Jamil Farshchi
Intrusion Detection In Depth

GCIA Practical Assignment, v3.0

Table of Contents

1.0 A Statistical-Based Approach to Intrusion Detection

Introduction

Network Intrusion Detection Systems (IDS) monitor computer network traffic and attempt to identify, alert, and present all anomalous activity to the user. The basic premise is that if a transmission is not allowed on the network, the IDS will have the ability to recognize and report the illegitimate traffic. The key to any Intrusion Detection System is to maximize accurate alerts (true-positive) while at the same time minimizing the occurrence of non-justified alerts (false-positive). This is much easier in theory than in practice, as attested by the variety of intrusion detection methods. These methods include but are not limited to Artificial Immune System [7], Control-Loop Measurement [8], Data Mining [9], Statistical [24], and Signature-Based (Rule-Based [25]). The most popular of these methods is Signature-Based Intrusion Detection. While there are many approaches to intrusion detection, this document specifically focuses on Statistical-Based Intrusion Detection Systems, Spade, and the deployment of Spade in concurrence with a current IDS.

Signature-Based systems

Some of the more popular Signature-Based IDS's are NFR [11], RealSecure [12], Dragon [13], Snort [14], and Cisco Secure IDS [15]. It has been shown that Signature-Based Intrusion Detection has many benefits, such as the potential for low alarm rates, accuracy of detection, and detailed textual logs [4]. With verbose signatures, it is relatively simple to specifically identify packets of interest. For example, it would be trivial to write a rule to alert on all TCP packets with the SYN flag set. Not all IDS's allow independent rule development, but some, like Snort and Dragon, accept user created rules. Nearly all IDS vendors provide rules for their products with variable numbers of signatures, usually in the range of 500-1500+ rules. Rules are developed over time as the security community identifies new vulnerabilities and scanning techniques. The extensiveness and speed with which these rules are developed by the vendor is a good benchmark for how effective the IDS will ultimately be. While the Signature-Based approach to intrusion detection is acceptable, it leaves much to be desired. With vendors coming out with new signatures on a weekly or daily basis it is difficult for an already overburdened security professional to keep up to date with the latest rule sets. A far more serious shortcoming of the Signature-Based IDS approach is the inability to detect new and previously unidentified attacks. A Signature-Based IDS is only as strong as its rule set, and if the attack is new, there will simply not be any signatures developed to identify the probe. Signature-Based Intrusion Detection also has a limited ability to detect port scanning. In fact, most IDS's use the rudimentary approach, whereby, if X events of interest are detected across a Y-sized time window [16], the system will generate an alert. By limiting the number of packets targeted at a network over a specified time frame, an attacker can easily escape detection by the IDS. These deficiencies are inherent in the Signature-Based model, which is why different methods of detection are needed to address the inadequacies of the Signature-Based approach.

An Introduction To The Statistical Approach

Statistical-Based Intrusion Detection Systems (SBIDS) can alleviate many of the aforementioned pitfalls of a Signature-Based IDS. Statistical-Based systems rely on statistical models such as the Bayes' Theorem [26], to identify anomalous packets on the network. To identify an anomaly, the system uses data compiled from previous network behavior. Since warnings are based on actual usage patterns, statistical systems can adapt to behaviors and therefore create their own rule usage-patterns. The usage-patterns are what dictate how anomalous a packet may be to the network. Anomalous activity is measured by a number of variables sampled over time and stored in a profile. Based on the anomaly score of a packet, the reporting process will deem it an alert if it is sufficiently anomalous; otherwise, the IDS will simply ignore the trace. The reporting process will alert the user if the packet's anomaly score is greater than or equal to the threshold level set by the user. So, the SBIDS identifies and tracks patterns and usage of the network data and then assigns an anomaly score to each packet. Once this is accomplished, the reporting facility will generate an alert if the anomaly score is greater than the alert threshold. As an example, let's say that every morning, you wake up and read the morning paper that is waiting outside the door. After a few days or weeks of this behavior, it becomes normal; you expect the paper to arrive at the door in the morning. One morning, the paper is not waiting at the doorstep. Instead, the paper is lying in the driveway. This is not normal; it is clearly anomalous activity, but probably not enough to warrant investigation. Now, let's say you continue to see approximately the same pattern of a few papers landing on the driveway every week. Then, one day, you wake up to no paper at all, or even worse, the paper is thrown through the window. Neither of these events is normal, and both would warrant some degree of investigation. If an anomaly number is associated with these events, we can begin to see how a SBIDS works. The action of receiving a paper at the door in the morning would be deemed "normal" activity. The system would recognize the pattern and learn that this is normal behavior. Other activities would be judged based on the number of occurrences and how "unique" they were in relation to normal activity. The importance of the threshold level is shown in this example as well. If the threshold is set to a low number, the SBID would have generated an alert for any discrepancy from the norm, so there would have been an alert produced when the paper landed on the driveway. If set it too high, an alert would be created only when the paper broke through the window (and maybe not even then). Optimally, a report will be generated on all significant anomalous activity. What constitutes "significant" can and will vary from user to user. Therefore, it is ultimately up to the user to decide how many alerts are generated for a specific environment. The particular environment is crucial to the proper functioning of a SBIDS. The SBIDS will "learn" what is "normal" for a network. Each Statistical-Based IDS in every individual environment will alert to discrepancies based on its specific knowledge of the network at hand. The benefit of this approach is that the system does not have to have predefined signatures to identify an anomaly on the network; instead, the IDS is free to flag anything it deems unusual. For example, H4x0r has a brand new exploit she wants to use on the network. She launches the attack knowing that there is no signature for this exploit because the vulnerability was found recently. If one of the systems is exploitable by the attack, it will be compromised and no alert will be generated because a Signature-Based IDS will not recognize this new attack (signature). If, on the other hand, there is a Statistical-Based IDS in addition to the current Signature-Based IDS, the results of the attack would differ greatly. The SBIDS would see the packets and may recognize that the properties were inconsistent with the traffic that usually traverses the network. Following this detection, the Statistical-Based system would compute a high score for the packets in the attackers packet stream (like the newspaper breaking through the window), which would lead to an alert generation. While notification of an attack on the systems is a highly desirable feature for an IDS, so too is the detection of an enemy trying to enumerate the network through portscanning.

A SBIDS can provide a more accurate notification of portscanning activities. Portscan detection is a byproduct of the methods in which SBIDS gather data, due to the fact that the scan will be anomalous. At least some of the portscan is likely to be highly anomalous traffic relative to the usual traffic distribution. If this packet has unusual features (i.e. is a crafted packet), this will be still more true [1]. With this in mind, even the portscans that are distributed over a lengthy time frame will be recorded because they will be inherently anomalous. SBIDS give us the ability to detect portscanning packets with much greater accuracy than the "X packets in a Y-sized time frame" method that RBIDS must rely on. The problem with the Statistical-Based system is not the detection of the portscan packets; they will be identified, as any other anomalous activity on the network will be. The problems lie in the dissemination and correlation of the data once it is collected. Correlation is beyond the scope of this document but Silicon Defense is currently developing a correlation engine called Spice. Refer to the Silicon Defense web site for more information.

