Global Information Assurance Certification Paper

GIAC Certified Intrusion Analyst (GCIA)
Practical Assignment
Version 4.0

Robert Perdue

September 29th, 2004

Abstract:

This paper is composed of three parts:

1. Executive Summary
2. IN-Depth Analysis
3. Analysis Process

For the analysis five logs were taken from http://isc.sans.org/logs/raw/ . These logs were named:

2002.10.9
2002.10.10
2002.10.11
2002.10.12
2002.10.13

For the IN-Depth analysis section there were three signatures evaluated.

1. IRC Nick Change
2. X11 xopen attempt
3. BAD-TRAFFIC same SRC/DST

Correlations are given throughout the document.

Part I

Executive Summary:

After analysis of five days of alerts logged at one IDS sensor there are a few items that have become apparent.

1. Your network is the target of a moderate amount of malicious traffic.
2. There are compromised boxes on the internal network.
3. Overall network security needs to be strengthened, especially at the perimeter.

The traffic that targeted your network was largely targeting the company web server. These attacks ranged from inbound worms such as Code Red to scans searching for vulnerable Internet Information Services Servers. Other traffic that was seen fell into the categories of reconnaissance and attacks against other services such as Xwindows. Publicly available services, such as Web Services, are very difficult to protect at the perimeter of a network since the public is supposed to be able to access the service. For this reason patch management and proper configuration are vital to protecting yourself from these kinds of attacks.

Contained in the logs was not only inbound attack traffic but also suspicious outbound traffic. Traffic outbound from your network was analyzed closely for signs of system compromise. Unfortunately, there is evidence of a compromised host or hosts within the protected part of your network. These compromised hosts were most likely infected by a worm or detected and exploited by a scan run from an already compromised host on the network. Once compromised Internet Rely Chat software was loaded onto the hosts which allow them to communicate to the outside world. It was the announcements of availability to the outside world using IRC that was detected in the logs. These compromised hosts present a very significant threat to the campus network and should be dealt with immediately.

Overall, judging from the data seen in the logs, there are many opportunities to strengthen your network. Industry best practices dictate that a layered approach to security is the model of choice. By layering security, all aspects of the network are looked at for possible security measures, from border routers down to user workstations. Though the challenges of securing an “edu” are understood please take under advisement the recommendations given in the Defensive Recommendations section of this report.

Part II

Section 1: Scenario

For the analysis, there were five raw logs submitted. These logs are as follows:

2002.10.9
2002.10.10
2002.10.11
2002.10.12
2002.10.13

Though the naming of the logs suggests that the data was collected from 10/9/02 through 10/13/02, this does not match the timestamps of the data contained in the logs. The data collected in the above logs spans November 8th, 2002 through November 13th, 2002.

Unfortunately, there was not any other information available about the sensor which produced the logs, the hardware and services on the network, or the network architecture. The only information presented with the logs was the following blurb:

The log files are the result of a Snort instance running in binary logging mode. This means that only the packets that violate the ruleset will appear in the log.

The logs themselves have been sanitized. All of the IP addresses of the protected network space have been "munged". Additionally, the checksums have been modified to prevent clever people from discovering the original IP addresses. You will find that certain keywords within the packets have been replaced with "X"s. All ICMP, DNS, SMTP and Web traffic has also been removed.

Section II: Analysis Overview

Fortunately, there was enough information contained within the data of the log to produce a solid picture of the campus network.

In this case Ethereal version 0.10.5a with WinPcap 3.0 was used to help determine the layout.

After loading the log into Ethereal and sorting by time stamp, a quick eyeball of source and destination IP's revealed that all the traffic was inbound or outbound to network 207.166.0.0/16. Since the IP's have been munged prior to delivery of the log files, a whois query would not produce any usable results. The owners of the 207.166.0.0 network have no relation to the events seen in the logs.

Since this network range is the only range seen in all the log traffic as either a source or destination IP, the analysis used this range as the protected network. To verify, other IP's from the log which did not belong to this network range were run through Arin and various other Internet registries. These IP's were found to belong to various public owners.

With the protected network determined, the MAC addresses were examined to get an idea of how many devices transmit traffic through the sensor which generated the log. Focusing on outbound traffic from the protected network, 207.166.0.0/16, the MAC address associated with this range, 00:00:0c:04:b2:33, was used as a starting point.

Using eth.src == 00:00:0c:04:b2:33 as an Ethereal display filter, it was found that the only outbound traffic associated with this MAC address was from 207.166.0.0/16. While examining the packets displayed with the above filter it appeared that not only was all outbound traffic coming from the same source MAC address but all the traffic was destined to a single MAC address, 00:03:e3:d9:26:c0. With these two MAC addresses identified the display filter, eth.src != 00:00:0c:04:b2:33 and eth.src != 00:03:e3:d9:26:c0, was used to see if there were any other MAC addresses sending traffic inbound or outbound. There were none found.

With the sole two MAC addresses found, that ended up in the log anyway, the next step was to see what kind of devices these were.

Even though Ethereal is nice enough to give us the manufacturers associated with these MAC addresses, they were manually referenced using the OUI list found at http://standards.ieee.org/regauth/oui/oui.txt. Using this list and Ethereal it was determined both MAC addresses belonged to Cisco devices.

00-03-E3	(hex)	Cisco Systems, Inc.
0003E3	(base 16)	Cisco Systems, Inc. 170 West Tasman Dr. San Jose CA 95134 UNITED STATES
00-00-0C	(hex)	CISCO SYSTEMS, INC.
00000C	(base 16)	CISCO SYSTEMS, INC. 170 WEST TASMAN DRIVE SAN JOSE CA 95134-1706

With the information found thus far a simple network layout was produced:

PROTECTED NETWORK	CISCO DEVICE 1	SNORT	CISCO DEVICE 2	OUTSIDE
207.166.0.0/16	00:00:0c:04:b2:33		00:03:e3:d9:26:c0	

The weakness in this diagram is that there is not much known about the protected network or the nature of the two Cisco devices. Analysis would suffer without more information.

Using the MAC OUI's an attempt was made to narrow down the possibilities of what these Cisco devices are.

While searching http://www.cisco.com an article was found pertaining to a pix software release. In this document, http://www.cisco.com/en/US/products/sw/secursw/ps2120/prod_release_note09186a00801a6d21.html , there was an example "show ver" run on a pix firewall which produced the following output(only relevant data shown):

0:ethernet0:address is 0003.e300.1552, irq 10
1:ethernet1:address is 0003.e300.1553, irq 7

The OUI's of these two pix interfaces match that of CISCO DEVICE 2 (00:03:e3:d9:26:c0). Though by no means a sure thing, with the above information and the common practice of sandwiching firewalls with NIDS, it would make sense that CISCO DEVICE 2 is the inside interface of a PIX Firewall.

