. I am not concerned with target addresses that are external because some normal web traffic has been mistaken as Unicode alerts in the past.
http://www.securityfocus.com/archive/96/183184

Target IP's in GIAC University
10.0.11.4
10.0.150.83
10.0.5.245
10.0.5.92
10.0.5.95
10.0.5.96

I examined the traffic originating from these IP addresses to see if there was any indication that the Unicode attacks were successful and I found the following:

10.0.11.4
Traffic originating from this IP address was all reported as scans there were 70 events in the scans logs showing what appeared to be FTP traffic since all the records contained source port 20. I would be concerned about this traffic but all the addresses were internal which leads me to believe that this server runs FTP services. Other traffic targeted at this IP address all seemed to be normal network traffic that was reported as scans from internal sources. The only alert from an external source was the Unicode attack. I would consider this to be a false alert.

10.0.150.83, 10.0.5.245, 10.0.5.92, 10.0.5.95, 10.0.5.96
I believe these addresses to be web servers, as indicated in the network topology at the beginning of this section. They should be treated as a high risk and critical servers. I looked through the supplied data and there were no alerts, scans, or OOS events originating from these IP addresses which indicates that they were most likely not compromised. Although they were all targets of several other web exploits, the only server that was targeted by this exploit from an external source was 10.0.5.245. On Jan 5th at 10:56 there were 6 attempts from the same source address to this web server.

Detect Time	Event Name	Source Address	Target Address
5 Jan 2001 10:56:10 PST	spp_http_decode: IIS Unicode attack detected	211.93.8.74	10.0.5.245
5 Jan 2001 10:56:08 PST	spp_http_decode: IIS Unicode attack detected	211.93.8.74	10.0.5.245
5 Jan 2001 10:56:08 PST	spp_http_decode: IIS Unicode attack detected	211.93.8.74	10.0.5.245
5 Jan 2001 10:56:07 PST	spp_http_decode: IIS Unicode attack detected	211.93.8.74	10.0.5.245
5 Jan 2001 10:56:07 PST	spp_http_decode: IIS Unicode attack detected	211.93.8.74	10.0.5.245
5 Jan 2001 10:56:07 PST	spp_http_decode: IIS Unicode attack detected	211.93.8.74	10.0.5.245

Following is the ARIN whois information regarding this source address
Results:
Asia Pacific Network Information Center (NETBLK-APNIC-CIDR-BLK)
These addresses have been further assigned to Asia-Pacific users.
Contact info can be found in the APNIC database,

at WHOIS.APNIC.NET or http://www.apnic.net/
Please do not send spam complaints to APNIC.
AU

Netname: APNIC-CIDR-BLK2
Netblock: 210.0.0.0 - 211.255.255.255

Coordinator:
Administrator, System (SA90-ARIN) [No mailbox]
+61-7-3367-0490

Domain System inverse mapping provided by:

NS.APNIC.NET	203.37.255.97
SVC00.APNIC.NET	202.12.28.131
NS.TELSTRA.NET	203.50.0.137
NS.RIPE.NET	193.0.0.193

- **Defensive Recommendations**

Since these systems all appear to be web servers you cannot block port 80 access to them. If these servers were running IIS I would ensure that they are up to the latest patch release from Microsoft. If they are not running IIS than I would not worry about this exploit since it only targets vulnerabilities found in IIS servers. As for the source address 211.93.8.74 unless there is a legitimate reason for this traffic I would consider blocking this IP address from accessing internal servers. Further investigation would be required.

- **Correlations**

I found no correlations for this external source IP on Incidents.org, Google, or on the SANS site.

SNMP public access

Event Name	count of Event Name	Distinct SRC	Distinct TGT
SNMP public access	11554	14	136

- **Brief Alert Description**

Simple Network Management Protocol (SNMP), is used by administrators to monitor various network devices. This alert confirms the use of SNMP, which is considered to be a security risk especially if you are using a public community string. SNMP uses no authentication and is not encrypted. The only authentication is the use of a community string that is usually set to public by most vendors. SNMP poses several vulnerabilities according to The Top Ten vulnerabilities published by SANS. "Attackers can use this vulnerability in SNMP to reconfigure or shut down devices remotely. Sniffed SNMP traffic can reveal a great deal about the structure of your network, as well as the systems

and devices attached to it. Intruders use such information to pick targets and plan attacks." http://www.sans.org/topten.htm

➢ **Defensive Recommendations**

Unless SNMP is necessary and justified I would suggest not using SNMP and would disable it wherever possible. If you require the use of SNMP you may want to take several steps to secure it as much as possible. First make the MIBS read only so that they cannot be modified. Additional information on this can be obtained from http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm#xtocid210315 Another step that should be considered would be to change the default community name to something that an attacker wouldn't be able to guess as easily.

➢ **Correlations**

Since these were all internal to internal addresses I could find no external correlations for the source addresses.

MISC Large UDP Packet

Event Name	Count of Event Name	Distinct SRC	Distinct TGT
MISC Large UDP Packet	8660	9	7

➢ **Brief Alert Description**

This alert is triggering off a UDP datagram larger than 4000 bytes. These alerts would have been triggered by a snort rule that looks like this.

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

Source Address	Count of Source Address	Target Address	Count of Target Address
203.248.242.22	1897	10.0.153.154	2323
210.76.63.49	1519	10.0.88.167	1897
211.233.70.161	1263	10.0.88.165	1519
211.233.70.162	1060	10.0.153.210	969
203.199.69.118	969	10.0.153.45	941
202.102.29.141	593	10.0.150.143	593
216.106.166.211	476	10.0.153.118	418
216.106.166.164	465		
211.43.209.7	418		

While investigating these alerts further I believe that I have found several compromised hosts. I was looking at the traffic targeted at 10.0.153.154 and I saw external hosts accessing that server on port 7000 from source port 7001. These ports are associated with the Andrew File System or AFS. AFS is a distributed file system that was created at Carnegie Mellon University. The following link has a great deal of information on AFS. http://www.alw.nih.gov/Docs/AFS/AFS_toc.html What alarmed me was that the sources

were external which means that external hosts were accessing internal file systems. I decided to look into any other traffic that meets that criteria and I believe I have a list of 16 GIAC University hosts that are compromised. The hosts are

10.0.150.120
10.0.150.145
10.0.150.49
10.0.151.125
10.0.153.118
10.0.153.151
10.0.153.152
10.0.153.154
10.0.153.184
10.0.153.185
10.0.153.210
10.0.153.211
10.0.153.46
10.0.88.165
10.0.88.167
10.0.88.244

The following Alert and scan traffic is a small subset of all the traffic I saw like this.

Scan	UDP	7 Jan 2002 18:36:44 PST 10.0.88.165	7001 4.19.71.20	7000
Scan	UDP	4 Jan 2002 16:50:55 PST 10.0.88.167	7001 203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:50:58 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:50:58 PST 10.0.88.167	7001 203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:51:26 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:51:26 PST 10.0.88.167	7001 203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:52:26 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:52:26 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:52:26 PST 10.0.88.167	7001 203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:52:30 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:52:30 PST 10.0.88.167	7001 203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:52:35 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:52:35 PST 10.0.88.167	7001 203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:53:56 PST 10.0.88.167	7001 203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:53:58 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:11 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:11 PST 10.0.88.167	7001 203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:54:12 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:12 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:13 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:13 PST 10.0.88.167	7001 203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:54:14 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:14 PST 10.0.88.167	7001 203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:54:42 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:42 PST 10.0.88.167	7001 203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:42 PST 10.0.88.167	7001 203.248.242.22	7000
Scan	UDP	7 Jan 2002 18:53:23 PST 10.0.88.244	7001 211.112.95.120	7000

Scan	UDP	7 Jan 2002 17:54:28 PST 10.0.88.244	7001 211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:54:32 PST 10.0.88.244	7001 211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:54:36 PST 10.0.88.244	7001 211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:54:39 PST 10.0.88.244	7001 211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:54:58 PST 10.0.88.244	7001 211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:55:29 PST 10.0.88.244	7001 211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:55:59 PST 10.0.88.244	7001 211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:56:02 PST 10.0.88.244	7001 211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:56:30 PST 10.0.88.244	7001 211.174.63.106	7000

I also discovered similar traffic patterns and analysis is a prior GCIA Practical Written By Kevin Black. http://www.giac.org/practical/Kevin_Black_GCIA.doc
I agree with his analysis of this traffic and would consider this to be extremely hostile. I have done whois lookups on all the external sources which are accessing internal file systems and I found the following.

12.25.239.5
AT&T ITS (NET-ATT) ATT 12.0.0.0 - 12.255.255.255
Inflow (NETBLK-ATT137321616-232)ATT137321616-232 12.25.232.0 - 12.25.239.255
3WK Radio (NETBLK-INFLOW-12254-4841) INFLOW-12254-4841
12.25.239.0 - 12.25.239.15

203.199.69.118
203.248.242.22
Asia Pacific Network Information Center (APNIC2)
These addresses have been further assigned to Asia-Pacific users.
Contact info can be found in the APNIC database,
at WHOIS.APNIC.NET or http://www.apnic.net/
Please do not send spam complaints to APNIC.
AU

Netname: APNIC-CIDR-BLK
Netblock: 202.0.0.0 - 203.255.255.255
Maintainer: AP

207.189.78.230
207.189.78.234
207.189.78.235
Digital Island, Inc. (NETBLK-DIGISLE-2NET)
45 Fremont Street, Suite 1200
San Francisco, CA 94105
US

Netname: DIGISLE-2NET
Netblock: 207.189.64.0 - 207.189.95.255
Maintainer: DIIS

Coordinator:
Pease, Holly (HP113-ARIN) netreg@digisle.net
(415) 738-4164

211.112.95.120
211.174.63.106
211.233.27.138
211.233.50.56
211.233.70.162
211.233.70.163
211.43.209.7
Asia Pacific Network Information Center (NETBLK-APNIC-CIDR-BLK)
   These addresses have been further assigned to Asia-Pacific users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   AU

   Netname: APNIC-CIDR-BLK2
   Netblock: 210.0.0.0 - 211.255.255.255

   Coordinator:
      Administrator, System (SA90-ARIN) [No mailbox]

216.206.179.23
216.54.221.197
Qwest Communications (NETBLK-NET-QWEST-BLKS-1) NET-QWEST-BLKS-1
                                        216.206.0.0 -
216.207.255.255
Qwest CyberCenters (NETBLK-QWEST-216-206-176) QWEST-216-206-176
                                      216.206.176.0 -
216.206.191.255

4.19.71.20
GENUITY (NET-GNTY-4-0)   GNTY-4-0   4.0.0.0 -
4.255.255.255
R&C Productions, Inc (NETBLK-RCPROD2-71-07) RCPROD2-71-07
                                          4.19.71.0 -
4.19.71.255

63.210.101.143
Level 3 Communications, Inc. (NETBLK-LEVEL4-CIDR) LEVEL4-CIDR
                                       63.208.0.0 -
63.215.255.255
Streaming Media Corporation (NETBLK-NETBLK-NETBLOCK-STRM5)
NETBLK-NETBLOCK-STRM5
                                      63.210.101.0 -
63.210.101.255

66.38.185.143
66.54.188.69
66.54.188.70
Yipes Communications, Inc. (NETBLK-YIPES-BLK3) YIPES-BLK3
                                       66.54.128.0 -
66.54.255.255
Yipes Web Services (NETBLK-YPWS-BLK1) YPWS-BLK1   66.54.188.0 -
66.54.188.255

I am most concerned with the Asian pacific addresses that are accessing GIAC University. The other traffic is all originating from large service providers like Level 3 and Qwest. The internal hosts that the traffic originating from the Asian Pacific address space is displayed bellow and the target hosts should be examined very carefully.