While there are many advantages to the Statistical-Based approach, there are also some shortcomings with this technology. To begin, a Statistical system must "learn" what is "normal" traffic for a particular network (SBIDS need a good baseline of network traffic). Unlike a Signature-Based system, which has the benefit of being implemented and immediately utilized, the Statistical-Based systems must initially adapt to the network at hand. The longer a SBIDS is placed on a specific network, the more accurate the results will be (assuming the network traffic doesn't significantly alter in form). The second issue with the Statistical-Based approach is related to the adaptive nature of the systems. SBIDS detect anomalies based on discrepancies in "normal" network traffic. If the "normal" network traffic is malicious, the SBIDS will be rendered useless. For example, if the SBIDS sees a numerous number of SYN scans on a network over a period of time the system will eventually assume that this is normal behavior and cease to alert on the activity. This example, while drastic, is a possible scenario. Finally, the alerts that a SBIDS will generate will be relatively difficult to assess compared to a Signature-Based system. The alerts will simply be packet information with no immediately obvious reason for the alert. This analysis will require the services of a trained security professional with the ability to identify abnormalities in traffic at the packet level. Although Statistical-Based systems have some deficiencies, the positive effects of this technology far outweigh the growing pains that will be experienced upon implementation.

The benefits of the statistical-based approach are threefold. Not only do we now have notification for previously unknown attacks, we also have a system that doesn't need constant signature updates, and we have a method to detect port scans that span extensive timeframes as well.

The Statistical Packet Anomaly Detection Engine: Spade

Spade is an anomaly detector publicly released under GNU GPL [20]. It can be downloaded from http://www.silicondefense.com/software/spice/. Spade is a Snort [14] preprocessor plug-in. Spade uses joint probability measurements to decide which packets are anomalous. Spade uses Snort's input/output facilities to grab packets and put them into tables, which are used to determine an anomaly score [1]. The anomaly score is assigned by evaluating the source IP, source port, destination IP, and destination port, among others. Based on the user specified threshold level, Spade will either flag the packet or allow it to pass through the network without notification. The threshold setting is critical in Spade because if it is set too high, the user will miss critical packets; if it is too low, the analyst will see many false-positives. Spade also has an option that will perform automatic threshold adjustment to let Spade decide what the critical threshold number should be. Spade can also generate other reports of importance such as a survey about the distribution of anomaly scores and various reports about the feature statistics such as entropy and conditional probabilities. For more specifics on how Spade calculates anomaly scores, threshold numbers, and probabilities, refer to the documentation present on the Silicon Defense web site [17].

The most critical output for the security analyst will be the Spade alerts, which look very similar to the Snort alerts. The list below is comprised of four Spade-generated alerts. Review the Snort documentation [23] for specifics on how to read these alerts.

```
[**] [104:1:1] spp_anomsensor: Anomaly threshold exceeded: 3.8919 [**]
08/22-22:37:00.419813 24.234.114.96:3246 -> VICTIM.HOST:80
TCP TTL:116 TOS:0x0 ID:25395 IpLen:20 DgmLen:48 DF
******S* Seq: 0xEBCF8EB7  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [104:1:1] spp_anomsensor: Anomaly threshold exceeded: 10.5464 [**]
08/22-22:22:46.577210 24.41.81.216:2065 -> VICTIM.HOST:27374
TCP TTL:108 TOS:0x0 ID:10314 IpLen:20 DgmLen:48 DF
******S* Seq: 0x63B97FE2  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [104:1:1] spp_anomsensor: Anomaly threshold exceeded: 7.8051 [**]
08/23-23:04:53.051245 VICTIM.HOST:31337 -> 64.230.133.196:3486
TCP TTL:255 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0x0  Ack: 0x22676B9  Win: 0x0  TcpLen: 20

[**] [104:1:1] spp_anomsensor: Anomaly threshold exceeded: 9.0907 [**]
09/02-01:30:31.545406 VICTIM.HOST:515 -> 24.42.220.45:1189
TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:60 DF
```

```
***A**S* Seq: 0x16FC5A7F  Ack: 0x529F8CE7  Win: 0x16A0  TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 124399151 14755839 NOP
TCP Options => WS: 0
```

Note the difference in these alerts from ordinary Snort alerts. Spade flags packets based on the degree of anomalousness the packet signifies, not a specific signature. So, unlike a normal Snort alert, we do not see an alert name associated with these traces. Instead, we see an anomaly score preceded by an “Anomaly threshold exceeded” message. We can assess how anomalous these packets are by noting the score in association with the packet; the higher the number, the more anomalous the packet. Also, note that these packets are flagged only if the packet’s anomaly score is higher than the set threshold level. The first alert is an attempt to connect to a local web server. There is not a web server at the VICTIM.HOST address, so this is unusual activity. Yet, Spade did not flag this packet with a high anomaly score. In this specific case, the low anomaly score is likely due to the Code Red [20] epidemic[1]. The anomaly score of this packet is very low because the system had become accustomed to seeing traffic to port 80. Spade clearly thought this packet was not exceedingly anomalous activity (instead, Spade likened the port 80 request to the scenario where the newspaper landed on the driveway, which was anomalous, but not particularly unusual). This packet is an example of a weakness in the Statistical-Based approach. If a large amount of illicit traffic is introduced to a network monitored by a SBIDS, the system will begin to assume this activity is normal and cease to report occurrences of the packet.

The second packet shows a highly anomalous trace. With a score of 10.5464, this packet is extremely unique to the network. When looking at the destination port, it becomes clear why this packet should not be transmitted to the network. Simply, there are no services on the network utilizing the 27374 port. In fact, upon further investigation, it is realized that this port is usually associated with the Sub Seven Trojan [22]. Therefore, the packet warrants investigation, and Spade correctly associated a high anomaly score to the trace.