The same search was performed for the MAC of CISCO DEVICE 1 but the results were inconclusive. Though best estimate would be that CISCO DEVICE 1 is a router.

With a good feeling about identifying the hardware involved in the network, attention was turned to gathering more information about the protected network. A look at the traffic inbound to the protected network, using display filter ip.dst == 207.166.0.0/16 and sorted by protocol, shows that most of the inbound traffic is for HTTP (Port 80). Though some of this traffic is due to less then honorable packets, there is enough traffic to imply the presence of this service inside the protected network.

Looking then at the outbound traffic from the 207.166.0.0/16 network using display filter ip.src == 207.166.0.0/16, it is seen that a high percentage of all outbound traffic from the protected network is from 207.166.87.157. Looking through this traffic there were outbound HTTP connections, IRC connections, and some outbound IM traffic. Most of the traffic was outbound HTTP so that was focused on for more information. Outbound requests to port 80 were focused on using a display filter of ip.src == 207.166.87.157 and tcp.dstport == 80. In these packets there are various GET requests to various web servers. In the payload of these packets it can be seen that there are various clients making these requests. Below is an export from Ethereal highlighting these findings. (Export has been edited to show only pertinent information)

Snip 1

Internet Protocol, Src Addr: 46.5.180.250, Dst Addr: 61.218.76.250
Transmission Control Protocol, Src Port: 63085, Dst Port: http (80), Seq: 512801141, Ack: 1905796758, Len: 425
Request Method: GET
*Accept: */*\r\n*
Referer: http://www.corega.com.tw/lan.htm\r\n
Accept-Language: en-us,zh-hk;q=0.7,zh-tw;q=0.3\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0; .NET CLR 1.0.3705)\r\n
Host: www.corega.com.tw\r\n
Connection: Keep-Alive\r\n

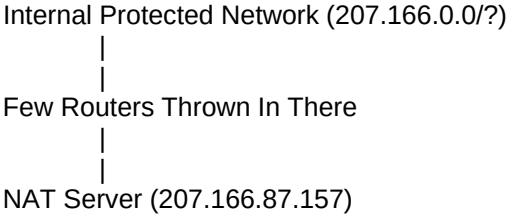
Snip 2
Internet Protocol, Src Addr: 46.5.180.250, Dst Addr: 209.225.0.6
Transmission Control Protocol, Src Port: 61962, Dst Port: http (80), Seq: 723942382, Ack: 3571025822, Len: 246
Request Method: GET
*Accept: */*\r\n*
Accept-Language: en-us\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)\r\n
Host: servedby.advertising.com\r\n
Connection: Keep-Alive\r\n

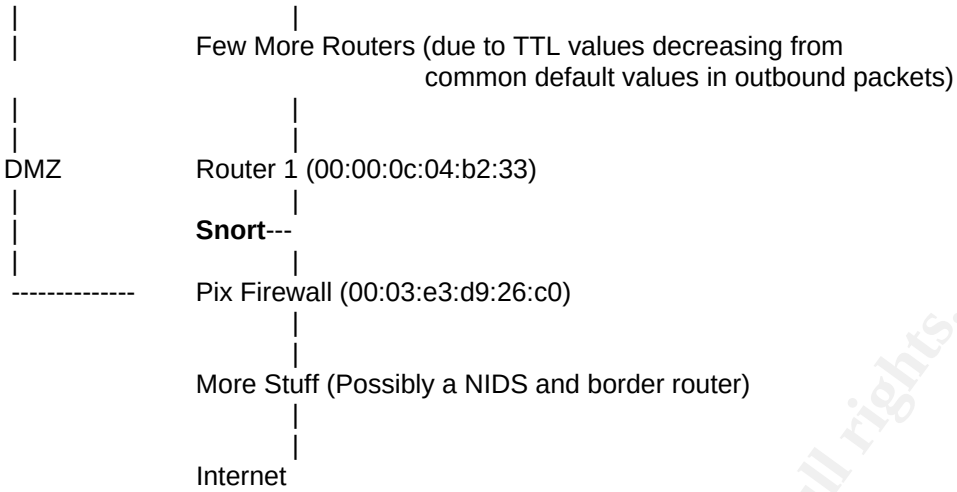
Looking at these packets shows a GET request made from a Windows 2k machine running IE 6 and a separate GET request being made from a Windows 98 box running IE 5.5.

Another clue was given in the data by examining the Time To Live values of the data sourced from 207.166.87.157. The packets with this source address had varying TTL's which suggest not only multiple operating systems but also points to 207.166.87.157 being a NAT device. A proxy would replace the TTL's of these packets, while a NAT device will leave them intact.

These findings would support that IP 207.166.87.157 is possibly a NAT device for allowing outbound traffic for internal clients on the protected network.

With this additional information the network seen in the log most likely resembles the following:





Section III: In-Depth Analysis

Network Detect 1: CHAT IRC nick change
Snort Alert:

11/11-14:27:49.616507 [**] [1:542:10] CHAT IRC nick change [**] [Classification: Potential Corporate Privacy Violation] [Priority: 1] {TCP} 207.166.87.157:61736 -> 207.44.150.220:6667

This traffic was deemed critical not because of the IRC usage but because of what it may signify. The usage of the word “r00ted” and “xdcc” raised alarms when discovered. “r00ted” is a common term used by hackers when referring to a compromised box and “xdcc” is commonly used when referring to boxes that are hosting files for distribution through IRC. In most cases the boxes hosting these files have been compromised.

What these detects show is the likely hood of compromised boxes in a protected part of the campus network.

Detect Was Generated By:

The detects used in the analysis were generated by:
-*> Snort! <*-
Version 2.1.3 (Build 27)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)

The version of Snort utilized in the analysis had all rules enabled and the stream4 preprocessor disabled.