Event Name	Protocol	Detect Time	Target Address	Target Port	Source Address	Source Port
Scan	UDP	4 Jan 2002 16:50:55 PST	10.0.88.167	7001	203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:50:58 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:50:58 PST	10.0.88.167	7001	203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:51:26 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:51:26 PST	10.0.88.167	7001	203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:52:26 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:52:26 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:52:26 PST	10.0.88.167	7001	203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:52:30 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:52:30 PST	10.0.88.167	7001	203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:52:35 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:52:35 PST	10.0.88.167	7001	203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:53:56 PST	10.0.88.167	7001	203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:53:58 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:11 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:11 PST	10.0.88.167	7001	203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:54:12 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:12 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:13 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:13 PST	10.0.88.167	7001	203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:54:14 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:14 PST	10.0.88.167	7001	203.248.242.22	7000
Scan	UDP	4 Jan 2002 16:54:42 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:42 PST	10.0.88.167	7001	203.248.242.22	7000
MISC Large UDP Packet		4 Jan 2002 16:54:42 PST	10.0.88.167	7001	203.248.242.22	7000
Scan	UDP	6 Jan 2002 17:58:57 PST	10.0.153.210	7001	203.199.69.118	7000
Scan	UDP	6 Jan 2002 17:58:59 PST	10.0.153.210	7001	203.199.69.118	7000
Scan	UDP	7 Jan 2002 13:05:54 PST	10.0.153.118	7001	211.43.209.7	7000
Scan	UDP	7 Jan 2002 13:08:21 PST	10.0.153.118	7001	211.43.209.7	7000
Scan	UDP	7 Jan 2002 13:10:23 PST	10.0.153.118	7001	211.43.209.7	7000
Scan	UDP	7 Jan 2002 13:13:37 PST	10.0.153.118	7001	211.43.209.7	7000
Scan	UDP	7 Jan 2002 13:13:44 PST	10.0.153.118	7001	211.43.209.7	7000
Scan	UDP	7 Jan 2002 13:13:49 PST	10.0.153.118	7001	211.43.209.7	7000
Scan	UDP	7 Jan 2002 13:15:10 PST	10.0.153.118	7001	211.43.209.7	7000
MISC Large UDP Packet		7 Jan 2002 13:15:10 PST	10.0.153.118	7001	211.43.209.7	7000
Scan	UDP	7 Jan 2002 13:17:11 PST	10.0.153.118	7001	211.43.209.7	7000
MISC Large UDP Packet		7 Jan 2002 13:17:11 PST	10.0.153.118	7001	211.43.209.7	7000
Scan	UDP	7 Jan 2002 14:46:20 PST	10.0.153.154	7001	211.233.70.162	7000
MISC Large UDP Packet		7 Jan 2002 14:46:20 PST	10.0.153.154	7001	211.233.70.162	7000
Scan	UDP	7 Jan 2002 14:46:45 PST	10.0.153.154	7001	211.233.70.162	7000

MISC Large UDP Packet		7 Jan 2002 14:46:45 PST	10.0.153.154	7001	211.233.70.162	7000
Scan	UDP	7 Jan 2002 17:54:28 PST	10.0.88.244	7001	211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:54:32 PST	10.0.88.244	7001	211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:54:36 PST	10.0.88.244	7001	211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:54:39 PST	10.0.88.244	7001	211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:54:58 PST	10.0.88.244	7001	211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:55:29 PST	10.0.88.244	7001	211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:55:59 PST	10.0.88.244	7001	211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:56:02 PST	10.0.88.244	7001	211.174.63.106	7000
Scan	UDP	7 Jan 2002 17:56:30 PST	10.0.88.244	7001	211.174.63.106	7000
Scan	UDP	7 Jan 2002 18:53:23 PST	10.0.88.244	7001	211.112.95.120	7000
Scan	UDP	8 Jan 2002 13:49:38 PST	10.0.153.184	7001	211.233.27.138	7000
Scan	UDP	8 Jan 2002 15:38:08 PST	10.0.153.151	7001	211.233.70.163	7000
Scan	UDP	8 Jan 2002 15:38:21 PST	10.0.153.151	7001	211.233.70.163	7000
Scan	UDP	8 Jan 2002 15:38:23 PST	10.0.153.151	7001	211.233.70.163	7000
Scan	UDP	8 Jan 2002 15:38:28 PST	10.0.153.151	7001	211.233.70.163	7000
Scan	UDP	8 Jan 2002 15:38:31 PST	10.0.153.151	7001	211.233.70.163	7000
Scan	UDP	8 Jan 2002 15:38:34 PST	10.0.153.151	7001	211.233.70.163	7000
Scan	UDP	8 Jan 2002 15:38:44 PST	10.0.153.151	7001	211.233.70.163	7000
Scan	UDP	8 Jan 2002 15:38:47 PST	10.0.153.151	7001	211.233.70.163	7000
Scan	UDP	8 Jan 2002 15:38:50 PST	10.0.153.151	7001	211.233.70.163	7000
Scan	UDP	8 Jan 2002 15:38:54 PST	10.0.153.151	7001	211.233.70.163	7000
Scan	UDP	8 Jan 2002 19:24:06 PST	10.0.153.185	7001	211.233.50.56	7000
Scan	UDP	8 Jan 2002 19:24:26 PST	10.0.153.185	7001	211.233.50.56	7000
Scan	UDP	8 Jan 2002 19:24:44 PST	10.0.153.185	7001	211.233.50.56	7000
Scan	UDP	8 Jan 2002 19:24:50 PST	10.0.153.185	7001	211.233.50.56	7000
Scan	UDP	8 Jan 2002 19:24:55 PST	10.0.153.185	7001	211.233.50.56	7000
Scan	UDP	8 Jan 2002 19:24:58 PST	10.0.153.185	7001	211.233.50.56	7000
Scan	UDP	8 Jan 2002 19:25:02 PST	10.0.153.185	7001	211.233.50.56	7000
Scan	UDP	8 Jan 2002 19:25:05 PST	10.0.153.185	7001	211.233.50.56	7000

INFO - ICQ Access

Event Name	count of Event Name	Distinct SRC	Distinct TGT
INFO - ICQ Access	1910	2	42

- **Brief Alert Description**

ICQ is an application used by messenger like programs that basically informs you when other users are online. As stated on ICQ.com "With ICQ, you can chat, send messages, files and URL's, play games, or just hang out with your fellow 'Netters' while still surfing the Net." So is this a problem? Well it depends. Cert advisory CA-2002-02 claims that there are several ICQ buffer overflows that allow an attacker to execute arbitrary commands on the compromised system. http://www.cert.org/advisories/CA-2002-02.html In this case the ICQ alerts originate from only two IP addresses in the GIAC university network. They are 10.0.151.79 and 10.0.5.239. I felt that further investigation was

necessary into these source addresses to determine if they were compromised or just chatting on the net.
After looking at all events originating from these two sources it is most likely that these are users who are accessing chat rooms and using internet services such as IRC and AOL messenger. Depending on GIAC University security policies this may require further investigation.

- **Defensive Recommendations**

Again defensive recommendations are dependent upon policy in this case. If users are allowed to access Internet chat rooms and service such ads ICQ then there are no defensive recommendations that need to be made. Although the policy makers may want to take into consideration that there are known exploits associated with the applications that use ICQ. If the security policy that's in place prohibits users from accessing such services then it may be a good idea to block the ports associated with these services at the perimeter firewall.

- **Correlations**

I found no Correlations for the external source addresses on Google, SANS, or Incidents.org.

INFO MSN IM Chat data

Event Name	count of Event Name	Distinct SRC	Distinct TGT
INFO MSN IM Chat data	1827	47	49

- **Brief Alert Description**

This alert is detecting the traffic associated with Instant Messenger by Microsoft. Instant messenger is used much the same as an ICQ based service like AOL Instant messenger. I would not consider these alerts to require further investigation but there are known exploits that can allow a hacker to execute arbitrary commands on the target host. I also found a virus which propagates via MSN messenger services. http://www.zdnet.com/products/stories/reviews/0,4161,2769395,00.html I would however for security reasons not allow instant messenger connections to external sources. Sometimes a messenger like service can be useful within a company for employees to communicate back and forth or in this case students at GIAC University may use IM to chat with fellow students, but I would make a recommendation that this be disallowed by the security policy of the university.

- **Defensive Recommendations**

If Instant messenger is not allowed by the security policy than the recommendation I can suggest would be to block the ports associated with IM at the perimeter firewall. The usual ports that IM uses are TCP High Ports. Just by allowing TCP High Port traffic in to your internal network opens you up for all kinds of other vulnerabilities where attackers may come into your network on these open ports. It is good practice to not allow any services into your network that are not specifically stated in the security policy and a true necessity for the operations of your organization.

➢ **Correlations**

I found no correlations for these external source addresses on Incidents.org, Google, or the SANS site.

High port 65535 udp - possible Red Worm - traffic

Event Name	count of Event Name	Distinct SRC	Distinct TGT
High port 65535 udp - possible Red Worm - traffic	1209	21	76

➢ **Top Five Sources generating this Alert:**

Source Address	count of Source Address
10.0.6.49	355
10.0.6.50	337
10.0.6.48	273
10.0.6.52	85
10.0.6.51	79

➢ **Top Ten Targets Receiving this Alert**

Target Address	count of Target Address
10.0.153.168	212
10.0.152.175	125
10.0.153.189	81
10.0.153.203	80
10.0.153.188	68
10.0.153.187	62
10.0.152.166	43
10.0.152.182	37
10.0.153.204	33
10.0.153.211	30

➢ **Brief Alert Description**

The Red Worm now known as the Adore Worm is similar to the Ramen or Lion worms. What this worm does is scan Linux hosts on the Internet looking to see if the are susceptible to several vulnerabilities. The checks it makes are to see if the following services are running LPRng, Wu-Ftpd, rpc-statd and Bind. Once the worm finds an vulnerable system it exploits it using one of the previously listed vulnerabilities and it then Trojans the `ps` binary and moves the original to `/usr/bin/adore`. It then attempts to send various system information including /etc/ftpusers, ifconfig output, ps -aux (using the original binary in /usr/bin/adore), /root/.bash_history, /etc/hosts, and /etc/shadow. There is a detailed explanation of this worm on the SANS site at http://www.sans.org/y2k/adore.htm Another thing to take into account is that the code for this worm can be modified very easily so one should not assume that it will be listening on port 65535.

➢ **Defensive Recommendations**

There is a utility that is available from Dartmouth University written by William Stearns which will detect the adore files on an infected system. This utility is available at http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm I would highly recommend that this tool be downloaded and run on all the hosts listed in the above tables. Another good recommendation would be to block emails to or from the following addresses adore9000@21cn.com, adore9000@sina.com, adore9001@21cn.com, adore9001@sina.com. The worm sends these email addresses your critical system information.

- **Correlations**

Since this potentially a very serious problem I felt that I should look further into traffic originating from the top internal targets and sources of this alert.

- **Internal Targets**

From the internal source address I found similar traffic from all of them that could indicate a code red infection. All the systems have traffic originating from them which could be the execution of a SYN scan of external hosts to see if they are listening on port 80. It could also just be a user on that system browsing the Internet. Either way I believe these hosts should be checked using the Adore Trojan utility from the link in defensive recommendations. The traffic bellow is from one such system. Some traffic has been removed due to the volume.