The third and fourth headers are two more examples of alerts that may be generated by Spade. The difference between Spade and Snort alerts lies primarily in the fact that Spade packets will not immediately identify the reason for capture. An analyst will initially have to analyze the Spade packets more closely than the Snort traces. They will have to inspect the trace and come to a conclusion as to why the particular packet was selected to become a candidate for investigation.

```
[**] [104:2:1] spp_anomsensor: Threshold adjusted to 9.9015 after 2 alerts (of 13) [**]
08/23-00:27:05.550128

[**] [104:2:1] spp_anomsensor: Threshold adjusted to 9.7523 after 0 alerts (of 12) [**]
```

[1] Code Red is a program that exploits a vulnerability in the Microsoft IIS web server. Once a system is compromised with this program it propagates by scanning for other vulnerable hosts on the Internet. When this program was infecting hosts at its peak (July-August, 2001), it flooded the Internet with probes to port 80.

```
08/23-02:19:52.870831

[**] [104:2:1] spp_anomsensor: Threshold adjusted to 8.5722 after 0 alerts (of 12) [**]
08/23-04:11:38.038936

[**] [104:2:1] spp_anomsensor: Threshold adjusted to 8.4727 after 0 alerts (of 11) [**]
08/23-05:08:20.683627
```

Above is a sample of the alert logs that show Spade adjusting the threshold automatically. Spade is decreasing the threshold due to a lack of activity. If not enabled before running Spade, this option would have a fixed number for the threshold and the log would not show these entries.

The survey log listed below displays the distribution of anomaly scores over time. The file shows the hour relative to the execution of the Spade program, the total number of packets of the specified hour, the average anomaly score (Median Anom), the 90th percentile, and the the 99th percentile anomaly scores. This log will only be created if specified in the Spade configuration.

```
60.00 minute interval
# Packet Count    Median Anom     90th Percentile Anom  99th Percentile
1       20      3.629443        9.708243        10.331995
2       16      5.620299        8.082586        8.135222
3       14      7.415492        10.130501       10.333078
4       25      7.001369        10.333560       10.333619
5       22      6.758892        9.193461        10.297281
6       16      3.575038        8.832395        8.947573
7       10      3.562193        8.530327        8.530327
8       8       5.730879        8.109143        8.109143
9       5       3.547780        3.548970        3.548970
10      8       3.542491        7.570529        7.570529
```

The log.txt file is of importance in that it displays, at minimum, the number of packets that Spade accepted (analyzed) and the number of alerts generated.

Below is an example of the log.txt file output; the results are typical of what would be seen if Spade executed in probability mode 3 (edited for brevity).

```
392 packets recorded
51 packets reported as alerts
Threshold learning results: top 200 anomaly scores over 23.58361 hours
Suggested threshold based on observation: 3.522590
Top scores:
3.52317,3.52433,3.52549,3.52665,3.52782,3.52898,3.53015,3.53132,3.53249,3.53366,3
.53483,3.53601,3.53718,3.53836,3.53954,3.54072,3….10.29728,
10.33199,10.33308,10.33351,10.33360,10.33362
First runner up is 3.52201, so use threshold between 3.52201 and 3.52317 for 8.523
packets/hr
H(dip)=5.30397479
```

```
H(dport|dip)=9.69742991
P(dip=44044824)= 0.064466877730
P(dip=44044824,dport=1)= 0.000062047043
P(dip=44044824,dport=2)= 0.000077558804
P(dip=44044824,dport=3)= 0.000062047043
P(dip=44044824,dport=4)= 0.000062047043
P(dip=44044824,dport=5)= 0.000062047043
```

Initially, the log displays basic packet statistics and the threshold learning results. This log shows how and why Spade is determining a certain threshold for a particular time. Towards the bottom of this file probability statistics are listed where H = entropy, dip = destination IP, dport = destination port, and P = probability.

In addition to the previously mentioned facilities, Spade also produces binary log output by using the Snort output method. This feature enables the user to later go back and do a more thorough analysis of the actual packet with other tools such as tcpdump [5], ethereal [6], or any other packet analyzer that will read tcpdump log file format. Spade has a lot of functionality, and because it is built on Snort, they can be utilized in conjunction with each other as a dual IDS solution. Snort benefits the network by alerting on packets with known signatures, where Spade will learn what is normal traffic for the network and alert to any discrepancies from that norm.

Deployment

The deployment of Spade is relatively easy but there are a few prerequisites.

1. A Unix operating system
2. Packet capture software (Snort)
3. A computer connected to an active network

The authors of Spade have made it very easy to deploy this SBIDS in addition to a current IDS. Snort is required on the system because Spade is built to utilize Snort's input/output facilities[2]. All versions of Snort above 1.7 have support for Spade installed by default. The documentation is located in /contrib/Spade-<version>.tar.gz (where <version> is the version of Spade) within the Snort directory of the unzipped snort source tarball. For example, to start by reading the Spade README document, proceed with the following steps:

Change into the Snort contrib directory:

```
> cd $SNORT/snort/contrib (where $SNORT is the snort root directory)
```

Untar and gunzip the Spade source:

```
> tar –xvzf Spade-010818.1.tar.gz
```

[2] The fact that Spade requires Snort to operate does not imply that Snort must be used as the complementary IDS; any IDS can be used in conjunction with Spade.

Change into the Spade directory:

```
> cd Spade-010818.1
```

Open the README file:

```
> less README
```

To upgrade to a newer version of Spade, follow the steps above, but view the Installation file in addition to the README. The upgrade process is detailed in the Installation file; upgrading is a simple two-step procedure.

Once Spade is installed correctly, make a decision as to whether Spade will be run in addition to Snort or as a separate process. The Spade authors advise users to initially try Spade as a separate process, especially if it is on a production system. The differences in configuration are minimal regardless of which method is chosen. Continue by configuring the spade.config file.

Open the spade.config file for editing:

```
> vi spade.config
```

The spade.config file is short and direct. The layout of this file is identical to that of the Snort configuration file. Snort actually processes the spade.config file and then hands it to Spade upon completion. The default comment for each variable is descriptive and valuable. If there are any questions regarding the specifics of each option, refer to the Usage file located in the same directory. The primary configuration options in the spade.config file are the threshold and the output methods.

Change the reporting threshold because it is off by default:

```
Preprocessor spade: 4 $SPADEDIR/spade.rcv $SPADEDIR/log.txt 3 50000
```

All packets with an anomaly score of at least as great as 4.0 will be reported as an alert. The "3" is the probability mode; this number bases probability on destination IP and destination port. Refer to the Usage file for more specifics on the modes available. The next configuration line to modify is the adaptive threshold feature. Comment them all out and use the static number mentioned earlier (4). When testing is complete it is highly recommended to modify the configuration and utilize the adaptive threshold methods available. The adaptive threshold allows Spade to decide what the optimal threshold level should be. Please review the Usage document to choose which adaptive method would be best suited for a particular environment.

```
#preprocessor spade-adapt3: 0.01 60 168
```

Enable the reporting options that Spade offers:

```
preprocessor spade-survey: $SPADEDIR/survey.txt 60
preprocessor spade-stats: entropy uncondprob condprob
```

The spade-survey option enables the generation of a report that shows anomaly scores produced in the last time interval (an example was listed previously in the Spade section of this document). The spade-stats configuration reports periodically on certain information about the network traffic but will not write to the log.txt file until Spade receives a SIGHUP, SIGQUIT, SIGUSR1 or Snort is exited. Refer to the Usage manual for the specific descriptions of each argument.