The raw log was run through snort using the following command line:

snort -c /etc/snort/snort.conf -k none -r /logs/logfull

Options-
-c - was used to specify a configuration file
-k none - was used to ignore the munged checksums
-r was used to read in the log file

The syntax of the rule that detected the IRC usage is as follows:

alert tcp $HOME_NET any -> $EXTERNAL_NET 6666:7000 (msg:"CHAT IRC nick change"; flow:to_server,established; content:"NICK "; offset:0; classtype:policy-violation; sid:542; rev:10;)

For traffic to trigger this rule it must meet five conditions. The first condition is that the traffic must be TCP. The second condition is that the traffic must be flowing from the protected network out to an external address. The third condition is that the traffic must be sent to a destination port of 6666 though 7000. The fourth condition is that the session containing the offending payload be established, meaning the three way handshake has been completed successfully. Lastly, the content of the packet must contain the value “NICK “. Below is a sample of the offending traffic using tcpdump output with the five conditions needed to trigger an alarm in **bold**:

14:27:49.616507 IP **207.166.87.157.61736 > 207.44.150.220.6667**: P 3020898549:3020898571(22) ack 197061730 win 15407
0x0000: 4500 003e ff22 4000 7b**06** bd94 cfa6 579d E..>."@.{.....W.
0x0010: cf2c 96dc f128 1a0b b40f 40f5 0bbe ec62 .,...(....@....b
0x0020: 5018 3c2f 0e32 0000 4e49 434b 205b 6357 P.</.2..**NICK**.[cW
0x0030: 5d5b 5844 4343 5d5b 3030 3139 5d0a][XDCC][0019].

3. Probability the source address was spoofed:

I do not believe this packet to be spoofed. There appear to be compromised boxes on the protected network reporting in to their respective IRC channels. This would not be able to happen if the IP’s were being spoofed, as a successful three way handshake could never be completed using a spoofed address.

4. Description of Attack:

The IRC alarms detected in the log files appear to be the result of compromised hosts communicating back to various IRC channels. Once these compromised hosts check into their respective channels they may do something along the lines of announcing a list of files being served or just announce their presence and await further commands from an attacker.

In the logs analyzed there are only two unique nicks in use, R00teD-04 and [cW] [XDCC] [0019]. Relying solely on the nature of the nicks it is possible R00teD-04

is a compromised machine reporting in and awaiting further instructions. While [cW] [XDCC] [0019], is being used to host files accessible through the IRC network. It is important to note that this is only an assumption and to keep in mind that the true nature of these compromised boxes cannot be determined with the data present.

5. Attack Mechanism:

The IRC events seen in the logfiles are not a pure attack against the protected network. Instead, they are the result of prior successful attacks against hosts inside the protected network. As mentioned above, there is no data of the compromises in the logs that were supplied. It is however possible to outline one way this may have happened to these boxes.

1. Scan was run on the protected network looking for boxes configured with weak or no passwords. Xscan and Hscan are two tools which can be used for such scans.

2. Once these boxes are found, they are accessed using the found credentials and the desired (or undesired) programs are installed.

3. After the programs are installed the boxes are under the control of the attacker and can be used for any devious means the attacker wishes.

6. Correlations:

Similar traffic was detected in an eelier audit performed by Marcus Wu. The findings of his report can be found at:
http://www.giac.org/practical/GCIA/Marcus_Wu_GCIA.pdf

A quick overview of the why's and how's of this kind of attack can be found at:
http://www.cs.rochester.edu/~bukys/host/tonikgin/EduHacking.html

7. Evidence of Active Targeting:

Since the IRC traffic seen in the logs does not represent a specialized/specific attack against the protected network, there is no evidence of active targeting.

8. Severity:

Severity is calculated using the following formula:
severity = (criticality + lethality) - (system countermeasures + network countermeasures)

Each category is given a value between 1 (lowest) and 5 (highest).

Criticality = 4:

The importance of the compromised boxes is unknown. However, having a compromised box in a protected part of the network is severe.

Lethality = 4

There is a high probability that the IRC traffic seen in the logs point to internal compromised boxes.

System Countermeasures = 1

There is no way to determine from the log what, if any, antivirus and firewall software is running on the host(s). I will assume the worst case that there are no countermeasures running on the box and assign a value of 1.

Network Countermeasures = 1

This traffic is being allowed to and from the protected network so there may not be any network countermeasures in place. By having a host initiate communications outbound, any security given by the use of NAT is bypassed.

Using the above formula this gives a severity rating of 6.

Severity = 6

Network Detect 2: X11 xopen
Snort Alert:

11/12-08:37:32.216507 [**] [1:1226:4] X11 xopen [**] [Classification: Unknown Traffic] [Priority: 3] {TCP} 61.222.198.26:1031 -> 207.166.87.157:6000

This traffic was deemed critical because of the impact of a successful attack. This signature alerts on the attempted opening of an X Windows application from an address outside of the protected network. If this attack were successful it could lead to possible theft of data and control of the targeted machine leading to a compromise of all resources the machine is connected to. This is particularly concerning since the target IP (207.166.87.157) is that of the NAT device of the protected network. If this box were to be compromised an attacker would have an open bridge into the protected network behind the NAT device and also have access to a choke point of the network. Meaning, an attacker could sniff all traffic in and out of the protected network. Bad news.

Since the connection must be established for this signature to fire, it means that this not a simple syn scan. The X11 (tcp 6000) was found to be open and listening and a connection to port 6000 has been made.

Detect Was Generated By:

The detects used in the analysis were generated by:
-*> Snort! <*-
Version 2.1.3 (Build 27)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)

The version of Snort utilized in the analysis had all rules enabled and the stream4 preprocessor disabled.

The raw log was run through snort using the following command line:

snort -c /etc/snort/snort.conf -k none -r /logs/logfull

Options-
-c - was used to specify a configuration file
-k none - was used to ignore the munged checksums
-r was used to read in the log file

The syntax of the rule that detected the X11 usage is as follows:

alert tcp $EXTERNAL_NET any -> $HOME_NET 6000 (msg:"X11 xopen"; flow:established; content:"l|00 0B 00 00 00 00 00 00 00 00 00|"; reference:arachnids,395; classtype:unknown; sid:1226; rev:4;)

For traffic to trigger this rule it must meet six conditions. The first condition being that the traffic must be TCP. The second condition is that the traffic must be flowing from an external address to an internal address. The third condition is that the traffic must be sent to a destination port of 6000. The fourth condition is that the session containing the offending payload be established, meaning the three way handshake has been completed successfully. The fifth condition is that the content of the packet must contain the ASCII value “l” (That is a lower case L). Lastly, the content must also contain the hex string “00 0B 00 00 00 00 00 00 00 00 00 “. Below is a sample of the offending traffic using tcpdump output with the six conditions needed to trigger an alarm in **bold**:

08:55:49.206507 IP **217.7.29.146.1076 > 207.166.87.157.6000**: P 4062431032:4062431044(12) ack 1757028118 win 5840 <nop,nop,timestamp 1003604423 295968162>
0x0000: 4500 0040 0e83 4000 35**06** 63a2 d907 1d92 E..@..@.5.c.....
0x0010: cfa6 579d 0434 1770 f223 c738 68ba 1f16 ..W..4.p.#.8h...
0x0020: 8018 16d0 8395 0000 0101 080a 3bd1 c9c7 ....;...