Event Name	Protocol	Detect Time	Target Address	Target Port	Source Address	Source Port	TCP Flags
Scan	TCP	7 Jan 2002 11:03:33 PST	202.101.165.244	80	10.0.153.168	3590	******S*
Scan	TCP	7 Jan 2002 11:03:29 PST	63.208.235.30	80	10.0.153.168	3589	******S*
Scan	TCP	7 Jan 2002 11:03:29 PST	216.37.13.196	80	10.0.153.168	3586	******S*
Scan	TCP	7 Jan 2002 11:03:26 PST	202.101.165.244	80	10.0.153.168	3585	******S*
Scan	TCP	7 Jan 2002 11:03:19 PST	202.101.165.244	80	10.0.153.168	3583	******S*
Scan	TCP	7 Jan 2002 10:59:59 PST	202.101.165.244	80	10.0.153.168	3463	******S*
Scan	TCP	7 Jan 2002 10:59:55 PST	202.101.165.244	80	10.0.153.168	3462	******S*
Scan	TCP	7 Jan 2002 10:59:51 PST	202.101.165.244	80	10.0.153.168	3456	******S*
Scan	TCP	7 Jan 2002 10:59:47 PST	202.101.165.244	80	10.0.153.168	3451	******S*
Scan	TCP	7 Jan 2002 10:59:43 PST	202.101.165.244	80	10.0.153.168	3449	******S*
Scan	TCP	7 Jan 2002 10:59:39 PST	202.101.165.244	80	10.0.153.168	3446	******S*
Scan	TCP	7 Jan 2002 10:59:35 PST	202.101.165.244	80	10.0.153.168	3441	******S*
Scan	TCP	7 Jan 2002 10:59:31 PST	202.101.165.244	80	10.0.153.168	3439	******S*
Scan	TCP	7 Jan 2002 10:59:27 PST	202.101.165.244	80	10.0.153.168	3432	******S*
Scan	TCP	7 Jan 2002 10:59:23 PST	202.101.165.244	80	10.0.153.168	3427	******S*
Scan	TCP	7 Jan 2002 10:46:38 PST	208.184.29.190	80	10.0.153.168	2704	******S*
Scan	TCP	7 Jan 2002 10:46:38 PST	205.138.3.22	80	10.0.153.168	2703	******S*

- **Internal Sources**

None of the internal targets of this alert made any connections to an external address which leads me to believe that they are most likely not compromised. Although I did find some very interesting traffic originating from one of those hosts.

Originating from **10.0.6.49** I found many alerts which were detected as scans, but none of them were targeted at any external addresses which would commonly be the case of a host infected with code red. Some traffic I did see which causes me to believe that this

host is possibly compromised with another trojan is displayed in the following chart. Some of the traffic has been removed due to the large volume.

Event Name	Detect Time	Target Address	Target Port	Source Address	Source Port
Scan	7 Jan 2002 22:43:50 PST	10.0.153.211	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:43:50 PST	10.0.152.216	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:43:50 PST	10.0.153.143	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:43:50 PST	10.0.153.162	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:43:50 PST	10.0.153.204	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:38:50 PST	10.0.153.211	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:38:50 PST	10.0.152.216	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:38:50 PST	10.0.153.143	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:38:50 PST	10.0.153.162	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:38:50 PST	10.0.153.204	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:28:50 PST	10.0.153.211	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:28:50 PST	10.0.153.143	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:28:50 PST	10.0.153.162	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:28:50 PST	10.0.153.148	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:28:50 PST	10.0.153.204	7001	10.0.6.49	7000
Scan	7 Jan 2002 22:23:50 PST	10.0.153.211	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:18:13 PST	10.0.153.143	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:18:13 PST	10.0.153.196	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:18:13 PST	10.0.153.178	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:18:13 PST	10.0.153.140	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:18:13 PST	10.0.153.162	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:18:13 PST	10.0.153.148	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:18:13 PST	10.0.153.204	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:13:13 PST	10.0.153.211	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:13:13 PST	10.0.152.216	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:13:13 PST	10.0.153.196	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:13:13 PST	10.0.153.178	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:13:13 PST	10.0.153.140	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:13:13 PST	10.0.153.162	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:13:13 PST	10.0.153.148	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:13:13 PST	10.0.153.204	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:08:12 PST	10.0.152.216	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:08:12 PST	10.0.153.196	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:08:12 PST	10.0.153.178	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:08:12 PST	10.0.153.140	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:08:12 PST	10.0.153.162	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:08:12 PST	10.0.153.148	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:08:12 PST	10.0.153.204	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:03:12 PST	10.0.152.216	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:03:12 PST	10.0.153.196	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:03:12 PST	10.0.153.178	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:03:12 PST	10.0.153.140	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:03:12 PST	10.0.153.162	7001	10.0.6.49	7000
Scan	7 Jan 2002 21:03:12 PST	10.0.153.148	7001	10.0.6.49	7000

Scan	7 Jan 2002 21:03:12 PST 10.0.153.204	7001 10.0.6.49	7000
Scan	7 Jan 2002 21:01:13 PST 10.0.153.143	7001 10.0.6.49	7000

The reason I believe this traffic may represent a problem is that port 7000 is commonly associated with a Trojan called Freak-88. Freak 88 is a client server program that allows attackers to launch denial of service attacks on victims using ping floods. If a host is infected with freak 88 the host will be used as a zombie to launch larger DDoS attacks. This Trojan affects Windows 2000 and Windows NT4 systems. More information on freak88 is available at the following link. http://www.tlsecurity.net/backdoor/freak88.htm To discover if this system is infected with freak88 open the task manager and look for a process called PROJECT1 and do an end process. The Trojan does not write to the registry and will not restart after being stopped. There is another explanation for this traffic that could be the use of the Andrew file system. The Andrew file system was discussed earlier.

SMB Name Wildcard

Event Name	count of Event Name	Distinct SRC	Distinct TGT
SMB Name Wildcard	908	32	25

- **Brief Alert Description**

SMB Name wild card I most likely triggering off the following string located in the payload of a packet.

```
00 D8 00 00 00 01 00 00 00 00 00 00 20 43 4B 41 ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21 AAAAAAAAAAAAA..!
00 01 ..
```

The source host in this case is requesting the NetBIOS information from the target host as part of the Windows file sharing protocol to obtain domain, user and host id Information. NetBIOS information is a great place for an attacker to do reconnaissance because of the information that NetBIOS queries will typically give you.

- **Defensive Recommendations**

  Unless totally necessary it is a good idea to disable Windows file and print sharing especially on critical servers. There are many CVE and CERT advisories for NetBIOS. A few are listed below:

  http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0288
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0347
  http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0673
  http://www.kb.cert.org/vuls/id/32650
  http://www.cert.org/incident_notes/IN-2000-02.html

If you do need these services than you may want to look into host based firewalls that will only allow connections from trusted sources. In this case I found no external addresses as a source or target for any of these alerts, but I would have in place a perimeter firewall blocking ports 137-139 TCP and UDP inbound and outbound. Typically you don't want external sources accessing internal file servers. This would prevent that.

- **Correlations**

Since all the target and source addresses were internal there will be no external correlations.

Watchlist 000220 IL-ISDNNET-990517

Event Name	count of Event Name	Distinct SRC	Distinct TGT
Watchlist 000220 IL-ISDNNET-990517	307	4	6

- **Brief Alert Description**

A watch list is typically a custom snort rule that watches for connection attempts from or to a pre defined set of addresses. You would typically set up a watch list to alert you if there are connection attempts from a subnet that you have already discovered to be hostile. In this case I believe that the watch list is looking for connections from the 212.179.*.* network block. Whois information for that block returned the following:

```
European Regional Internet Registry/RIPE NCC (NET-RIPE-NCC-)
   These addresses have been further assigned to European users.
   Contact info can be found in the RIPE database, via the
   WHOIS and TELNET servers at whois.ripe.net, and at
   http://www.ripe.net/perl/whois/
   NL

   Netname: RIPE-NCC-212
   Netblock: 212.0.0.0 - 212.255.255.255
   Maintainer: RIPE

   Coordinator:
      Reseaux IP European Network Co-ordination Centre Singel 258
(RIPE-NCC-ARIN)  nicdb@RIPE.NET
      +31 20 535 4444

   Domain System inverse mapping provided by:

   NS.RIPE.NET                 193.0.0.193
   NS.EU.NET                   192.16.202.11
   AUTH03.NS.UU.NET            198.6.1.83
   NS2.NIC.FR                  192.93.0.4
   SUNIC.SUNET.SE              192.36.125.2
   MUNNARI.OZ.AU        128.250.1.21
   NS.APNIC.NET                203.37.255.97

   To search on arbitrary strings, see the Database page on
   the RIPE NCC website at http://www.ripe.net/perl/whois/
```

The following IP addresses were the sources for all of the watch list alerts.

Source Address	count of Source Address
212.179.35.118	231
212.179.35.119	64
212.179.38.137	9
212.179.28.133	3

These source IP addresses attempted to access the following internal GIAC hosts.

Target Address	count of Target Address
10.0.153.148	156
10.0.153.162	72
10.0.153.143	36
10.0.153.178	31
10.0.150.220	9
10.0.5.97	3

- **Defensive Recommendations**

Since this network seems to be considered hostile I would consider just blocking it from accessing internal hosts all together. As far as I am concerned if there are legitimate reasons for hosts on that network to access yours then they will complain to their ISP which will turn around and ask GIAC University why they are being blacklisted. At this point you can explain to them that you have had suspicious traffic originating from their address space and if they can fix the problems on their end then you will lift the access control list.

- **Correlations**

I found no correlations on SANS, Incidents.org, or Google for these external source addresses. I did however look into other internal events with a target address in the 212.179 address space and I found the following

Detect Time	Event Name	Source Address	Target Address	Target Port	Source Port
7 Jan 2002 09:31:07 PST	Scan	10.0.88.183	212.179.218.94	1214	1770
7 Jan 2002 09:31:04 PST	Scan	10.0.88.183	212.179.218.94	1214	1770
7 Jan 2002 09:20:53 PST	Scan	10.0.88.183	212.179.218.94	1214	1740
5 Jan 2002 14:00:23 PST	Scan	10.0.150.209	212.179.236.241	6257	6257
4 Jan 2002 17:22:29 PST	Scan	10.0.150.209	212.179.249.81	6257	6257
4 Jan 2002 17:17:09 PST	Scan	10.0.150.209	212.179.199.196	6257	6257
4 Jan 2002 17:15:32 PST	Scan	10.0.150.209	212.179.199.196	6257	6257
4 Jan 2002 16:55:29 PST	Scan	10.0.150.209	212.179.238.135	6257	6257
4 Jan 2002 16:50:45 PST	Scan	10.0.150.209	212.179.238.193	6257	6257
4 Jan 2002 16:49:27 PST	Scan	10.0.150.209	212.179.201.27	6257	6257
4 Jan 2002 16:48:44 PST	Scan	10.0.150.209	212.179.238.193	6257	6257

I would assume the 1214 traffic to be associated with Kazza www.kazza.com. Kazza is a file-sharing engine much like Napster that also is a very good source for downloading viruses and Trojans. I couldn't find an explanation for port 6257 traffic so I would recommend that further investigation be done as to the reason for this traffic.

FTP DoS ftpd globbing

Event Name	count of Event Name	Distinct SRC	Distinct TGT
FTP DoS ftpd globbing	189	3	2

➢ **Brief Alert Description**

This attack is targeting servers running a version of FTP developed by Washington University called Wu-Ftpd. Wu-Ftpd allows clients to have files available for ftp actions based on "file globbing" patterns. The implementation of file globbing that is used by Wu-Ftpd contains a heap corruption vulnerability that allows an attacker to execute arbitrary code on a remote server

Target Address	count of Target Address
10.0.153.164	163
10.0.150.145	26

Source Address	count of Source Address
12.3.135.250	130
12.3.134.202	33
198.173.24.162	26

Whois returned the following information for these source addresses

Whois look up on: **12.3.134.202 and 12.3.135.250** using server:whois.arin.net at port:43

Results:

```
AT&T ITS (NET-ATT)            ATT             12.0.0.0 -
12.255.255.255
Lycoming College (NETBLK-LYCOMINGCOLLE-134) LYCOMINGCOLLE-134
                                          12.3.134.0 -
12.3.135.255
```

Whois look up on: **198.173.24.162** using server:whois.arin.net at port:43

Results:

```
Verio, Inc. (NET-VRIO-198-170)
   8005 South Chester Street
   Englewood,, CO 80112
   US

   Netname: VRIO-198-170
   Netblock: 198.170.0.0 - 198.173.255.255
   Maintainer: VRIO

   Coordinator:
     Verio, Inc.  (VIA4-ORG-ARIN)  vipar@verio.net
     303.645.1900

   Domain System inverse mapping provided by:

   NS0.VERIO.NET        129.250.15.61
   NS1.VERIO.NET        204.91.99.140
   NS2.VERIO.NET        129.250.31.190

   *Rwhois information on assignments from this block available
   at rwhois.verio.net port 4321

   Record last updated on 26-Sep-2001.
```

Database last updated on 10-Feb-2002 19:55:25 EDT.