The configuration file in its entirety (comments edited out for brevity).

```
var SPADEDIR /var/log/snort
preprocessor spade: 4 $SPADEDIR/spade.rcv $SPADEDIR/log.txt 3 50000
preprocessor spade-homenet: 0.0.0.0/0
#preprocessor spade-adapt3: 0.01 60 168
#preprocessor spade-adapt: 20 2 0.5
#preprocessor spade-adapt2: 0.01 15 4 24 7
preprocessor spade-threshlearn: 200 24
preprocessor spade-survey:  $SPADEDIR/survey.txt 60
preprocessor spade-stats: entropy uncondprob condprob
```

Execute Spade by running Snort with the following option:

```
> /usr/local/bin/snort –c spade.config
```

Spade should now be monitoring packets on the network. The above command will run Spade as it's own process, so as not to interfere with other instances of Snort that may be running. If Snort IDS and Spade are required to be run at the same time with the same process, the snort.conf file must be modified. The snort.conf section that deals with Spade (commented out by default) will need to be edited to mirror the configuration options in the spade.config file.

To assure everything is working properly, check the specified logging directory (/var/log/snort in the example) to see if the files spade.rcv, survey.txt, and log.txt are present. There will be a spade.rcv file as soon as the process captures the prespecified number of packets – this is called the "checkpointing" process of Spade. In the above example this number would be 50000. The spade.rcv file is what maintains state for the program. So the spade.rcv file should be produced sometime after the initial execution of Spade.

For further information regarding installation and configuration, refer to the documentation in the Spade directory or the Silicon Defense web site.

Conclusion

Statistical-Based Intrusion Detection Systems are an extremely effective method to supplement a current Intrusion Detection System. The benefits of a SBIDS, like Spade, should not be overlooked. Utilizing Spade is a second layer of defense. Spade is one of the first tools of its kind that shows the security community the possibilities of Statistical-Based Intrusion Detection. Never before has there been the ability to accurately identify rogue packets by comparing them with what is "normal" for a specific network. Never before has there been a method to easily recognize portscans spanning lengthy time frames. With automated threshold discovery and constant assessment of network activity to identify anomalous traffic, Spade is also a relatively low-labor IDS. The SBID technology is still in its infancy though, so there is still a lot of progress to be made in terms of functionality and false-positive control. Nevertheless, by utilizing both a Signature-Based and Statistical-Based Intrusion Detection System, the vast majority of anomalous traffic on network will be identified. There is no one silver bullet in the IDS field, but layering the systems and experimenting with new methods of intrusion detection can greatly improve the chances of winning the uphill battle against electronic intruders.

1.1.0 References

[1] S. Staniford, J. Hoagland, J. McAlerney. “Practical Automated Detection of Stealthy Portscans.” In: *CCS IDS Workshop Athens*. November 1, 2000.

[3] A. Sundaram. “An Introduction to Intrusion Detection.” http://www.acm.org/crossroads/xrds2-4/intrus.html

[4] H. Debar. “What is knowledge-based intrusion detection?” In: Intrusion Detection FAQ. http://www.sans.org/newlook/resources/IDFAQ/knowledge_based.htm

[5] H. Debar. “What is behavior-based intrusion detection?” In: Intrusion Detection FAQ. http://www.sans.org/newlook/resources/IDFAQ/behavior_based.htm

[6] D. Lehmann. “What is ID?” In: Intrusion Detection FAQ. http://www.sans.org/newlook/resources/IDFAQ/what_is_ID.htm

[7] J. Kim. “An Artificial Immune System for Network Intrusion Detection.” http://www.cs.ucl.ac.uk/staff/J.Kim/GECCO_WS99.html

[8] M. Craymer, J. Cannady, J. Harrell. “New Methods of Intrusion Detection using Control-Loop Measurement.” In: Fourth Technology for Information Security Conference’96. May, 16, 1996.

[9] W. Lee, S. Stolfo. “Data Mining Approaches for Intrusion Detection.” In: Proceedings of the 7th USENIX Security Symposium. 1998.

[10] M. Gerken. “Statistical-Based Intrusion Detection.” http://www.sei.cmu.edu/str/descriptions/sbid_body.html

[11] http://www.nfr.com/products/NID/

TAKE OUT [12] http://www.checkpoint.com/products/firewall-1/realsecure.html

[13] http://www.portcullis-security.com/products/index.htm

[14] http://www.snort.org

[15] http://www.cisco.com/warp/public/cc/pd/sqsw/sqidsz/

[16] S. Northcutt. Network Intrusion Detection: An Analyst's Handbook. New Riders, Indianapolis, 1999. p. 125.

[17] http://www.silicondefense.com/software/spice/index.htm

[18] http://www.tcpdump.org

[19] http://www.ethereal.com

[20] http://www.gnu.org/copyleft/gpl.html

[21] R. Permeh, M. Maiffret. ".ida "Code Red" Worm."
http://www.eeye.com/html/Research/Advisories/AL20010717.html.

[22] R. Lyttle. http://www.sub-seven.com

[23] D. Ruiu. "Snort FAQ Version 1.8."
http://snort.sourcefire.com/docs/faq.html

[24] M. Prabhaker. "Intrusion Detection."
http://www.cs.wright.edu/~pmateti/Courses/499/IntrusionDetection/

[25] M. Gerken. "Rule-Based Intrusion Detection."
http://www.sei.cmu.edu/str/descriptions/rbid_body.html

[26] R. Lupton. Statistics In Theory And Practice. Princeton University Press, Princeton, NJ, 1993. p. 50.

2.0.0 Analysis of 5 Detects

In this section we will analyze five network detects. Some or all of the actual logs will be displayed initially as evidence of the attack. While there are two types of log formats show in this section, both are identical. Below is a description of each field.

Example packet:

```
11/20-07:00:17.352241 198.119.49.82:49570 -> 17.254.3.223:80
TCP TTL:252 TOS:0x0 ID:62821 IpLen:20 DgmLen:547 DF
***AP*** Seq: 0x42409D22  Ack: 0xF6081141  Win: 0x8000  TcpLen: 20
```

ROW 1: 11/20-07:00:17.352241 (timestamp), 198.119.49.82 (source ip address), 49570 (source port), 17.254.3.223 (destination ip address), 80 (destination port)

ROW 2: TCP (protocol), TTL:252 (time to live), TOS:0x0 (type of service), ID:62821 (IP ID), IpLen:20 (IP length), DgmLen:547 (datagram length), DF (don't fragment bit set)

ROW 3: ***AP*** (tcp options – the ack and push are set in this example), Seq: 0x42409D22 (sequence number), Ack: 0xF6081141 (acknowledge number), Win: 0x8000 (window size), TcpLen:20 (length of the tcp portion of the packet).