0x0030: 11a4 1da2 6c**00 0b00 0000 0000 0000 0000** ....**l**.....

3. Probability the source address was spoofed:

I do not believe these packets to be spoofed. In order for this connection to be established legitimate IP's must be used.

4. Description of Attack:

This attack appears to have three phases. The first phase is reconnaissance, before a connection can be made to the X11 service it must be found running on a live host on the network. This can be done with many tools, such as Nmap. Once a host is found with X11 running an attempt is made to connect to this service. X11 is not easy to configure properly and securely, in the case of a misconfiguration an attacker can connect to the target with relative ease. Aside from possible misconfigurations there are many vulnerabilities with various implementations of X11 which may be used as other attack vectors.

5. Attack Mechanism:

In the case of the X11 xopen attack there is no special mechanism in use . The attack is able to proceed due to a lack of perimeter security and improper host configuration.

The signatures fired in this case only show an attempt to launch an Xwindows application on an Xwindows server. This may by a sign that a more serious attack may occur.

There is a plethora of CVE's pertaining to X11 vulnerabilities, though they may not pertain to this specific implementation, this list should be consulted with your X11 implementation in mind.

http://www.cve.mitre.org/cgi-bin/cvekey.cgi?keyword=x11

6. Correlations:

Various CVE's for X11 implementations can be found at:
http://www.cve.mitre.org/cgi-bin/cvekey.cgi?keyword=x11

A brief explanation of this violation and the Snort rule that detected it can be found at:
http://www.snort.org/snort-db/sid.html?sid=1226

7. Evidence of Active Targeting:

There is strong evidence that this attack was specifically targeted. In the five days worth of logs there is no evidence of recon looking for X11 servers. The NAT server was already known to have a X11 service available and was deliberately targeted for attack.

8. Severity:

Severity is calculated using the following formula:
severity = (criticality + lethality) - (system countermeasures + network countermeasures)

Each category is given a value between 1 (lowest) and 5 (highest).

Criticality = 5:

The target box is a gateway to the protected network and also a choke point for large amounts of network traffic. If this box were to be compromised there could be severe consequences.

Lethality = 3

It is not possible to tell from the logs if this was a successful attack. These signatures alone do not represent an attack but strongly point to future attempts of exploitation which may lead to system compromise.

System Countermeasures = 1

There is no way to determine from the log what, if any, antivirus and firewall software is running on the target box. I will assume the worst case that there are no countermeasures running on the box and assign a value of 1.

Network Countermeasures = 1

Since a connection was made to the target from and external address, there does not appear to by any Network Countermeasures in place.

Using the above formula this gives a severity rating of 6.

Severity = 6

Network Detect 3: BAD-TRAFFIC same SRC/DST
Snort Alert:

11/10-20:02:51.796507 [**] [1:527:8] BAD-TRAFFIC same SRC/DST [**] [Classification: Potentially Bad Traffic] [Priority: 2] {IGMP} 207.166.71.243 -> 207.166.71.243

This traffic was deemed critical for two reasons. The first reason is the nature of the attempted attack. This was not an attack aimed at a particular host or service, this was an attack which was aimed at the network infrastructure through the targeting of possible routers. Though the exact nature and purpose of this attack is unknown it is believed this is an attempted Denial of Service attack against your network. The second cause of concern is the fact that this traffic was allowed into the network using source IP's from the protected network range. This points to a deficiency in perimeter security.

Detect Was Generated By:

The detects used in the analysis were generated by:
-*> Snort! <*-
Version 2.1.3 (Build 27)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)

The version of Snort utilized in the analysis had all rules enabled and the stream4 preprocessor disabled.

The raw log was run through snort using the following command line:

snort -c /etc/snort/snort.conf -k none -r /logs/logfull

Options-
-c - was used to specify a configuration file
-k none - was used to ignore the munged checksums
-r was used to read in the log file

The syntax of the rule that detected the bad traffic is as follows:

alert ip any any -> any any (msg:"BAD-TRAFFIC same SRC/DST"; sameip; reference:bugtraq,2666; reference:cve,1999-0016; reference:url,www.cert.org/advisories/CA-1997-28.html; classtype:bad-unknown; sid:527; rev:8;)

For traffic to trigger this rule it must meet two conditions. The first condition being that the traffic must be IP. The second condition is that the source and destination addresses must the same value.. Below is a sample of the offending traffic using tcpdump output with the two conditions needed to trigger an alarm in **bold**:

20:02:51.796507 **IP 207.166.71.243 > 207.166.71.243**: igmp query v2 [gaddr 240.0.3.126]
0x0000: **45**00 001c 0000 0000 2f02 f141 cfa6 47f3 E...../..A..G.
0x0010: cfa6 47f3 1164 fb1c f000 037e 0000 0000 ..G..d.....~....

0x0020: 0000 0000 0000 0000 0000 0000 0000 ....

3. Probability the source address was spoofed:

There is a very high probability that the source IP's have been spoofed. Traffic with the same source and destination IP address should not been seen in regular network traffic. Also, looking at the source and destination MAC addresses of these packets the traffic appears to be originating from outside the protected network. Spoofing the source address as an address from the protected network is also a technique which can be applied to avoid lose or poorly written firewall rules.

4. Description of Attack:

With the information at hand, the type of attack seen in these packets cannot be determined. However, it is possible to discuss possibilities.

It is likely that these packets were an attempt at a Denial of Service attack against the network. This can be said for the following reasons:
1. The source IP's are spoofed so normal two way communication cannot take place. These packets are only concerted about getting to a target not getting back.
2. The spoofed IP's match the destination IP's. This may be used to "confuse" the target.
3. The packets do not conform to the specifications for IGMP traffic given in http://www.ietf.org/rfc/rfc2236.txt. The TTL is not set to 1, multicast MAC addresses are not used, and the multicast group numbers are from a class E address range, not a class D. The mangling of the RFC specifications may be in an attempt to crash an IGMP enabled host.