It appears that the 12.3 address space belongs to Lycoming College. I would assume that there is a legitimate reason that they are attempting to access FTP services on the GIAC network. If there were a legitimate reason for this access I would consider these to be false alerts. However I would be more concerned about the alerts originating from 198.173.24.162. This address belongs to Verio Inc., who provides DSL and dialup Internet access. I would want an explanation as to why this address is attempting to use FTP services on your network.

➢ **Defensive Recommendations**

First off I would recommend using a service besides FTP to transfer files. You may want to look into secure FTP or SCP if external file sharing is necessary. If there is not a legitimate reason for these hosts to be accessing FTP servers on your network then you may want to block port 20 and 21 on your perimeter firewall. If you are running Wu-Ftpd I would also suggest patching your systems with the patches available from the following website. There is also a good explanation on the exploit associated with Wu-Ftpd.
http://www.securiteam.com/unixfocus/6U00V0035Q.html

➢ **Correlations**

I found no correlations for these source IP addresses on Incidents.org, the SANS web site or Google.

WEB-MISC Attempt to execute cmd

Event Name	count of Event Name	Distinct SRC	Distinct TGT
WEB-MISC Attempt to execute cmd	153	12	5

➢ **Brief Alert Description**

These alerts were most likely caused by the DoS.Storm.Worm. What this worm does is looks for Microsoft IIS servers and when it finds one it infects it with the worm. The worm attempts to find vulnerable systems on the internet. It exploits a Web Server Folder Traversal vulnerability that is described at the following web site.
http://www.microsoft.com/technet/security/bulletin/MS00-078.asp
The Following technical description of the worm is from the Symantec Security response website at http://securityresponse.symantec.com/avcenter/venc/data/dos.storm.worm.html
"When the worm finds a vulnerable system, it copies itself to the targeted system and sets it up to automatically run the worm, effectively making that system a zombie that participates in the hacker's e-war. To make sure that the worm is run on next system startup, the worm adds the value 666 c:\winnt\system32\storm\start.bat to the registry keys HKEY_LOCAL_MACHINE\Software\Microsoft\,
Windows\CurrentVersion\RunServices, and
HKEY_LOCAL_MACHINE\Software\Microsoft\
Windows\CurrentVersion\Run
This worm has two payloads:

A denial of service attack is initiated against http:/ /www.microsoft.com.
An email bombing session is started that sends email messages containing an obscene message to gates@microsoft.com."

The following tables represents the sources and targets of this alert.

Source Address	count of Source Address	Target Address	count of Target Address
203.229.99.13	22	10.0.150.83	52
194.226.220.22	16	10.0.5.95	36
130.212.18.250	14	10.0.5.96	33
130.82.102.68	14	10.0.5.92	21
203.229.98.65	14	10.0.5.245	11
203.229.98.10	11		
211.93.8.74	11		
211.181.253.31	11		
203.229.99.115	11		
130.251.80.10	10		
203.229.99.74	10		
203.229.99.1	9		

I decided to look into the source addresses to see if they were possibly hostile hosts. The following whois information is regarding the sources of this alert.

203.299.9*.*

```
Asia Pacific Network Information Center (APNIC2)
   These addresses have been further assigned to Asia-Pacific
users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   AU

   Netname: APNIC-CIDR-BLK
   Netblock: 202.0.0.0 - 203.255.255.255
   Maintainer: AP

   Coordinator:
      Administrator, System  (SA90-ARIN)  [No mailbox]
      +61-7-3367-0490

   Domain System inverse mapping provided by:

   SVC00.APNIC.NET              202.12.28.131
   NS.APNIC.NET                 203.37.255.97
   NS.TELSTRA.NET               203.50.0.137
   NS.RIPE.NET                  193.0.0.193

   Regional Internet Registry for the Asia-Pacific Region.
```

194.226.220.22

```
European Regional Internet Registry/RIPE NCC (NETBLK-RIPE-C2)
```

```
   These addresses have been further assigned to European users.
   Contact info can be found in the RIPE database, via the
   WHOIS and TELNET servers at whois.ripe.net, and at
   http://www.ripe.net/perl/whois/
   NL

   Netname: RIPE-CBLK2
   Netblock: 194.0.0.0 - 194.255.255.255
   Maintainer: RIPE

   Coordinator:
      Reseaux IP European Network Co-ordination Centre Singel 258
(RIPE-NCC-ARIN)  nicdb@RIPE.NET
      +31 20 535 4444

   Domain System inverse mapping provided by:

   NS.RIPE.NET                 193.0.0.193
   NS.EU.NET                   192.16.202.11
   AUTH03.NS.UU.NET            198.6.1.83
   NS2.NIC.FR                  192.93.0.4
   SUNIC.SUNET.SE              192.36.125.2
   MUNNARI.OZ.AU        128.250.1.21
   NS.APNIC.NET                203.37.255.97

   To search on arbitrary strings, see the Database page on
   the RIPE NCC website at http://www.ripe.net/perl/whois/

   Record last updated on 16-Oct-1998.
   Database last updated on  10-Feb-2002 19:55:25 EDT.
```

130.212.18.250

```
San Francisco State University (NET-FOGNET)
   1600 Holloway Avenue
   San Francisco, CA  94132
   US

   Netname: FOGNET
   Netblock: 130.212.0.0 - 130.212.255.255

   Coordinator:
      San Francisco State University  (ZS148-ARIN)
abuse@sfsu.edu
      415-338-1211

   Domain System inverse mapping provided by:

   THESUN.SFSU.EDU             130.212.10.163
   MERCURY.SFSU.EDU            130.212.10.162

   Record last updated on 12-Jul-2001.
   Database last updated on  10-Feb-2002 19:55:25 EDT.
```

130.82.102.68

```
University of St. Gallen (NET-UNISG)
   Informatikbereich
```

```
   Dufourstrasse 50
   CH-9000 St. Gallen
   CH

   Netname: UNISG
   Netblock: 130.82.0.0 - 130.82.255.255

   Coordinator:
      Germann, Clemens  (CG47-ARIN)  Clemens.Germann@UNISG.CH
      +41 71 224 2675

   Domain System inverse mapping provided by:

   BETA.UNISG.CH         130.82.128.1
   GAMMA.UNISG.CH               130.82.128.2
   SCSNMS.SWITCH.CH             130.59.1.30 130.59.10.30

   Record last updated on 13-May-1998.
   Database last updated on  10-Feb-2002 19:55:25 EDT.
```

211.*.*.*

```
Asia Pacific Network Information Center (NETBLK-APNIC-CIDR-BLK)
   These addresses have been further assigned to Asia-Pacific
users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   AU

   Netname: APNIC-CIDR-BLK2
   Netblock: 210.0.0.0 - 211.255.255.255

   Coordinator:
      Administrator, System  (SA90-ARIN)  [No mailbox]
      +61-7-3367-0490

   Domain System inverse mapping provided by:

   NS.APNIC.NET                 203.37.255.97
   SVC00.APNIC.NET              202.12.28.131
   NS.TELSTRA.NET               203.50.0.137
   NS.RIPE.NET                  193.0.0.193

   Regional Internet Registry for the Asia-Pacific Region.

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

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

130.251.80.10

```
University of Genova (NET-UG)
   Via Opera Pia, 11A
   I-16145 Genova
   IT
```

```
Netname: GENUANET
Netblock: 130.251.0.0 - 130.251.255.255

Coordinator:
   Podesta', Tiziana  (TP177-ARIN)  tiziana@cisi.unige.it
   +39 10 3532622

Domain System inverse mapping provided by:

SUN.CISI.UNIGE.IT            130.251.21.8
DIST.DIST.UNIGE.IT           130.251.1.4

Record last updated on 15-Mar-1993.
Database last updated on  10-Feb-2002 19:55:25 EDT.
```

➢ **Defensive Recommendations**

First of all I would contact the Administrator at Asia Pacific and inform them that hosts in their address space have been attacking your network and that you are going to block their entire address space from accessing your network. Alerting them to this fact will probably not accomplish much so I would put an ACL in place on your perimeter firewall blocking the address block or at least the hosts. Secondly I would contact the administrators at the University networks and inform them that they may have compromised hosts on their network. It would be a safe assume that they don't have very strict security measures in place therefore compromised hosts are likely. I would assume that these are all critical servers as they have been previously identified as University Web Severs. If these servers are running IIS I would get a virus detection tool like Symantec antivirus and run it on the hosts targeted by this alert. Symantec also has removal instructions at the following link.
http://www.symantec.com/avcenter/venc/data/dos.storm.worm.html
I would pay special attention to 10.0.5.92 as I believe it may be compromised. You may consider rebuilding that server and installing all the latest security patches available from Microsoft.
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-078.asp

➢ **Correlations**

The following site reported 1 attack(s) from - 203.229.98.10
http://www.jaosa.net/codered.txt
Most of the other source IP addresses have been reported to incidents.org numerous times. This would indicate that it is definitely a good idea to consider blocking these addresses from accessing your network.

WEB-CGI scriptalias access

Event Name	count of Event Name	Distinct SRC	Distinct TGT
WEB-CGI scriptalias access	96	1	1

➢ **Brief Alert Description**

According to CVE there is a vulnerability in apache that the "ScriptAlias directory in NCSA and Apache httpd allowed attackers to read CGI programs."
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0236
All of these alerts were generated from one source and target at one of the University's Web Servers. The whois information returned the following on the possible hostile IP address.
Whois look up on: 24.180.201.71 using server:whois.arin.net at port:43

Results:
@Home Network (NETBLK-HOME-2BLK)HOME-2BLK 24.176.0.0 - 24.183.255.255
@Home Network (NETBLK-BLTMMD1-MD-3) BLTMMD1-MD-3 24.180.192.0 - 24.180.207.255

➢ **Defensive Recommendations**

I would suggest blocking this IP address from accessing your network and you may want to advise their ISP of this traffic.

➢ **Correlations**

The following traffic was reported on the sans website from a previous date range of MY.NET traffic.
http://www.sans.org/capsans/snort/SnortA54.txt

```
11/20-07:38:53.855910  [**] spp_portscan: PORTSCAN DETECTED from
24.180.201.71 (STEALTH) [**]
11/20-07:38:55.497901  [**] spp_portscan: portscan status from
24.180.201.71: 1 connections across 1 hosts: TCP(1), UDP(0) STEALTH
[**]
11/20-07:38:57.788982  [**] spp_portscan: End of portscan from
24.180.201.71 (TOTAL HOSTS:1 TCP:1 UDP:0) [**]
```

http://www.sans.org/giactc/snort2/OOSche4.txt

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
02/02-15:35:24.509643 24.180.201.71:0 -> MY.NET.130.123:1185
TCP TTL:117 TOS:0x0 ID:52483  DF
21*F**AU Seq: 0x500003   Ack: 0x159BCDB6   Win: 0x5010
TCP Options => EOL EOL EOL EOL EOL EOL
```

spp_http_decode: CGI Null Byte attack detected

Event Name	count of Event Name	Distinct SRC	Distinct TGT
spp_http_decode: CGI Null Byte attack detected	96	1	1

➢ **Brief alert Description**

This alert is detecting a 'null byte' in the CGI string. A null byte would be %00, what that does is mask system commands from CGI security checks by hiding the commands behind a null byte. A packet of data that CGI scripts do not detect unless specifically

programmed to look for them. Basically a NULL Byte attack is when an attacker appends a %00 to a URL, in order to confuse a Perl script about where the end of it's input is. These alerts all originated from one IP address 24.180.201.71 and were all targeted at one of the University's web servers 10.0.5.96. One thing to note is that the source for these alerts is the same as the source of the previous WEB-CGI scriptalias access alerts. It appears to me that the person who this address belongs is trying to break into this web server using various CGI exploits, although I don't believe them to have been successful.