2.1.0 Detect 1, OS Fingerprinting

```
09/12-12:22:14.653043 211.167.27.161:3022 -> EMP.NET.30.11:0
TCP TTL:108 TOS:0x0 ID:16803 IpLen:20 DgmLen:48 DF
**UA*RSF Seq: 0x8006337D  Ack: 0xD3A71BA1  Win: 0xD623  TcpLen: 48  UrgPtr: 0x50
TCP Options (1) => Opt 255 (40): FFC8 0000 0000 0B71 0050 4FF9 6625 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x0000: 50 AA 00 04 00 FD 3F AA 00 04 00 19 18 AA AA 03        P.....?.........
0x0010: 00 00 00 08 00 45 00 00 30 41 A3 40 00 6C 06 09        .....E..0A.@.l..
0x0020: 76 D3 A7 1B A1 C0 56 14 10 0B CE 00 00 80 06 33        v.....V........3
0x0030: 7D D3 A7 1B A1 C0 37 D6 23 11 82 00 50 FF 8C FF        }.....7.#...P...
0x0040: C8 00 00 00 00                                         .....

09/12-12:22:23.132881 211.167.27.161:3022 -> EMP.NET.225.88:0
TCP TTL:108 TOS:0x0 ID:16803 IpLen:20 DgmLen:48 DF
**UA*RSF Seq: 0x8006337D  Ack: 0xD3A71BA1  Win: 0xD623  TcpLen: 48  UrgPtr: 0x50
TCP Options (1) => Opt 255 (40): FFC8 0000 0000 0B71 0050 4FF9 6625 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x0000: 50 AA 00 04 00 FD 3F AA 00 04 00 19 18 AA AA 03        P.....?.........
0x0010: 00 00 00 08 00 45 00 00 30 41 A3 40 00 6C 06 09        .....E..0A.@.l..
0x0020: 76 D3 A7 1B A1 C0 56 14 10 0B CE 00 00 80 06 33        v.....V........3
0x0030: 7D D3 A7 1B A1 C0 37 D6 23 11 82 00 50 FF 8C FF        }.....7.#...P...
0x0040: C8 00 00 00 00
```

2.1.1 Source of Trace

This trace was detected on one of the networks that I monitor for my employer. The monitor box is placed on a spanning port and receives all traffic that is in or outbound from the entire center. This system succumbs to an approximate 10-15% packet loss at any given moment. We gather approximately 250,000 Snort alerts every 7 hours on this system – even though we utilize an extremely modified rule set. While there are numerous detects to analyze, it is sometimes difficult to assess the results due to the fact that statistically we are missing approximately 10-15% of each attack. I feel that these factors should be noted before we proceed with the analysis of these traces. As an analyst, these factors make it much more difficult and time consuming to identify an attack with relative assurance.

2.1.2 Detect was generated by

This detect was generated by Snort 1.8.1-RELEASE running on a dual Pentium 800mhz with 1gig of ram. The system is a kernel-modified FreeBSD 4.3 box that has been stripped down to only provide the ssh2 service. There is no compiler or other unneeded tools installed on the system. There are two interfaces, one NIC has an non-routeable address associated with it, and is therefore virtually invisible. This NIC is connected to the spanning port on the switch and therefore receives all of the traffic we analyze in the following traces. The second NIC has been assigned a routable address and is used to remotely communicate with the monitor box through ssh2. The binary logs generated by Snort are compressed, MD5'd, and sent to a log host. All of the logs are processed and analyzed in the local office.

2.1.3 Probability the source address was spoofed

This particular scan is presumably a reconnaissance information acquisition technique. If this assumption is correct, and the victim host is not susceptible to TCP sequence number prediction, this trace is most likely NOT spoofed. The attacker is trying to gather information from the victim host and therefore will insist that she receives the packets sent by the victim in response to her queries. Note that there is no need for the attacker to complete a three way handshake with the victim. Therefore, this scan has a remote possibility of being spoofed, but to receive the response from this probe, the attacker would have to be on the same network segment as the spoofed address.

2.1.4 Description of the attack

This is not an attack. Rather, it is a method used to gather information from the victim as to what operating system it is utilizing (OS fingerprinting). By identifying the operating system of a host, an attacker can target specific services and vulnerabilities thereof that are unique to the particular operating system distribution. Here is a great article that describes OS fingerprinting http://www.insecure.org/nmap/nmap-fingerprinting-article.txt

2.1.5 Attack mechanism

This attack attempts to successfully gather a "fingerprint" of the system which will identify the operating system. When approached with packets, most operating systems will react in an individual manner due to different implementations of the TCP/IP stack. These unique characteristics can be used to identify the hosts' operating system. By sending a packet with a multitude of flags set and that is directed at a reserved and unused port, the attack attempts to illicit a response from the victim host. The response to this odd looking packet is what the attack will ultimately analyze to recognize what type of operating system the host is using. This packet is so illegal in regards to the TCP RFC (RFC 791) that many operating systems react differently to it. The operating system can be determined by analyzing certain characteristics of the response packet from the victim. This attack was captured simply because it was directed at a reserved port (0), but upon further investigation, it is clear that this packet is malicious. The urgent pointer is set, which means that the attacker is specifying a portion of the datagram as "urgent data". In a normal TCP operation, this data is interpreted by the application, since there is no service (and subsequently, application) that operates on port 0, this is obviously bogus. The numerous flags contradict each other and the seq and ack are identical in sequential packets. These packets also have a TCP option set, which is option 255. I could not identify what this particular option is used for (or if it is even valid). Overall this packet is clearly not something that should be traversing the network.

2.1.6 Correlations

Here are some articles which describe this attack and the implications. There are also other forms of this attack which utilize other protocols such as ICMP, I have listed a good article (3) on it as well because both probes are closely tied in principle.

1. CVE: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454
2. http://www.insecure.org/nmap/nmap-fingerprinting-article.html
3. http://www.phrack.org/show.php?p=57&a=7

2.1.7 Evidence of active targeting

This looks like a specific scan of these two hosts. After running searches through a months worth of IDS logs, these were the only three instances of this IP address. This leads me to believe that the attacker had a specific desire to gather knowledge about these two hosts.

2.1.8 Severity

Direction	Category	Value	Reasons
Attack	Criticality	5	This host is a dns server. While there are multiple dns servers at our disposal, this is a critical machine because it could cause

			serious problems if it were compromised.
	Lethality	2	If successful, attacker gains key knowledge that could possibly be used to ultimately compromise the machine.
Response	System	2	The system replied to the attack giving the attacker a good idea of the operating system in use.
	Network	1	The network did not stop the system's reply. This was a successful operating system probe.
Severity = (5 + 2) – (2 + 1) = 4			

2.1.9 Defense recommendations

This probe can be defeated with a few simple rules at the border router or firewall. Port 0 is a reserved address and should never be accessed from an external host. This port should be blocked outright from either the border router or the network firewall. For added assurance, there should be an IDS set up directly outside the network to monitor all in and outbound traffic.