There is a CVE for Microsoft Windows, in which a malformed fragmented IGMP packet can cause a DoS condition. Though these packets are not fragmented, this could be a new attack or someone testing new exploit code.

This may also be an attempt to DoS querying routers. When a querying router receives a membership query, a computation is performed to see if the source of the query has a lower IP than the receiving router. If the source of the query has a lower IP value than the destination, than the lower IP becomes the querying router. Perhaps it is possible that weaker implementations of IGMP do not take into account the possibility of two IGMP querying routers having the same IP which may cause an undesirable condition.

5. Attack Mechanism:

The attack mechanism is hard to determine. This could be the product of a new/experimental tool, an existing IGMP Nuke tool, or just good old fashioned packet crafting using tools such as Hping.

If this traffic did reach any live hosts, if available to logs of those hosts should be analyzed to determine any impact from these packets.

6. Correlations:

Various CVE's for IGMP implementations can be found at:
http://www.cve.mitre.org/cgi-bin/cvekey.cgi?keyword=igmp

A brief explanation of this violation and the Snort rule that detected it can be found at:
http://www.snort.org/snort-db/sid.html?sid=527

The CVE for traffic with the same source and destination can be found at:
http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0016

This traffic was also discussed in a GCIA paper written by Michael Bernstein:
http://cert.uni-stuttgart.de/archive/intrusions/2004/01/msg00100.html

Finally, a paper discussing IGMP attacks by Josep Blanquer and Robert Chalmers and be found at:
http://www.nmsl.cs.ucsb.edu/~blanquer/papers/internal/igmp.ps

7. Evidence of Active Targeting:

With the information at hand it is difficult to determine if this was a targeted attack or not. If the IP's targeted in this attack belong to routers on the network than this would be a targeted attack by someone who knows a good deal about your network. The other possibility is that these packets were sent to random IP's in hopes of hitting a router.

8. Severity:

Severity is calculated using the following formula:
severity = (criticality + lethality) - (system countermeasures + network countermeasures)

Each category is given a value between 1 (lowest) and 5 (highest).

Criticality = 3:

I believe these packets were intended to attack routers, which are a critical piece of network infrastructure. However, since this cannot be said for sure a value of 3 will be given.

Lethality = 3

It is unknown at this point what effect this attack had on the network, or the impact of future attacks. Until results can be mined from logs during the first attack, the lethality of this attack will remain unknown. It is this unknown factor which will keep the score in the middle of the road at 3.

System Countermeasures = 1

There is no way to determine from the log what, if any, antivirus and firewall software is running on the targeted boxes. I will assume the worst case that there are no countermeasures running on the box and assign a value of 1.

Network Countermeasures = 1

Since traffic was permitted from outside of the protected network using spoofed IP's that matched the protected network range, there does not appear to by any Network Countermeasures in place.

Using the above formula this gives a severity rating of 4.

Severity = 4

Section IV: Network Statistics

Top Talkers:

Top Five Source IP's				
Source IP	**DNS Name**	**Total Events**	**Unique Signatures**	**Destination Addresses**
207.166.87.157	Unknown	4632	14	87
255.255.255.255	Unknown	211	1	211
192.77.15.39	esd39.ihs.com	80	2	1
211.47.255.22	Unknown	48	1	3
211.47.255.24	Unknown	48	1	3

The chart above represents the top five "talkers" seen in the logs. These top five represent the top volume producing IP's which required further analysis. Common reconnaissance events such as proxy scans were omitted.

The first IP seen is that of the internal NAT device of the protected network. Though this may not be a surprise as all traffic leaving the network from behind

this device will have 207.166.87.157 as its source IP, the volume of alerts should be a focus. A break down of these events has shown that 44% of the traffic attributed to 207.166.87.157 can be traced to Bare Byte Encoding events. Bare Byte Encoding is an IIS trick that uses non-ASCII chars as valid values in decoding utf-8 values, this is not in the HTTP standard. During analysis of these events it appears this traffic is due to two main contributors. One is the transfer of data, possible cookie information, to external IP's which belong Hitbox.com. This can be because of the presence of spyware within the protected network. The other main reason for the high-count of these events is due to what appears to be the transfer of images to www.imagestation.com.

The second IP in the list 255.255.255.255 is more of a concern than the outbound Bare Byte Encoding. This source IP is defiantly spoofed, it is a broadcast address and should not be allowed into the protected network. There was only one signature fired by this source address, [arachNIDS][snort] BACKDOOR Q access. Q is a UNIX backdoor, which when installed offers a channel for remote communications to an attacker. In the logs there were 211 unique destination IP's for this signature, this does not indicate a targeted attack. Instead this may be a scan to probe for previously compromised UNIX boxes. These scans have been seen running every day included in the logs. This attack has been well documented in previous audits. Please refer to, http://www.giac.org/practical/GCIA/Al_Maslowski-Yerges_GCIA.pdf, for further explanation and possibilities for this traffic.

Sample of BACKDOOR Q traffic:

```
20:40:15.136507 IP 255.255.255.255.31337 > 207.166.64.154.printer: R 0:3(3)
ack 0 win 0
        0x0000:  4500 002b 0000 0000 0f06 e5d7 ffff ffff  E..+............
        0x0010:  cfa6 409a 7a69 0203 0000 0000 0000 0000  ..@.zi..........
        0x0020:  5014 0000 9aff 0000 636b 6f00 0000       P.......cko...
```

Next on the list is 192.77.15.39 (esd39.ihs.com). This source IP triggered two unique alerts, [snort] (http_inspect) NON-RFC HTTP DELIMITER and [arachNIDS][snort] WEB-MISC http directory traversal. At first glance these alarms seemed, well…alarming. In the traffic was seen groups of HEAD requests for various files on your company web server (207.166.87.40), these head requests were than followed by GET requests for these files. This appeared to be scan and access attempts to potentially confidential information. Upon investigation it was found through ARIN that the source IP belonged to:

OrgName: Information Handling Services
OrgID: IHS-7
Address: 15 Inverness Way East
City: Englewood
StateProv: CO
PostalCode: 80112
Country: US

NetRange: 192.77.15.0 - 192.77.15.255
http://www.ihs.com

It turns out that IHS is a aggregator of technical documents. This company maintains a list of technical documents from venders of all types. This allows a user to use one point to access thousands of technical documents. With this in mind, the traffic was looked at again. It is common for caching devices, such as proxies, to submit HEAD requests to verify the data in its cache. If the HEAD request does not return a valid response a GET request is sent to update the cache. I believe this is what is happening with these alerts. The Directory Traversals fired on the ../ string found in the HEAD and GET requests. This would be normal if this is how the files are to be accessed on the web server. Though I believe this to be a false positive, this traffic should be verified.