- **Defensive Recommendations**

Since the IP address generating these alerts has been very actively targeting this GIAC webserver I would again recommend that it be blocked from accessing GIAC University.

- **Correlations**

See the correlations section for the previous alert.

EXPLOIT x86 NOOP

Event Name	count of Event Name	Distinct SRC	Distinct TGT
EXPLOIT x86 NOOP	82	3	3

- **Brief Alert Description**

The NOOP alert is triggered when snort detects a number of contiguous bytes that could be no-operation machine language codes. NOOPs are often used to pad out buffer overflow attacks, so this alert is indicating that it may have found an attempt to run attack code via a buffer overflow exploit.

Target Address	count of Target Address	Source Address	count of Source Address
10.0.150.190	80	24.95.245.166	80
10.0.153.185	1	207.46.177.148	1
10.0.153.211	1	211.233.30.231	1

All of the traffic originating from 24.95.245.166 was targeted at port 20 which is the data exchange port for the FTP protocol. These could either be false alerts because of data that's being transferred, or this host could be trying to execute a buffer overflow on the 10.0.150.90 host. I also noticed Anonymous FTP access attempts from this host targeted at the same host. Since this is an external host trying to access FTP services on GIAC University's network I did a whois on the source address. I found the following.

Whois look up on: 24.95.245.166 using server:whois.arin.net at port:43

Results:
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 11-Feb-2002 19:56:34 EDT.

Road Runner is an ISP in Texas and parts of the east coast. I would consider this traffic to be possibly hostile and I believe that further investigation is necessary to determine if 10.0.150.190 is compromised since this host is believed to be a university FTP server. Looking at the events originating from 10.0.150.190 I found nothing that appeared to be hostile or indicate that the system is compromised.

- **Defensive Recommendations**

Since buffer overflows exploit vulnerable services I would try to ensure that all the servers targeted bye these alerts are up to the latest patch level and I would also consider blocking FTP access into the network from external sources. I would also recommend further investigation of the host 24.95.245.166 to see if there is a legitimate reason for them to be accessing the internal GIAC FTP servers.

- **Correlations**

I found no correlations for these external source addresses on the SANS site, Google or Incidents.org.

OOS Event

Event Name	count of Event Name	Distinct SRC	Distinct TGT
OOS Event	64	14	6

- **Brief Alert Description**

OOS events are reported when snort sees packets that are out of specification as described in the RFC's. This could include but is not limited to bad TCP flag combinations, invalid sequence numbers, and illegal fragments. OOS packets are often indications of packet crafting. Which can be used for various reconnaissance purposes. For example if an attacker crafts a packet with illegal flag combinations and sees how a host responds he has some indication as to the OS the host is running. By determining the

operating system of the remote host they know have a better idea how to plan their attack. This is known as OS Finger Printing.

Top 5 Source Address	Count of Source Address	Top5 Target Address	Count of Target Address
144.122.42.38	16	10.0.88.162	36
195.132.240.41	10	10.0.150.204	13
24.158.117.251	9	10.0.150.143	8
130.104.19.73	8	10.0.153.206	5
4.61.46.216	6	10.0.150.220	1

I looked at other traffic originating from these source addresses and found that **144.122.42.38** generated the following alerts. Null scan!, OOS Event, SCAN FIN, SCAN XMAS, SYN-FIN scan!, and Scan. This IP address has the following information registered with arin.net.

Middle East Technical University (NET-METU-NET)
METU Computer Center Inonu Bulvari - ODTU
Ankara, 06531
TR
Netname: METU-NET
Netblock: 144.122.0.0 - 144.122.255.255

Coordinator:
METU Hostmaster (MH2-ORG-ARIN) hostmaster@METU.EDU.TR
+90 312 2103330
Fax- +90 312 2101120

Domain System inverse mapping provided by:

NS1.METU.EDU.TR	144.122.199.90
NS2.METU.EDU.TR	144.122.199.93
NS1-AUTH.SPRINTLINK.NET	206.228.179.10
AUTH60.NS.UU.NET	198.6.1.181

Record last updated on 27-Oct-1998.
Database last updated on 11-Feb-2002 19:56:34 EDT.

All of these source IP addresses were the sources of numerous scans and other events. I would consider them to be hostile.
As for the targets of these alerts I decided to look into traffic originating from them in order to determine if they have been possibly compromised. What I did was look at all traffic originating from these target IP addresses with a destination being one of these source addresses. I figure that if there was a hostile code in one of these OOS packets than there will be connections from the compromised host back out to the attacking host or another host that the attacker has under their control.

10.0.88.162

Detect Time	Target Address	Target Port	Source Address	Source Port	TCP Flags

8 Jan 2002 20:16:56 PST	144.122.42.38	1214	10.0.88.162	3701 ******S*
8 Jan 2002 17:06:43 PST	144.122.42.38	1214	10.0.88.162	3328 ******S*
7 Jan 2002 20:22:57 PST	144.122.42.38	1214	10.0.88.162	4780 ******S*
7 Jan 2002 18:25:35 PST	144.122.42.38	1214	10.0.88.162	4525 ******S*
7 Jan 2002 17:49:01 PST	144.122.42.38	1214	10.0.88.162	4433 ******S*
7 Jan 2002 16:55:21 PST	144.122.42.38	1214	10.0.88.162	4336 ******S*
7 Jan 2002 16:42:42 PST	144.122.42.38	1214	10.0.88.162	4307 ******S*
7 Jan 2002 06:09:25 PST	144.122.42.38	1214	10.0.88.162	3107 ******S*
7 Jan 2002 05:46:16 PST	144.122.42.38	1214	10.0.88.162	3079 ******S*

This appears to be traffic associated with Kazza. Kazza is an Internet file sharing service that was discussed earlier. I would assume that this host is not compromised.
From the other addresses I saw traffic that appeared to be Napster and more Kazza traffic. I would assume that these are not a high priority.

- **Defensive recommendations**

Again depending on the security policy for GIAC University you may want to consider disallowing Napster and Kazza traffic.

- **Correlations**

I found no correlations for these external addresses on Google, The SANS site, or Incidents.org

Possible trojan server activity

Event Name		count of Event Name		Distinct SRC	Distinct TGT
Possible trojan server activity		48		7	7
Target Address	**count of Target Address**	**Source Address**	**count of Source Address**		
10.0.13.12	8	10.0.5.83	8		
10.0.150.220	8	170.235.1.118	8		
170.235.1.118	8	10.0.150.220	8		
10.0.5.83	7	10.0.13.12	7		
10.0.5.83	4	10.0.5.119	4		
213.77.129.108	4	10.0.150.220	4		
10.0.5.119	3	10.0.5.83	3		
10.0.5.83	3	10.0.5.44	3		
10.0.150.220	2	213.77.129.108	2		
10.0.5.44	1	10.0.5.83	1		

- **Brief Alert Description**

These alerts are triggering off the target or the source port of 27374, which is commonly associated with the sub-seven Trojan. Sub seven is similar to Back Orifice or NetBus. The effect of this Trojan will enable people to access your system over the Internet with out you knowing. The Sub Seven Trojan can also be configured to inform someone when its infected computer connects to the Internet, and tells that person all the information

about you they need to use the Trojan against you. The notification will take place via ICQ or email. Numerous ICQ connections originating from any of the systems targeted by this alert would be a good indication that the system is compromised. There are very detailed description of Sub Seven located at the following links…
http://www.sans.org/newlook/resources/IDFAQ/subseven.htm
http://www.symantec.com/avcenter/venc/data/backdoor.subseven.html

Event Name	Detect Time	Target Address	Target Port	Source Address	Source Port
Possible trojan server activity	8 Jan 2002 07:22:19 PST	213.77.129.108	27374	10.0.150.220	1214
Possible trojan server activity	8 Jan 2002 07:22:13 PST	213.77.129.108	27374	10.0.150.220	1214
Possible trojan server activity	8 Jan 2002 07:22:13 PST	213.77.129.108	27374	10.0.150.220	1214
Possible trojan server activity	8 Jan 2002 07:22:13 PST	10.0.150.220	1214	213.77.129.108	27374
Possible trojan server activity	8 Jan 2002 07:22:10 PST	213.77.129.108	27374	10.0.150.220	1214
Possible trojan server activity	8 Jan 2002 07:22:10 PST	10.0.150.220	1214	213.77.129.108	27374
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.119	27374	10.0.5.83	7938
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.83	7938	10.0.5.119	27374
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.119	27374	10.0.5.83	7938
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.83	7938	10.0.5.119	27374
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.83	7938	10.0.5.119	27374
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.119	27374	10.0.5.83	7938
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.83	7938	10.0.5.119	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:27:05 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	7 Jan 2002 08:27:57 PST	170.235.1.118	27374	10.0.150.220	1214
Possible trojan server activity	7 Jan 2002 08:27:57 PST	10.0.150.220	1214	170.235.1.118	27374
Possible trojan server activity	7 Jan 2002 08:27:57 PST	10.0.150.220	1214	170.235.1.118	27374
Possible trojan server activity	7 Jan 2002 08:27:57 PST	170.235.1.118	27374	10.0.150.220	1214
Possible trojan server activity	7 Jan 2002 08:27:57 PST	10.0.150.220	1214	170.235.1.118	27374
Possible trojan server activity	7 Jan 2002 08:27:57 PST	170.235.1.118	27374	10.0.150.220	1214
Possible trojan server activity	7 Jan 2002 08:27:55 PST	10.0.150.220	1214	170.235.1.118	27374
Possible trojan server activity	7 Jan 2002 08:27:55 PST	170.235.1.118	27374	10.0.150.220	1214
Possible trojan server activity	7 Jan 2002 08:27:55 PST	170.235.1.118	27374	10.0.150.220	1214
Possible trojan server activity	7 Jan 2002 08:27:55 PST	170.235.1.118	27374	10.0.150.220	1214
Possible trojan server activity	7 Jan 2002 08:27:55 PST	10.0.150.220	1214	170.235.1.118	27374
Possible trojan server activity	7 Jan 2002 08:27:55 PST	170.235.1.118	27374	10.0.150.220	1214

Possible trojan server activity	7 Jan 2002 08:27:55 PST	10.0.150.220	1214	170.235.1.118	27374
Possible trojan server activity	7 Jan 2002 08:27:55 PST	10.0.150.220	1214	170.235.1.118	27374
Possible trojan server activity	7 Jan 2002 08:27:55 PST	170.235.1.118	27374	10.0.150.220	1214
Possible trojan server activity	7 Jan 2002 08:27:55 PST	10.0.150.220	1214	170.235.1.118	27374
Possible trojan server activity	5 Jan 2002 00:06:21 PST	10.0.5.83	7938	10.0.5.44	27374
Possible trojan server activity	5 Jan 2002 00:06:21 PST	10.0.5.44	27374	10.0.5.83	7938
Possible trojan server activity	5 Jan 2002 00:06:21 PST	10.0.5.83	7938	10.0.5.44	27374
Possible trojan server activity	5 Jan 2002 00:06:21 PST	10.0.5.83	7938	10.0.5.44	27374

Some of this traffic appears to be Kazza traffic targeted at port 27374 as a random ephemeral port but I am more concerned with the traffic that is not associated with Kazza.