2.1.10 Multiple choice question

What characteristics do these packets have that indicate that they are suspicious traffic? (choose the *best* answer)

a. Same Sequence and Acknowledge numbers for both packets
b. Same source and destination IP addresses for both packets
c. TCP options
d. Urgent pointer set
e. Destination Port and Flags
f. All of the above except b

Answer: f. We should not see to sequential packets with identical seq and ack numbers. The TCP options are not necessary for this packet, especially when coupled with the multiple conflicting flags being set as well as the destination port of 0. The urgent pointer being set to a location which doesn't even exist in this size datagram is also clearly suspicious activity. All of these things in the header should make any security analyst suspicious!

2.2.0 Detect 2, NULL Session

```
09/12-03:00:09.904704 198.118.96.118:3929 -> EMP.NET.196.3:139
TCP TTL:125 TOS:0x0 ID:51373 IpLen:20 DgmLen:217
***AP*** Seq: 0x188BBCA9  Ack: 0x5AD44F04  Win: 0x21C9  TcpLen: 20
0x0000: 50 AA 00 04 00 FC 3F AA 00 04 00 19 18 AA AA 03          P.....?.........
0x0010: 00 00 00 08 00 45 00 00 D9 C8 AD 40 00 7D 06 C8          .....E.....@.}..
```

```
0x0020: C9 C6 76 60 76 80 B7 C4 03 0F 59 00 8B 18 8B BC          ..v`v.....Y.....
0x0030: A9 5A D4 4F 04 50 18 21 C9 31 40 00 00 00 00 00          Z.O.P.!.1@.....
0x0040: AD FF 53 4D 42 73 00 00 00 00 18 03 80 00 00 F1          ..SMBs..........
0x0050: 7A 82 66 9E 40 4D 00 00 00 00 00 FE CA 00 00 00          z.f.@M..........
0x0060: 00 0D 75 00 84 00 04 41 32 00 00 00 00 00 00 00          ..u....A2.......
0x0070: 01 00 00 00 00 00 00 00 D4 00 00 00 47 00 00 00          ............G...
0x0080: 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77 00 73          ...W.i.n.d.o.w.s
0x0090: 00 20 00 4E 00 54 00 20 00 31 00 33 00 38 00 31          . .N.T. .1.3.8.1
0x00A0: 00 00 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77          .....W.i.n.d.o.w
0x00B0: 00 73 00 20 00 4E 00 54 00 20 00 34 00 2E 00 30          .s. .N.T. .4...0
0x00C0: 00 00 00 00 00 04 FF 00 00 00 00 00 01 00 1E 00          ................
0x00D0: 00 5C 00 5C 00 52 00 45 00 4D 00 4F 00 54 00 45          .\.\.R.E.M.O.T.E
0x00E0: 00 37 00 5C 00 45 00 54 00 00 00 41 3A 00                .7.\.E.T...A:.

09/12-03:00:30.073312 198.118.96.118:3935 -> EMP.NET.196.3:139
TCP TTL:125 TOS:0x0 ID:63661 IpLen:20 DgmLen:217
***AP*** Seq: 0x188C0BF2  Ack: 0x5B265310  Win: 0x21C9  TcpLen: 20
0x0000: 50 AA 00 04 00 FC 3F AA 00 04 00 19 18 AA AA 03          P.....?.........
0x0010: 00 00 00 08 00 45 00 00 D9 F8 AD 40 00 7D 06 98          .....E.....@.}..
0x0020: C9 C6 76 60 76 80 B7 C4 03 0F 5F 00 8B 18 8C 0B          ..v`v....._.....
0x0030: F2 5B 26 53 10 50 18 21 C9 C3 91 00 00 00 00 00          .[&S.P.!........
0x0040: AD FF 53 4D 42 73 00 00 00 00 18 03 80 00 00 A5          ..SMBs..........
0x0050: 5E 48 BA 53 A4 37 66 00 00 00 00 FE CA 00 00 00          ^H.S.7f.........
0x0060: 00 0D 75 00 84 00 04 41 32 00 00 00 00 00 00 00          ..u....A2.......
0x0070: 01 00 00 00 00 00 00 00 D4 00 00 00 47 00 00 00          ............G...
0x0080: 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77 00 73          ...W.i.n.d.o.w.s
0x0090: 00 20 00 4E 00 54 00 20 00 31 00 33 00 38 00 31          . .N.T. .1.3.8.1
0x00A0: 00 00 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77          .....W.i.n.d.o.w
0x00B0: 00 73 00 20 00 4E 00 54 00 20 00 34 00 2E 00 30          .s. .N.T. .4...0
0x00C0: 00 00 00 00 00 04 FF 00 00 00 00 00 01 00 1E 00          ................
0x00D0: 00 5C 00 5C 00 52 00 45 00 4D 00 4F 00 54 00 45          .\.\.R.E.M.O.T.E
0x00E0: 00 37 00 5C 00 45 00 54 00 00 00 41 3A 00                .7.\.E.T...A:.

09/12-03:00:40.137245 198.118.96.118:3938 -> EMP.NET.196.3:139
TCP TTL:125 TOS:0x0 ID:4526 IpLen:20 DgmLen:217
***AP*** Seq: 0x188C32B8  Ack: 0x5B4F838F  Win: 0x21C9  TcpLen: 20
0x0000: 50 AA 00 04 00 FC 3F AA 00 04 00 19 18 AA AA 03          P.....?.........
0x0010: 00 00 00 08 00 45 00 00 D9 11 AE 40 00 7D 06 7F          .....E.....@.}..
0x0020: C9 C6 76 60 76 80 B7 C4 03 0F 62 00 8B 18 8C 32          ..v`v.....b....2
0x0030: B8 5B 4F 83 8F 50 18 21 C9 E6 B0 00 00 00 00 00          .[O..P.!........
0x0040: AD FF 53 4D 42 73 00 00 00 00 18 03 80 00 00 DA          ..SMBs..........
0x0050: 74 09 00 76 1F A4 FE 00 00 00 00 FE CA 00 00 00          t..v............
0x0060: 00 0D 75 00 84 00 04 41 32 00 00 00 00 00 00 00          ..u....A2.......
0x0070: 01 00 00 00 00 00 00 00 D4 00 00 00 47 00 00 00          ............G...
0x0080: 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77 00 73          ...W.i.n.d.o.w.s
0x0090: 00 20 00 4E 00 54 00 20 00 31 00 33 00 38 00 31          . .N.T. .1.3.8.1
0x00A0: 00 00 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77          .....W.i.n.d.o.w
0x00B0: 00 73 00 20 00 4E 00 54 00 20 00 34 00 2E 00 30          .s. .N.T. .4...0
0x00C0: 00 00 00 00 00 04 FF 00 00 00 00 00 01 00 1E 00          ................
0x00D0: 00 5C 00 5C 00 52 00 45 00 4D 00 4F 00 54 00 45          .\.\.R.E.M.O.T.E
0x00E0: 00 37 00 5C 00 45 00 54 00 00 00 41 3A 00                .7.\.E.T...A:.

09/12-03:01:00.293434 198.118.96.118:3944 -> EMP.NET.196.3:139
TCP TTL:125 TOS:0x0 ID:17070 IpLen:20 DgmLen:217
***AP*** Seq: 0x188C81CF  Ack: 0x5BA1D30F  Win: 0x21C9  TcpLen: 20
0x0000: 50 AA 00 04 00 FC 3F AA 00 04 00 19 18 AA AA 03          P.....?.........
```

```
0x0010: 00 00 00 08 00 45 00 00 D9 42 AE 40 00 7D 06 4E            .....E...B.@.}.N