Sample of traffic:

```
19:15:01.266507 IP 192.77.15.39.50656 > 207.166.87.40.http: P
1928345024:1928345099(75) ack 3903857153 win 8760
        0x0000:  4500 0073 caab 4000 ef06 14e0 c04d 0f27  E..s..@......M.'
        0x0010:  cfa6 5728 c5e0 0050 72f0 35c0 e8b0 2201  ..W(...Pr.5...".
        0x0020:  5018 2238 de9b 0000 4845 4144 202f 6d61  P."8....HEAD./ma
        0x0030:  696e 2f2e 2e2f 6d61 696e 2f64 6174 6173  in/../main/datas
        0x0040:  6865 6574 732f 7669 6374 6f72 7962 7836  heets/victorybx6
        0x0050:  362e 7064 6620 4854 5450 2f31 2e31 0a48  6.pdf.HTTP/1.1.H
        0x0060:  6f73 743a 2077 7777 2e58 5858 5858 5858  ost:.www.XXXXXXX
        0x0070:  580a 0a                                  X..
```

The last two IP's on the list, 211.47.255.22 and 211.47.255.24, will be discussed together as they have the same owner and are responsible for the same type of traffic. The traffic seen from these two IP's fired the alarm, [snort] BAD-TRAFFIC tcp port 0 traffic. This alarm will fire when traffic is destined for TCP Port 0. TCP traffic to port zero should not be seen during normal network communications, this could indicate attempted reconnaissance. Further investigation of this traffic showed that the IP's in question were allotted to the Korea Network Information Center, see below.

inetnum: 211.46.0.0 - 211.49.255.255
netname: KRNIC-KR
descr: KRNIC
descr: Korea Network Information Center
country: KR

The KNIC is a Korean Registry much like Arin. A whois query of KNIC did not return any owner of the above IP range.

This traffic was seen on a daily basis in the logs, there were two other IP's generating this traffic as well which belonged to the same network range (211.46.255.20 and 211.46.255.21).

Top Attacked Ports:

Top Targeted Ports	
Target Port	**Total Events**
80	352
515	211
0	141
64052	20
6000	5

The table above shows the most targeted ports from inbound traffic based purely on event count, though ports 8080 and 3128 have been excluded due to extremely high counts from proxy scans.

It is no surprise that port 80 was the most targeted port in the five days worth of logs analyzed. HTTP, which commonly runs on port 80, is one of the most actively attacked services. In the case here a majority of the events targeting port 80 were from the document scan run by IHS.com which was discussed earlier. There was also a fair amount of events from inbound IIS scans and inbound Code Red.

The next two ports in the table, 515 and 0, were also discussed in detail in the previous section. The high values of attacks on ports 515 and 0 were due to a Q Backdoor scan and what appears to be a host scan, respectively.

When putting this list together, the first three ports were to be expected due to the volume of traffic seen attacking those ports during earlier analysis. Then there was port 64052, which stuck out like a sore thumb. This is a very high ephemeral port which one would not expect to see a lot of attacks targeted against. Upon investigation it was found that these twenty events were from a single source IP, 64.37.156.27(eqftp.station.sony.com) and source port (39326) and only fired one unique signature which was [arachNIDS][snort] SHELLCODE x86 NOOP. This signature fires on a series of NOP instructions (Hex value of 90), for the x86 architecture. The strings of NOP's are sometimes referred to as NOP sleds and can be seen in some buffer overflow style attacks.

During investigation of the source IP it was found to be owned by:

OrgName: Verant Interactive
OrgID: EQ
Address: 8958 Terman Court
City: San Diego
StateProv: CA
PostalCode: 92121
Country: US
NetRange: 64.37.128.0 - 64.37.191.255

Verant Interactive, it turns out hosts the www.sonyonline.com website which is dedicated to what looks like online gaming and entertainment. In turn it is Sonyonline which maintains the site run on the source address seen in the offending traffic. When 64.37.156.27(eqftp.station.sony.com) was then investigated it was found http://eqftp.station.sony.com was not much more than an informational website. However, guided by the DNS name eq**ftp**.station.sony.com ftp://eqftp.station.sony.com was accessed. This turned out to be an accessible anonymous ftp server hosting various exe files. It is the downloading of these files through a passive FTP connection (See http://slacksite.com/other/ftp.html for a description of passive FTP) which is believed to have triggered the SHELLCODE signatures. This evaluation was reinforced by the following false positives in the description of the Snort rule which flagged this traffic (http://www.snort.org/snort-db/sid.html?sid=648): *The x86 NOP can frequently be found in day-to-day traffic, particularly when transferring large files.*

Lastly, is port 6000. This traffic was seen from the X11 attacks described in detail in the previous section of this report.

Suspicious External IP's:

Top Suspicious IP's				
Address	DNS Name	Total Events	Unique Signatures	Destinations
172.20.10.199	Unable to resolve address	9	1	9
200.200.200.1	Unable to resolve address	26	1	7
192.9.100.154	Unable to resolve address	1	1	1

172.20.10.199:
Signature Fired: [snort] (snort_decoder) WARNING: TCP Data Offset is less than 5!
Registration Information:
OrgName: Internet Assigned Numbers Authority
OrgID: IANA
NetRange: 172.16.0.0 - 172.31.255.255
CIDR: 172.16.0.0/12
NetName: IANA-BBLK-RESERVED
NetType: IANA Special Use
Comment: This block is reserved for special purposes.
Comment: Please see RFC 1918 for additional information.

Probable OS: According to Lance Spitzner's list of OS footprints (http://honeynet.spenneberg.org/papers/finger/traces.txt) it appears the source of this traffic could be a Cisco box running IOS 12.0. This is based on an IP ID of 0 and a TTL of 234 and 235 and most packets having the Don't Fragment Bit set to zero.

Suspicious Traits:

1. According to RFC 1918 The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:
   10.0.0.0 - 10.255.255.255 (10/8 prefix)
   172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
   192.168.0.0 - 192.168.255.255 (192.168/16 prefix)
The source IP falls within the second group.

2. All packets have the RST flag set.
3. All packets have unique destinations.
4. These packets are seen every day in the five days examined.
5. One sole packet has the Do Not Fragment bit set.