When I remove the Kazza traffic from the previous list I get a smaller number of hosts that may be infected.

Event Name	Detect Time	Target Address	Target Port	Source Address	Source Port
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	5 Jan 2002 00:06:21 PST	10.0.5.44	27374	10.0.5.83	7938
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:27:05 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.119	27374	10.0.5.83	7938
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.119	27374	10.0.5.83	7938
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.13.12	27374	10.0.5.83	8733
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.119	27374	10.0.5.83	7938
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	8 Jan 2002 01:27:12 PST	10.0.5.83	8733	10.0.13.12	27374
Possible trojan server activity	5 Jan 2002 00:06:21 PST	10.0.5.83	7938	10.0.5.44	27374
Possible trojan server activity	5 Jan 2002 00:06:21 PST	10.0.5.83	7938	10.0.5.44	27374
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.83	7938	10.0.5.119	27374
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.83	7938	10.0.5.119	27374
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.83	7938	10.0.5.119	27374
Possible trojan server activity	8 Jan 2002 01:42:48 PST	10.0.5.83	7938	10.0.5.119	27374
Possible trojan server activity	5 Jan 2002 00:06:21 PST	10.0.5.83	7938	10.0.5.44	27374

Table 2

There are two other explanations for this traffic. Port 7938 is associated with Legato Network Backup Software that could be running on some of the hosts in question. It would seem that 10.0.5.83 would then be a back up server. The other port in question is 8733, which is commonly associated with the Ibus protocol. Ibus is a java messaging protocol, and further research as to why it is being used on university computers would

be necessary. Another note is that all the traffic that is not related to Kazza is from internal sources to internal targets.

- **Defensive Recommendations**

I would thoroughly check all the targets and internal sources of this alert for possible compromise and there are detailed instructions on doing so at following Symantec link. http://www.symantec.com/avcenter/venc/data/backdoor.subseven.html
I would also consider blocking port 27374 at the perimeter firewall inbound and out. I would also check the servers listed in the **table 2** for ICQ or email traffic to external sources as this could indicate that they have been compromised and communicating with attackers on the Internet.

- **Correlations**

I found no correlations for these external addresses on Incidents.org, The SANS site or Google.

TCP SRC and DST outside network

Event Name	count of Event Name	Distinct SRC	Distinct TGT
TCP SRC and DST outside network	37	9	5

- **Brief Alert Description**

This alert is triggering off snort seeing packets on the internal network that are not destined to or originating from an internal host. This is most likely signs of packet crafting originating from an internal host that has been compromised. It is extremely difficult to track down the true source of spoofed packets unless you can get the mac address or the physical address of the NIC card that's sending them. That is assuming that they are not passing through a router. If they are passing through a router then the Mac address you will see is that of the last hop router. I believe this could also be an indication of source routing. An attacker could specify in his source route that the packets are to pass through your router.

All of the traffic generating these alerts was from the 169.254.0.0 address space and was destined for the same. I noticed that all the traffic appeared to be NetBios connections either originating from or targeted for port 139. I also found an interesting quote that someone posted to SANS that said 169.254 addresses could also be explained as "supposed to be for newer DHCP clients that cannot get an internal address" Curtis L. Blais.

The who is information for the 169.254 address block returned the following information.

> Whois look up on: 169.254.146.18 using server:whois.arin.net at port:43
> IANA (NETBLK-LINKLOCAL)
> Internet Assigned Numbers Authority
> 4676 Admiralty Way, Suite 330
> Marina del Rey, CA 90292-6695

US
Netname: LINKLOCAL
Netblock: 169.254.0.0 - 169.254.255.255
Coordinator:
Internet Corporation for Assigned Names and Numbers (IANA-ARIN) res-ip@iana.org
(310) 823-9358
Domain System inverse mapping provided by:
BLACKHOLE-1.IANA.ORG 192.0.32.18
BLACKHOLE-2.IANA.ORG 192.0.32.19

➢ **Defensive Recommendations**

If this traffic is originating from within your network it is highly likely that you have an internal host that is compromised. I would recommend running tcpdump with the –e option which will allow you to view the mac address information. Once you do this you can write filters that will allow you to narrow down the traffic. I would recommend filtering on the 169.254 addresses and try to find the mac address where these packets are originating. Once you know the Mac address, write a filter based on that address and try to locate some "normal" or un-spoofed traffic to find out the real IP address of the compromised system. Another recommendation would be to implement egress filtering at your perimeter router. What egress filtering does is block any traffic outbound that is not originating from your address space. I would use the following paper written by Chris Brenton called "What is Egress Filtering and How Can I Implement It?" This paper gives an excellent of the problem and also real world examples of a solution. It is available at the following link. http://rr.sans.org/firewall/egress.php I also have to wonder if these addresses are some type of reserved addresses that shouldn't be routed in the first place. Then I may want to look at the traffic originating from the DHCP server which as earlier stated could assign these addresses. That would not however explain why other people have seen scans for NetBios ports originating from these IP addresses. If this just a problem with a dhcp server, I would still recommend egress filtering to avoid spoofed traffic from leaving your network in the future. I believe that this traffic requires further investigation before a concrete conclusion can be drawn as to the reason behind it.

➢ **Correlations**

There was a report of NetBios scans originating from the 169.254. address range at the following link.
http://www.theorygroup.com/Archive/Argus/1999/msg00165.html

Curtis Blais reported to incidents.org seeing NetBios scans of his network originating from the 169.254 address space. An example of the traffic is following.

```
11:15:44.684687 169.254.65.17.1078 hawk.MY.NETWORK.netbios-ssn:
  S 6292402:6292402(0) win 8192 (DF)
11:15:44.684960 169.254.65.17.1078 hawk.MY.NETWORK.netbios-ssn:
  S 6292402:6292402(0) win 8192 (DF)
11:15:44.685440 hawk.MY.NETWORK.netbios-ssn 169.254.65.17.1078:
  S 23800483:23800483(0) ack 6292403 win 8760 (DF)
```

```
11:15:47.654496 169.254.65.17.1078 hawk.MY.NETWORK.netbios-ssn:
  S 6292402:6292402(0) win 8192 (DF)
11:15:47.654756 169.254.65.17.1078 hawk.MY.NETWORK.netbios-ssn:
  S 6292402:6292402(0) win 8192 (DF)
11:15:47.655072 hawk.MY.NETWORK.netbios-ssn 169.254.65.17.1078:
  . ack 6292403 win 8760 (DF)
```

http://www.incidents.org/archives/intrusions/msg00055.html

INFO FTP anonymous FTP

Event Name	count of Event Name	Distinct SRC	Distinct TGT
INFO FTP anonymous FTP	21	6	2

➢ **Brief Alert Description**

FTP anonymous access is referring to someone trying to access an FTP server using the default anonymous login. FTP is the File Transfer Protocol. It is used primarily for transferring files from one host to another or from site to site. These alerts were targeted at 10.0.150.190 which has been identified as one of the University's FTP servers. All the sources for this alert were external and it should be looked into as to why they are attempting to logon to the FTP server anonymously. I would think that if they had a legitimate reason to access this server they would have a user name and a password. The following addresses were sources for this alert.

Source Address	count of Source Address
24.95.245.166	14
62.163.158.112	2
66.30.59.155	2
193.253.42.45	1
208.242.127.31	1
24.13.140.41	1

It appears that 24.95.245.166 tried 14 times to access this server anonymously 10 times on the 6th of January 2002 and 4 more times on the 7th. I find this to be curious so I did a who is on the address and found that the address belongs to Road Runner which is an ISP. This is most likely a home user. They may be a student that is allowed to access this ftp server maybe to get assignments or something but again I would think that they would have a user name and a password.

➢ **Defensive Recommendations**

I will make several recommendations in this case the first of which is don't use FTP if at all possible. There are many exploits that target FTP servers that allow the attacker to gain root access to the host. Another problem with FTP is that the username and password are transferred across the wire in the clear, meaning that anyone with a sniffer now knows your username and password. Rather than FTP I would consider using an service such as secure ftp or scp which will do encrypted authentication. If FTP is a necessity then I would recommend disabling anonymous access to all FTP servers.

➢ **Correlations**

I found no correlations for these source addresses on Google, Incidents.org or the SANS site.

Incomplete Packet Fragments Discarded

Event Name	count of Event Name	Distinct SRC	Distinct TGT
Incomplete Packet Fragments Discarded	17	3	3

- **Brief Alert Description**

Packet Fragments are often used to conceal malicious code and to avoid Intrusion detection systems. Attackers also use fragments to pass through not statefull firewalls. In this case the alert is being generated by snorts defrag plugin which is used to reassemble fragments to look at the contents. What this alert means is that while reassembling the packet snort never saw the final fragments and discarded them. You typically don't want to see fragmentation on your network and this alert indicates that you have anomalous fragments because snort never received the last fragment.
These alerts were targeted at 10.0.88.165, 10.0.150.143, and 10.0.153.207. I believe that these are not necessarily malicious because the originate from 3 separate hosts and each source targets the same target for each alert. I would want to understand further why we would be seeing fragmentation. One thing I noticed that was rather strange was that one of the source addresses was 172.16.2.100. 172 is a reserved address space like 10.0 and should not be routed on the internet. I would consider this to be a possibly spoofed source address or at least very strange and it would require further investigation as to how this could be possible with out spoofing. I looked into other traffic originating from the target of the alert with this strange source address and I didn't see anything out of the ordinary.

- **Defensive Recommendations**

I would suggest deploying a statefull firewall at the perimeter of the network so that attackers cannot bypass acl's on non statefull routers. I would also check the targeted hosts for signs of compromise. You may also consider looking into the source addresses of these alerts to see if they have a legitimate reason to be accessing the GIAC university network. They are listed bellow.
172.16.2.100
202.102.29.141
210.76.63.49

- **Correlations**

I obviously found no correlations for the 172.16.2.100 address. I did find correlations for the other two but they were in foreign languages that I do not speak. There was nothing on incidents.org or the SANS site.

High port 65535 tcp - possible Red Worm - traffic

Event Name	count of Event Name	Distinct SRC	Distinct TGT
High port 65535 tcp - possible Red Worm - traffic	13	4	4

➢ **Brief Alert Description**

I discussed what The code Red worm Is in detail in a previous section labeled High port 65535 UDP - possible Red Worm – traffic. This is the TCP version of the alert since Red Worm uses both TCP and UDP. I am very concerned with the highlighted sections of the following chart these are the 13 instances of this alert for the time period I analyzed. The non highlighted traffic I assume to be related to Kazza, but it may be a good idea to investigate that host further. The other two hosts are most likely compromised by the Code Red worm and I would recommend applying the defensive recommendations I made earlier to these hosts. I will copy those recommendations into this alert analysis so you don't have to go back and read them.