0x0020: C9 C6 76 60 76 80 B7 C4 03 0F 68 00 8B 18 8C 81             ..v`v.....h.....
0x0030: CF 5B A1 D3 0F 50 18 21 C9 DC 03 00 00 00 00 00             .[...P.!........
0x0040: AD FF 53 4D 42 73 00 00 00 00 18 03 80 00 00 9E             ..SMBs..........
0x0050: DB 56 32 C8 EA AC 57 00 00 00 00 FE CA 00 00 00             .V2...W.........
0x0060: 00 0D 75 00 84 00 04 41 32 00 00 00 00 00 00 00             ..u....A2.......
0x0070: 01 00 00 00 00 00 00 00 D4 00 00 00 47 00 00 00             ............G...
0x0080: 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77 00 73             ...W.i.n.d.o.w.s
0x0090: 00 20 00 4E 00 54 00 20 00 31 00 33 00 38 00 31             . .N.T. .1.3.8.1
0x00A0: 00 00 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77             .....W.i.n.d.o.w
0x00B0: 00 73 00 20 00 4E 00 54 00 20 00 34 00 2E 00 30             .s. .N.T. .4...0
0x00C0: 00 00 00 00 00 04 FF 00 00 00 00 00 01 00 1E 00             ................
0x00D0: 00 5C 00 5C 00 52 00 45 00 4D 00 4F 00 54 00 45             .\.\.R.E.M.O.T.E
0x00E0: 00 37 00 5C 00 45 00 54 00 00 00 41 3A 00                   .7.\.E.T...A:.

09/12-03:01:10.355902 198.118.96.118:3947 -> EMP.NET.196.3:139
TCP TTL:125 TOS:0x0 ID:23214 IpLen:20 DgmLen:217
***AP*** Seq: 0x188CA952  Ack: 0x5BCB1A32  Win: 0x21C9  TcpLen: 20
0x0000: 50 AA 00 04 00 FC 3F AA 00 04 00 19 18 AA AA 03             P.....?.........
0x0010: 00 00 00 08 00 45 00 00 D9 5A AE 40 00 7D 06 36             .....E...Z.@.}.6
0x0020: C9 C6 76 60 76 80 B7 C4 03 0F 6B 00 8B 18 8C A9             ..v`v.....k.....
0x0030: 52 5B CB 1A 32 50 18 21 C9 C7 28 00 00 00 00 00             R[..2P.!..(.....
0x0040: AD FF 53 4D 42 73 00 00 00 00 18 03 80 00 00 90             ..SMBs..........
0x0050: 1C B6 1B D4 21 F5 FE 00 00 00 00 FE CA 00 00 00             ....!...........
0x0060: 00 0D 75 00 84 00 04 41 32 00 00 00 00 00 00 00             ..u....A2.......
0x0070: 01 00 00 00 00 00 00 00 D4 00 00 00 47 00 00 00             ............G...
0x0080: 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77 00 73             ...W.i.n.d.o.w.s
0x0090: 00 20 00 4E 00 54 00 20 00 31 00 33 00 38 00 31             . .N.T. .1.3.8.1
0x00A0: 00 00 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77             .....W.i.n.d.o.w
0x00B0: 00 73 00 20 00 4E 00 54 00 20 00 34 00 2E 00 30             .s. .N.T. .4...0
0x00C0: 00 00 00 00 00 04 FF 00 00 00 00 00 01 00 1E 00             ................
0x00D0: 00 5C 00 5C 00 52 00 45 00 4D 00 4F 00 54 00 45             .\.\.R.E.M.O.T.E
0x00E0: 00 37 00 5C 00 45 00 54 00 00 00 41 3A 00                   .7.\.E.T...A:.

09/12-03:01:20.428547 198.118.96.118:3950 -> EMP.NET.196.3:139
TCP TTL:125 TOS:0x0 ID:29358 IpLen:20 DgmLen:217
***AP*** Seq: 0x188CD022  Ack: 0x5BF5227F  Win: 0x21C9  TcpLen: 20
0x0000: 50 AA 00 04 00 FC 3F AA 00 04 00 19 18 AA AA 03             P.....?.........
0x0010: 00 00 00 08 00 45 00 00 D9 72 AE 40 00 7D 06 1E             .....E...r.@.}..
0x0020: C9 C6 76 60 76 80 B7 C4 03 0F 6E 00 8B 18 8C D0             ..v`v.....n.....
0x0030: 22 5B F5 22 7F 50 18 21 C9 1D DC 00 00 00 00 00             "[.".P.!........
0x0040: AD FF 53 4D 42 73 00 00 00 00 18 03 80 00 00 B3             ..SMBs..........
0x0050: B4 32 5A 13 FD 90 4F 00 00 00 00 FE CA 00 00 00             .2Z...O.........
0x0060: 00 0D 75 00 84 00 04 41 32 00 00 00 00 00 00 00             ..u....A2.......
0x0070: 01 00 00 00 00 00 00 00 D4 00 00 00 47 00 00 00             ............G...
0x0080: 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77 00 73             ...W.i.n.d.o.w.s
0x0090: 00 20 00 4E 00 54 00 20 00 31 00 33 00 38 00 31             . .N.T. .1.3.8.1
0x00A0: 00 00 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77             .....W.i.n.d.o.w
0x00B0: 00 73 00 20 00 4E 00 54 00 20 00 34 00 2E 00 30             .s. .N.T. .4...0
0x00C0: 00 00 00 00 00 04 FF 00 00 00 00 00 01 00 1E 00             ................
0x00D0: 00 5C 00 5C 00 52 00 45 00 4D 00 4F 00 54 00 45             .\.\.R.E.M.O.T.E
0x00E0: 00 37 00 5C 00 45 00 54 00 00 00 41 3A 00                   .7.\.E.T...A:.

09/12-03:01:40.590172 198.118.96.118:3956 -> EMP.NET.196.3:139
TCP TTL:125 TOS:0x0 ID:41902 IpLen:20 DgmLen:217
***AP*** Seq: 0x188D1F21  Ack: 0x5C4804A1  Win: 0x21C9  TcpLen: 20
0x0000: 50 AA 00 04 00 FC 3F AA 00 04 00 19 18 AA AA 03      P.....?.........
0x0010: 00 00 00 08 00 45 00 00 D9 A3 AE 40 00 7D 06 ED      .....E.....@.}..
0x0020: C8 C6 76 60 76 80 B7 C4 03 0F 74 00 8B 18 8D 1F      ..v`v.....t.....
0x0030: 21 5C 48 04 A1 50 18 21 C9 8E 18 00 00 00 00 00      !\H..P.!........
0x0040: AD FF 53 4D 42 73 00 00 00 00 18 03 80 00 00 AB      ..SMBs..........
0x0050: 10 3F 10 33 C1 CA C2 00 00 00 00 FE CA 00 00 00      .?.3............
0x0060: 00 0D 75 00 84 00 04 41 32 00 00 00 00 00 00 00      ..u....A2.......
0x0070: 01 00 00 00 00 00 00 00 D4 00 00 00 47 00 00 00      ...........G...
0x0080: 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77 00 73      ...W.i.n.d.o.w.s
0x0090: 00 20 00 4E 00 54 00 20 00 31 00 33 00 38 00 31      . .N.T. .1.3.8.1
0x00A0: 00 00 00 00 00 57 00 69 00 6E 00 64 00 6F 00 77      .....W.i.n.d.o.w
0x00B0: 00 73 00 20 00 4E 00 54 00 20 00 34 00 2E 00 30      .s. .N.T. .4...0
0x00C0: 00 00 00 00 00 04 FF 00 00 00 00 00 01 00 1E 00      ................
0x00D0: 00 5C 00 5C 00 52 00 45 00 4D 00 4F 00 54 00 45      .\.\.R.E.M.O.T.E
0x00E0: 00 37 00 5C 00 45 00 54 00 00 00 41 3A 00            .7.\.E.T...A:.
[snip -- 1453 records omitted for brevity]
```