This could be a recon tactic (though doubtful), it is more likely the result of your IP range being spoofed as a source IP in another attack and the reset packets are the response. Though this traffic warrants further monitoring and investigation to be certain.

200.200.200.1:
Signature Fired:[snort] BAD-TRAFFIC ip reserved bit set
Registration Information:
inetnum: 200.128/9
status: allocated
owner: Comite Gestor da Internet no Brasil
ownerid: BR-CGIN-LACNIC

Probable OS: Very unsure due to the unusual traits of the packet. Going by the TTL of 224 and an IP ID of 0, best guess is a UNIX based box.

Suspicious Traits:
1. Normal IP traffic does not use the reserved bit.
2. Packets were seen on the 10th and again on the 13th, destined for different IP's.
3. All packets have a data offset of 17184 and an ID of 0.

From the Snort rule description (http://www.snort.org/snort-db/sid.html?sid=523); This may be an indicator of the use of the reserved bit by a malicious user to instigate covert channel communications, an indicator of unauthorized network use, reconnaissance activity or system compromise. These rules may also generate an event due to improperly configured network devices.

This traffic should be monitored and examined more thoroughly.

192.9.100.154:
Signature Fired:[snort] MISC Tiny Fragments
Registration Information:
OrgName: Sun Microsystems, Inc

OrgID: SUN
NetRange: 192.9.10.0 - 192.9.199.255

Probable OS: Unsure due to the very unusual traits of the packet. Going by the TTL of 235 and an IP ID of 0, best guess is a UNIX based box, possibly Solaris.

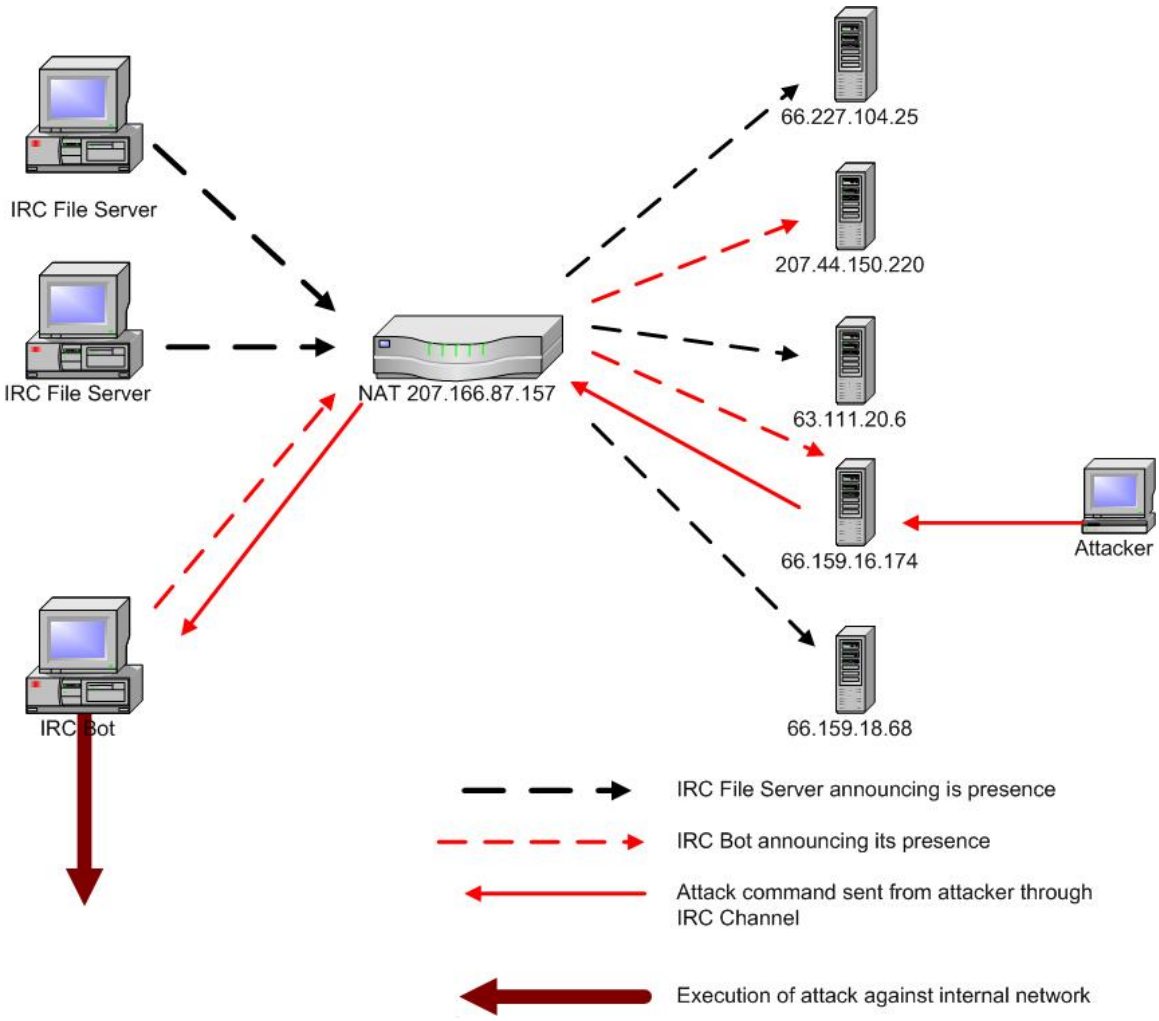
Suspicious Traits:
More Fragments Bit set, yet this fragment is less than 25 bytes.

From the Snort rule description (http://www.snort.org/snort-db/sid.html?sid=522);
Many Ides are known to have issues regarding the reassembly of IP fragments, and could miss an attack carried over such means. Firewalls suffer from the same issues, and can be tricked into allowing packets through that should normally be rejected. Furthermore, there is a small history of OS issues related to unorthodox fragmentation.
This traffic should be monitored and examined more thoroughly.

This packet could represent the use of Dug Song's fragrouter program.

Evidence of Compromised Computers:

While examining the events found in the logs there was very strong circumstantial evidence of compromised systems on the protected network. This evidence was seen in the IRC Nick Change events discussed earlier in the report. Please refer to this discussion for detail on this attack. Unfortunately, the placement of the sensor which produced the logs is outside of the NAT device, this placement obfuscates the true originating IP address of this traffic. In order to find the true IP's of the compromised systems there are a few options: a) examine logs of NAT device b) If one exists, examine logs of NIDS behind NAT device c) Perform a sniff of network traffic just inside the NAT device for destination port 6667. This can be done using tcpdump and a filter of "dst port 6667". Below is a link graph showing possible traffic due to the compromised host(s).