Detect Time	Target Address	Target Port	Source Address	Source Port
7 Jan 2002 19:46:47 PST	10.0.152.249	1460	10.0.60.8	65535
7 Jan 2002 19:46:47 PST	10.0.152.249	1460	10.0.60.8	65535
7 Jan 2002 19:46:50 PST	10.0.152.249	1460	10.0.60.8	65535
5 Jan 2002 13:56:33 PST	10.0.88.162	1214	195.219.155.203	65535
5 Jan 2002 13:56:34 PST	10.0.88.162	1214	195.219.155.203	65535
5 Jan 2002 13:56:32 PST	10.0.88.162	1214	195.219.155.203	65535
7 Jan 2002 19:46:47 PST	10.0.60.8	65535	10.0.152.249	1460
7 Jan 2002 19:46:47 PST	10.0.60.8	65535	10.0.152.249	1460
7 Jan 2002 19:46:47 PST	10.0.60.8	65535	10.0.152.249	1460
7 Jan 2002 19:46:47 PST	10.0.60.8	65535	10.0.152.249	1460

➢ **Defensive Recommendations**

There is a utility that is available from Dartmouth University written by William Stearns which will detect the adore files on an infected system. This utility is available at http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm I would highly recommend that this tool be downloaded and run on all the hosts listed in the above tables. Another good recommendation would be to block emails to or from the following addresses adore9000@21cn.com, adore9000@sina.com, adore9001@21cn.com, adore9001@sina.com. The worm sends these email addresses your critical system information.

➢ **Correlations**

Since I am only concerned with these two internal hosts and not the traffic I assume to be Kazza there are no external correlations.

TOP TALKERS

This section focus on the top external, and overall source addresses from the Scans, Alert, and OOS data files. As well as the top GIAC University targets. For the external sources I have included Whois information, which helps in determining the legitimacy of the traffic as well as deciding if the source should be considered hostile.

Top Ten Source Addresses From Alert Files

Top Ten Alert Source Addresses

Source Address	Count of Source Address
10.0.5.202	30109
10.0.153.114	5544
10.0.153.146	4580
10.0.88.181	2930
10.0.70.177	2903
10.0.150.198	2173
10.0.150.41	1933
203.248.242.22	1898
10.0.151.79	1866
10.0.153.119	1864

Since most of these are internal addresses I would investigate each as to the traffic that is triggering these alerts. If the traffic is not malicious I would consider modifying your Snort rule set.

203.248.242.22 This address belongs to the Asian Pacific Network and should be considered Hostile. I would suggest blocking access to GIAC University from this address.

```
Asia Pacific Network Information Center (APNIC2)
   These addresses have been further assigned to Asia-Pacific
users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   AU

   Netname: APNIC-CIDR-BLK
   Netblock: 202.0.0.0 - 203.255.255.255
   Maintainer: AP

   Coordinator:
      Administrator, System  (SA90-ARIN)  [No mailbox]
      +61-7-3367-0490

   Domain System inverse mapping provided by:

   SVC00.APNIC.NET                202.12.28.131
   NS.APNIC.NET                   203.37.255.97
```

```
NS.TELSTRA.NET                  203.50.0.137
NS.RIPE.NET                     193.0.0.193
```

Top Ten External Source Addresses From Alert Files

Top Ten External Alert Source Addresses

Source Address	count of Source Address
203.248.242.22	1898
210.76.63.49	1523
211.233.70.161	1263
211.233.70.162	1060
203.199.69.118	969
202.102.29.141	605
216.106.166.211	476
216.106.166.164	465
211.43.209.7	424
64.4.12.179	257

203.248.242.22
This address was discussed in the prior report. It is owned by the Asian Pacific Network and should be considered hostile.

210.76.63.49
211.233.70.161
211.233.70.162
202.102.29.141
211.43.209.7
I would consider blocking this address from accessing your internal servers.

```
Asia Pacific Network Information Center (NETBLK-APNIC-CIDR-BLK)
   These addresses have been further assigned to Asia-Pacific
users.
   Contact info can be found in the APNIC database,
   at WHOIS.APNIC.NET or http://www.apnic.net/
   Please do not send spam complaints to APNIC.
   AU

   Netname: APNIC-CIDR-BLK2
   Netblock: 210.0.0.0 - 211.255.255.255

   Coordinator:
      Administrator, System  (SA90-ARIN)  [No mailbox]
      +61-7-3367-0490

   Domain System inverse mapping provided by:

   NS.APNIC.NET                  203.37.255.97
```

```
SVC00.APNIC.NET                 202.12.28.131
NS.TELSTRA.NET                  203.50.0.137
NS.RIPE.NET                     193.0.0.193
```

216.106.166.211
216.106.166.164
This could be legitimate traffic I would investigate further before restricting access.

```
iBEAM Broadcasting Corporation (NETBLK-IBEAM)
   645 Almanor Ave., suite 100
   Sunnyvale, CA 94085
   US

   Netname: IBEAM
   Netblock: 216.106.160.0 - 216.106.175.255
   Maintainer: BEAM

   Coordinator:
      Le, Stewart  (SL895-ARIN)  stle@ibeam.com
      408-830-3572

   Domain System inverse mapping provided by:

   NS1.IBEAM.COM          204.233.70.15
   NS2.IBEAM.COM          204.247.99.125
```

64.4.12.179 I would want to investigate the traffic from this address further and most likely want an explanation as to the reason for it.

```
MS Hotmail (NETBLK-HOTMAIL)
   1065 La Avenida
   Mountain View, CA 94043
   US

   Netname: HOTMAIL
   Netblock: 64.4.0.0 - 64.4.63.255

   Coordinator:
      Myers, Michael  (MM520-ARIN)  icon@HOTMAIL.COM
      650-693-7072

   Domain System inverse mapping provided by:

   NS1.HOTMAIL.COM              216.200.206.140
   NS3.HOTMAIL.COM              209.185.130.68
```

Top Ten Alert Target Addresses

Top Ten Alert Target Addresses

Target Address	count of Target Address
10.0.5.1	30109
10.0.150.198	19537

10.0.152.109	5288
211.115.213.202	2720
10.0.153.154	2324
211.32.117.26	1927
10.0.88.167	1898
211.32.117.27	1729
10.0.88.165	1536
224.0.0.2	1119

Any internal GIAC University addresses that appear in this report I would check for signs of compromise.

SCAN LOG TOP TALKERS

Top Ten External SCAN Source Addresses

Top Ten External SCAN Source Addresses

Source Address	count of Source Address
205.188.233.153	6893
205.188.228.17	5536
205.188.233.121	5375
205.188.244.57	5339
205.188.228.33	4943
205.188.228.65	4829
12.25.239.5	4755
205.188.233.185	4159
205.188.228.1	3894
66.38.185.143	3369

205.188.233.153
205.188.228.17
205.188.233.121
205.188.244.57
205.188.228.33
205.188.228.65
205.188.233.185
205.188.228.1

```
America Online, Inc (NETBLK-AOL-DTC)
   22080 Pacific Blvd
   Sterling, VA 20166
   US

   Netname: AOL-DTC
   Netblock: 205.188.0.0 - 205.188.255.255
```

```
Coordinator:
   America Online, Inc.   (AOL-NOC-ARIN)   domains@AOL.NET
   703-265-4670

Domain System inverse mapping provided by:

DNS-01.NS.AOL.COM             152.163.159.232
DNS-02.NS.AOL.COM             205.188.157.232

Record last updated on 27-Apr-1998.
Database last updated on  14-Feb-2002 19:56:41 EDT.
```

12.25.239.5

```
AT&T ITS (NET-ATT)              ATT                  12.0.0.0 -
12.255.255.255
Inflow (NETBLK-ATT137321616-232)ATT137321616-232   12.25.232.0 -
12.25.239.255
3WK Radio (NETBLK-INFLOW-12254-4841) INFLOW-12254-4841
                                                   12.25.239.0 -
12.25.239.15
```

66.38.185.143

```
GT Group Telecom Services Corp. (NETBLK-GROUPTELECOM-BLK-3)
GROUPTELECOM-BLK-3
                                                   66.38.128.0 -
66.38.255.255
Steaming Media Copr (NETBLK-GT-66-38-185-0) GT-66-38-185-0
                                                   66.38.185.0 -
66.38.185.255
```

Top Ten Internal SCAN Target Addresses Src=External

Top Ten Internal SCAN Target Addresses Src=External

Target Address	count of Target Address
10.0.151.17	8063
10.0.151.85	7990
10.0.151.105	7939
10.0.151.80	7653
10.0.151.125	6158
10.0.150.120	4755
10.0.151.70	4477
10.0.151.72	4381
10.0.151.98	4039
10.0.151.122	719

This report shows the top ten internal hosts that were scanned by external addresses. These hosts should be checked for signs of compromise.

OOS Top Talkers

For my selected date range I had a surprisingly low number of OOS events. The total count was 64. I believe that most of the packets were considered to be OOS because of invalid TCP Flag combinations. The reason for these OOS packets could be OS fingerprinting and certain types of scans like Syn-Fin scans. I will show examples of Syn-Fin scans after the top OOS talkers reports.

Top Ten External OOS Source Addresses

Top Ten External OOS Source Addresses

Source Address	count of Source Address
144.122.42.38	16
195.132.240.41	10
24.158.117.251	9
130.104.19.73	8
4.61.46.216	6
192.116.55.2	5
66.121.247.51	3
193.226.113.248	1
200.207.18.19	1
213.67.0.17	1

144.122.42.38

I would consider this address to be hostile

```
      Middle East Technical University (NET-METU-NET)
         METU Computer Center  Inonu Bulvari - ODTU
         Ankara, 06531
         TR

         Netname: METU-NET
         Netblock: 144.122.0.0 - 144.122.255.255

         Coordinator:
            METU Hostmaster  (MH2-ORG-ARIN)  hostmaster@METU.EDU.TR
            +90 312 2103330
      Fax- +90 312 2101120

         Domain System inverse mapping provided by:

         NS1.METU.EDU.TR              144.122.199.90
         NS2.METU.EDU.TR              144.122.199.93
         NS1-AUTH.SPRINTLINK.NET      206.228.179.10
         AUTH60.NS.UU.NET             198.6.1.181
```

195.132.240.41
192.116.55.2
193.226.113.248
213.67.0.17

Many Attacks originate from Europe so it would be recommended to treat these addresses as hostile.

```
European Regional Internet Registry/RIPE NCC (NETBLK-RIPE-C)
   These addresses have been further assigned to European users.
   Contact info can be found in the RIPE database, via the
   WHOIS and TELNET servers at whois.ripe.net, and at
   http://www.ripe.net/perl/whois/
   NL

   Netname: RIPE-CBLK3
   Netblock: 195.0.0.0 - 195.255.255.255
   Maintainer: RIPE

   Coordinator:
      Reseaux IP European Network Co-ordination Centre Singel 258
(RIPE-NCC-ARIN)  nicdb@RIPE.NET
      +31 20 535 4444

   Domain System inverse mapping provided by:

   NS.RIPE.NET                 193.0.0.193
   NS.EU.NET                   192.16.202.11
   AUTH03.NS.UU.NET            198.6.1.83
   NS2.NIC.FR                  192.93.0.4
   SUNIC.SUNET.SE              192.36.125.2
   MUNNARI.OZ.AU         128.250.1.21
   NS.APNIC.NET                203.37.255.97
```

24.158.117.251

I would need to do further investigation to determine if this is a hostile address.

```
Charter Communications, Inc. (NETBLK-CHARTER-NET-2BLK) CHARTER-
NET-2BLK
                                          24.158.0.0 -
24.158.255.255
Charter Communications (NETBLK-KNGPT-TN-24-158-112) KNGPT-TN-24-
158-112
                                        24.158.112.0 -
24.158.127.255
```

130.104.19.73

This is a university in France this could be a hostile address.

```
Universite Catholique de Louvain (NET-UCLOUVAIN)
   Place de l'Universite, 1
   Louvain-la-Neuve, B-1348
   BE

   Netname: UCLOUVAIN
   Netblock: 130.104.0.0 - 130.104.255.255

   Coordinator:
      Fontaine, Alain  (AF194-ARIN)  fontaine@sri.ucl.ac.be
      +32 10 472625 (FAX) +32 10 472650

   Domain System inverse mapping provided by:
```

```
NS1.SRI.UCL.AC.BE            130.104.1.1
NS2.SRI.UCL.AC.BE            130.104.1.2
NS3.SRI.UCL.AC.BE            130.104.254.1
NS.BELNET.BE                 193.190.198.10 193.190.198.2
NS2.KULNET.KULEUVEN.AC.BE    134.58.127.1

Record last updated on 22-Nov-1999.
Database last updated on  14-Feb-2002 19:56:41 EDT.
```

4.61.46.216
Genuity is an ISP I would like an explanation as to why they are sending you packets that don't conform to the RFC's.

```
GENUITY (NET-GNTY-4-0)
   3 Van de Graaff Dr.
   Burlington, MA 01803
   US

   Netname: GNTY-4-0
   Netblock: 4.0.0.0 - 4.255.255.255
   Maintainer: GNTY

   Coordinator:
      Soulia, Cindy  (CS15-ARIN)  csoulia@genuity.net
      800-632-7638

   Domain System inverse mapping provided by:

   NIC.NEAR.NET                 192.52.71.4
   VIENNA1-DNS-AUTH1.BBNPLANET.COM 4.1.16.4
   NIC3.BARRNET.NET             131.119.245.6

   Record last updated on 24-Sep-2001.
   Database last updated on  14-Feb-2002 19:56:41 EDT.
```

66.121.247.51 This is most likely a home user. It would require further investigation to determine if this address is hostile.

```
Pac Bell Internet Services (NETBLK-PBI-NET-9) PBI-NET-9
                                          66.120.0.0 -
66.127.255.255
Robert Fonvergne (NETBLK-SBCIS-10168-19618) SBCIS-10168-19618
                                        66.121.247.48 -
66.121.247.55
```

200.207.18.19
This address is from Brazil. Brazil is considered by many to be a hostile region so this address should be treated accordingly.

```
Comite Gestor da Internet no Brasil (NETBLK-BRAZIL-BLK2)
   R. Pio XI, 1500
   Sao Paulo, SP 05468-901
   BR

   Netname: BRAZIL-BLK2
```

```
Netblock: 200.128.0.0 - 200.255.255.255
Maintainer: BR

Coordinator:
   Registro.br  (NF-ORG-ARIN)  blkadm@nic.br
   +55 19 9119-0304

Domain System inverse mapping provided by:

NS.DNS.BR                    143.108.23.2
NS1.DNS.BR                   200.255.253.234
NS2.DNS.BR                   200.19.119.99

These addresses have been further assigned to Brazilian users.
Contact information can be found at the WHOIS server located
at whois.registro.br and at http://whois.nic.br

Record last updated on 30-Aug-2001.
Database last updated on  14-Feb-2002 19:56:41 EDT.
```

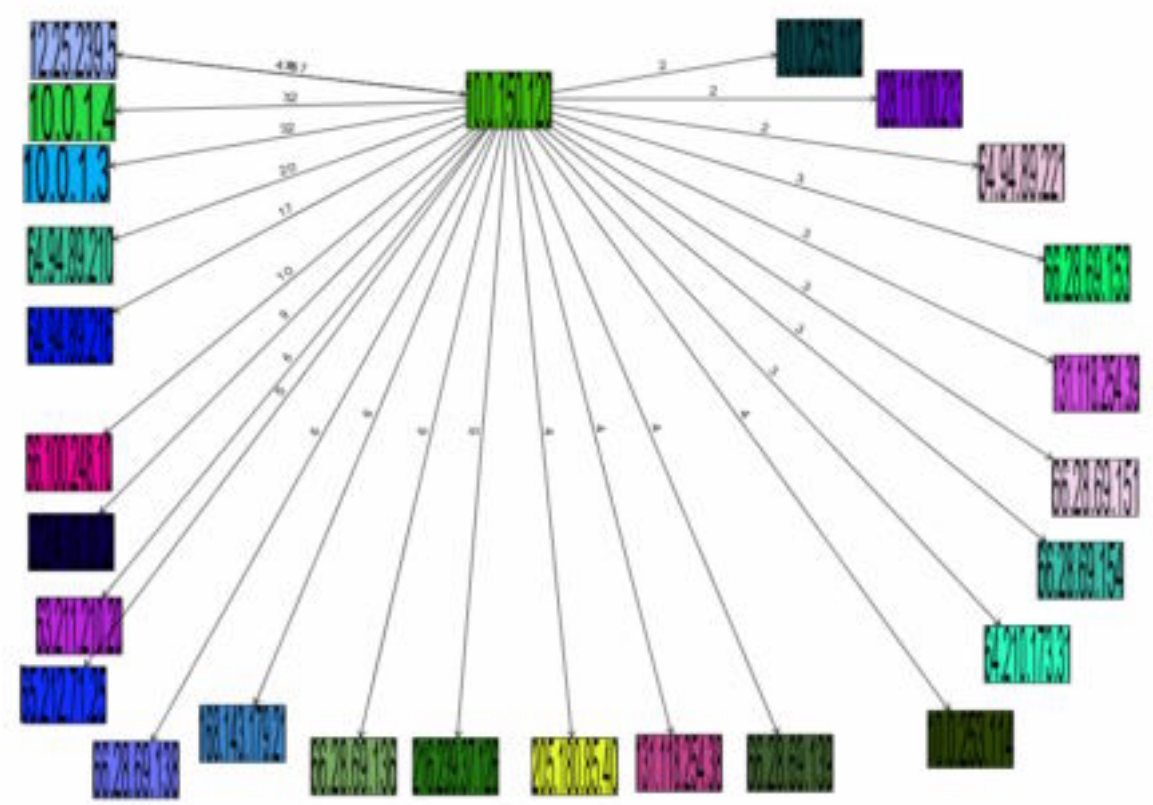
The following OOS alerts seem to be Syn-Fin scans. Note the flags:

Event Name	Detect Time	Target Address	Target Port	Source Address	Source Port	TCP Flags
OOS Event	7 Jan 2002 16:15:01 PST	10.0.88.162	1214	144.122.42.38	2718	**..SF....**
OOS Event	8 Jan 2002 16:08:46 PST	10.0.153.206	6699	192.116.55.2	1089	**..SF....**
OOS Event	8 Jan 2002 08:35:27 PST	10.0.150.204	1697	24.158.117.251	217	**..SF....**

The first one could be some one scanning for a Kazza Server that's listening. They may know of an exploit that targets hosts running Kazza.

LINK GRAPH

I decided to do my link graph on all the traffic to and from 10.0.150.120, which is a host that I have determined to be compromised. The link graph shows all the traffic that this host has had with other hosts and the direction of the traffic. From looking at graphical representations of data it gives an analyst a way to find a starting point as to where anomalous or malicious is occurring most. By looking at the graphical representation you will notice an anomalous amount of traffic between 12.25.239.5 and the compromised host. The numbers on the arrows represent the number of events in the Scan, Alert, and OOS data
files.



Brief Analysis Process

Included in the data files I chose there were over a million records. These records were scans, OOS, and snort alerts. Each type of log file was formatted differently so with out normalizing them I wouldn't have a chance at descent analysis. First I parsed out all the fields that I deemed to be important and exported them into CSV format. I did this using bits and pieces of various scripts that I found.
Once I had all the fields parsed out and converted to a CSV I then wrote a small program that would load them into a database. The first thing I had to do was to come up with a schema that would accommodate all the fields I was interested in. I used the following schema.

Event ID	Event Name	Protocol	Detect Time	Target Address	Target Port	Source Address	Source Port	Product	TCP Flags

I used the event ID field so that each event would have a distinct field that would not match any other record in the database. The event name was populated with the name of the Alert. The rest of the fields are self explanatory except for product. I used the product field to indicate the source of the alert. This field was populated with either OOS, Scan, or Snort, depending which file it came from. I also converted all the IP addresses to 10.0.*.* because I found they were much easier to work with.
Once I had all the data in a common database, I was able to issue queries and run reports across all the records I was supplied. The first thing I did was try to generate a simple topology of the network. I did this by looking for large amounts of traffic targeted to well known service ports such as port 25 for mail. I used Visio to generate the network map. I found that having an assumed topology as a reference gave me a good starting point and also helped me in determining the lethality of the alerts I was seeing. From that point on the process was mainly queries to generate the figures for my reports Queries were also issued to generate statistics as to number of events from particular sources, or count events of a particular event type targeted at a particular host.
I would like to note that the network topology was assumed based on the traffic patterns I saw in the supplied data files and that the only devices I added were a router, which I assume to be the gateway to the Internet, and the host running snort these two devices were not based on events I saw, rather it was assumed that they must be present.

References

The following websites were referenced throughout my analysis.

1. www.snort.org
2. http://www.cert.org/advisories/CA-2000-13.html
3. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0573
4. http://www.cert.org/advisories/CA-2000-13.html
5. http://www.securityfocus.com/archive/1/66544
6. http://project.honeynet.org/scans/scan19/scan/som6/timeline.xls
7. http://www.sans.org/y2k/072100.htm
8. http://www.snort.org/docs/writing_rules/chap2.html#tth_chAp2
9. http://cve.mitre.org/cgibin/cvekey.cgi?keyword=port+514
10. http://www.iss.net/security_center/static/114.php
11. http://lists.insecure.org/incidents/2001/Oct/0018.html
12. http://www.sans.org/y2k/020901-1200.htm
13. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0666
14. http://www.incidents.org/archives/intrusions/msg02798.html
15. http://www.sans.org/newlook/resources/IDFAQ/blocking.htm
16. http://www.tcpdump.org/tcpdump_man.html
17. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0917
18. http://www.kb.cert.org/vuls/id/382365
19. http://www.rdcrew.com.ar/files/rdC-LPRng.c
20. http://rr.sans.org/malicious/ramen.php
21. http://www.cert.org/advisories/CA-2000-22.html
22. http://www.cert.org/advisories/CA-2001-35.html
23. http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=EXPLOIT+ssh+CRC32+overflow
24. http://www.kb.cert.org/vuls/id/945216
25. http://www.securiteam.com/securitynews/5LP042K3FY.html
26. http://www.securityfocus.com/archive/1/161444
27. http://lists.bikkel.org/archive/whitehat/Week-of-Mon-20011105/000215.html
28. http://www.sans.org/newlook/alerts/port515.htm
29. http://www1.dshield.org/port_report.php?port=515
30. http://www.sans.org/newlook/alerts/port515.htm
31. http://xforce.iss.net/alerts/advise68.php
32. http://www.securityfocus.com/archive/96/183184
33. http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm#xtocid210315
34. http://www.alw.nih.gov/Docs/AFS/AFS_toc.html
35. http://www.cert.org/advisories/CA-2002-02.html
36. http://www.zdnet.com/products/stories/reviews/0,4161,2769395,00.html
37. http://www.sans.org/y2k/adore.htm
38. http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm
39. http://www.tlsecurity.net/backdoor/freak88.htm
40. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0288
41. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0347
42. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0673

43. http://www.kb.cert.org/vuls/id/32650
44. http://www.cert.org/incident_notes/IN-2000-02.html
45. http://www.securiteam.com/unixfocus/6U00V0035Q.html
46. http://www.microsoft.com/technet/security/bulletin/MS00-078.asp
47. http://securityresponse.symantec.com/avcenter/venc/data/dos.storm.worm.html
48. http://www.symantec.com/avcenter/venc/data/dos.storm.worm.html
49. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-078.asp
50. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0236
51. http://www.sans.org/newlook/resources/IDFAQ/subseven.htm
52. http://www.symantec.com/avcenter/venc/data/backdoor.subseven.html
53. http://www.symantec.com/avcenter/venc/data/backdoor.subseven.html
54. http://rr.sans.org/firewall/egress.php
55. http://www.theorygroup.com/Archive/Argus/1999/msg00165.html
56. http://www.ists.dartmouth.edu/IRIA/knowledge_base/tools/adorefind.htm