Defensive Recommendations:

After review of the logs supplied the biggest challenge facing the Campus Security team is to strengthen the perimeter security. Though securing an "edu" can be difficult there are steps that can be taken to reduce the amount of malicious traffic that gets into the protected network.

One area that should be focused on is the Ingress and Egress filtering applied on the network. In the logs there is evidence of packets entering the network using the IP's which belong to the Campus' network. As a general rule address ranges designated by IANA for the use of private internets (RFC 1918), bogon address, and your own owned address block should not be allowed into the network if they are used as source addresses.

This could be a bit tricky for the Campus environment, but firewall rules should start with a "deny all" posture and only open what is needed from that point. One of the more serious attacks seen in the log targeted port 6000 (X11) on the NAT

device of the network. Access to any non-public service should not be allowed into the protected network.

Compartmentalize. Especially in such a hostile environment as an "edu" campus, the network should be broken down into compartments, each with its own security measures such as firewalls. There is strong evidence in the logs to suggest compromised hosts inside the protected network. Through the use of these compromised hosts it is possible for attackers to penetrate deeper into the campus network. Segmenting resources on the network may aid in containing the possible damage.

Lastly, a strong effort should be placed on patch management. Not so much on student computers, as that may consume too many resources, but focus on school owned equipment. For example, there is evidence in the logs that the campus web server (207.166.87.40) is running Apache 1.3.12 on RedHat Linux. If this is the true version of the server, it is outdated and contains vulnerabilities and should be updated. The patch level of the OS should be checked as well. A strong effort in making sure systems are up to date will greatly reduce the risks of successful attacks.

Part III

Analysis Process:

Hardware	OS	Software
IBM T22 Thinkpad 264 Megs RAM 20 Gig Harddrive 900Mhz PIII	Win2k Workstation	Ethereal 10.6 WinPcap 3.1 beta Putty .54 IE 6.0.2800.1106
IBM 300GL 512 Megs RAM 20 Gig Harddrive 600Mhz PIII	Fedora Core 2 (2.6.6-1.435.2.1)	Snort 2.1.3 ACID v0.9.6b23 MySQL 2.1.3 Apache 2.0.49 Tcpdump 3.8

The Fedora box was setup strictly as an IDS system. This was done by following the instructions created by Patrick Harper which can be found at: http://www.internetsecurityguru.com/documents/Snort_SSL_FC1.pdf

Once a box is setup using the above instructions it can be accessed through SSH using a client such as Putty. Once configured all access to the Fedora box was through an SSH connection from the Win2k Workstation. This allowed access to many resources from just one machine.

IMPORTANT NOTE: This instruction set is for an installation of Fedora Core 1, if Core 2 is to be used please refer to Patrick Harpers website (http://www.internetsecurityguru.com/) for a link to updated RPM's which will work with Core 2.

Since there were five log files analyzed for this report, the first challenge was to create one large log file. This was easily done using Mergecap, a utility which is installed with Ethereal and created just for this purpose. An example of the usage of mergecap would be: C:\>mergecap –w logfull 2002.10.9 2002.10.10

or for a more general usage: C:\>mergecap -w <outfile> <infile> [...]

Once the master log was created it was moved to the Fedora box to be pumped through Snort. For this the following command line was used to have Snort analyze the log:
snort -c /etc/snort/snort.conf -k none -r /logs/logfull
Options-
-c - was used to specify a configuration file
-k none - was used to ignore the munged checksums
-r was used to read in the log file

IMPORTANT NOTE: By default Snort will run with the Stream4 preprocessor enabled. Since the raw log is only composed of offending packets of a communication stream, Snort did not recognize the flow to the target as established. As a result a very high percentage of packets in the raw logs were ignored. To increase the effectiveness of Snort the Stream4 preprocessor was disabled in the snort.conf file by placing a # in front of the following two entries:
preprocessor stream4_reassemble
preprocessor stream4: disable_evasion_alerts

Once the log was run through Snort, all the alerts were logged into the MySQL database which was setup during the configuration on the Fedora box. From there the Analysis Console for Intrusion Databases (ACID) was used to view and query the alerts. The analysis console is accessible as a webpage and in this case was accessed using IE 6 from the W2k workstation. The ACID interface was invaluable during analysis.

All alerts found using ACID and Snort (that just doesn't sound right), were then investigated thoroughly using Ethereal and Tcpdump and evaluated using the network diagram from Part II of the report.

IMPORTANT NOTE: If Antivirus software is running on any computer that is used to analyze these raw log files, *KEEP THAT IN MIND*! Do not be like this analyst and go crazy for an hour, or six, wondering why log files would mysteriously disappear after attempting to load them into Ethereal. Contained in these log files is data which will be picked up by Antivirus software and,

depending on your configuration, proceed to delete the log files. Also helps if your Antivirus software is configured to alert.

References:

Standards:

http://standards.ieee.org/regauth/oui/oui.txt

http://www.ietf.org/rfc/rfc2236.txt

CVEs:

http://www.cve.mitre.org/cgi-bin/cvekey.cgi?keyword=x11

http://www.cve.mitre.org/cgi-bin/cvekey.cgi?keyword=igmp

http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0016

Snort Rules:

http://www.snort.org/snort-db/sid.html?sid=1226

http://www.snort.org/snort-db/sid.html?sid=527

http://www.snort.org/snort-db/sid.html?sid=648

http://www.snort.org/snort-db/sid.html?sid=523

http://www.snort.org/snort-db/sid.html?sid=522

GIAC Practicals:

http://www.giac.org/practical/GCIA/Marcus_Wu_GCIA.pdf

http://cert.uni-stuttgart.de/archive/intrusions/2004/01/msg00100.html

http://www.giac.org/practical/GCIA/Al_Maslowski-Yerges_GCIA.pdf

Snort Installation:

http://www.internetsecurityguru.com/documents/Snort_SSL_FC1.pdf

Miscellaneous:

http://www.cisco.com/en/US/products/sw/secursw/ps2120/prod_release_note09186a00801a6d21.html

http://www.cs.rochester.edu/~bukys/host/tonikgin/EduHacking.html

http://www.nmsl.cs.ucsb.edu/~blanquer/papers/internal/igmp.ps

http://slacksite.com/other/ftp.html

http://honeynet.spenneberg.org/papers/finger/traces.txt