2.2.1 Source of trace

This trace was detected on the same network as the previous analysis. This trace was found in the same data set, which means that it was captured with the same sensor and within the same 7 hour period.

2.2.2 Detect was generated by

Snort-1.8.1-RELEASE. Please refer to the previous detect for a description of the Snort log format and all relevant fields in the output.

2.2.3 Probability the source address was spoofed

The odds that this trace is spoofed is rare. The perpetrator is trying to enumerate the network and therefore will require a response from the victim host.

2.2.4 Description of attack

This is a NULL session. A NULL session is a method of "anonymously" connecting to a Windows share. It is a way of letting an anonymous user retrieve information such as user names and shares over the network or connect without authentication. This attack utilizes the fact that this connection is an anonymous logon and proceeds to enumerate the network shares. A NULL session can also be used to edit the registry with the same permissions as the group "Everyone". A NULL session can also be used to gather accounts on the server, last logon time for users, RAS callback numbers etc. This attack is described in CVE-2000-0347.

2.2.5 Attack mechanism

This attack uses a standard option available on the Windows NT operating system. To connect, a user would simply type:

```
        c:>net use <computer>ipc$ "" /user:""
```

Where <computer> is the NetBIOS, hostname, or IP of the system to connect to. This particular trace was quite frequent so it is clearly not a manual process. Therefore it is either a malicious program being run against the network or it is legitimate traffic. From the look of the timestamps, this looks like a manual process rather than the utilization of a tool though. This attack simply connects to the host and the amount of information derived is up to the attacker. Upon review of the network policies at the center being monitored and a review of the attacking host, it was found that the attacker was attempting this exploit after access to the victim machine was revoked.

2.2.6 Correlations

This attack can be correlated with information from Bugtraq and the Arachnids database. The Bugtraq information can be located at http://www.securityfocus.com/bid/1163. The Arachnids information can be viewed at http://www.whitehats.com/info/ids204.

A simple description of the attack is located at http://frasier.dpo.uab.edu/security/slides/img3.htm. A thorough description of how to make a NULL session is located at http://service1.symantec.com/SUPPORT/ent-security.nsf/docid/2000092916544448.

2.2.7 Evidence of active targeting

There is evidence of active targeting in this scenario because we see no other traces of this activity within a three week period. In this 7 hour period this single host triggered over 1500 NULL SESSION alerts on the IDS. The host that was attacked in this instance was a Windows host that allowed NULL sessions. The attacker either had prior knowledge of the network through reconnaissance, or she was extremely lucky in guessing that this host was exploitable. It should be noted that this attack may also not be an example of active targeting. While there were numerous attempts to a single host (and to no others on the network), there is a possibility that the attacker was randomly selecting hosts to attempt this attack on.

2.2.8 Severity

Direction	**Category**	**Value**	**Reasons**
Attack	Criticality	4	The host is the Primary Domain Controller
	Lethality	4	If successful, this attack can allow the attacker to enumerate the shares of the network, identify uses, and edit the registry.
Response	System	1	There are no host based countermeasures in effect for this system.
	Network	2	There is little in place to deter this attack; only the IDS discovered the presence of this activity.
Severity = (4 + 4) – (1 + 2) = 5			

2.2.9 Countermeasures

Severity =

This system needs to be patched. Windows NT 4 service pack 3 cleans up this issue if applied. Once this patch is enabled, the administrator simply needs to disallow NULL sessions. There also needs to be a firewall set up at the border router to deny packets destined for ports 135, 137, 138, and 139. These countermeasures will effectively alleviate this issue.

2.2.10 Multiple choice question

What identifiers are available in this trace that leads us to believe that this is a NULL session exploit? (choose the *best* answer)

A) IP ID incrimination
B) Source port number
C) Payload of the packet
D) Both Destination port number and payload
E) Destination port number

2.3.0 Detect 3, MTU discovery

```
09/12-03:02:50.309677 EMP.NET.10.134 -> 165.21.86.84
ICMP TTL:253 TOS:0x0 ID:57207 IpLen:20 DgmLen:1420
Type:0  Code:0  ID:61952  Seq:26761  ECHO REPLY
0x0000: 50 AA 00 04 00 19 18 AA 00 04 00 FD 3F AA AA 03        P..........?...
0x0010: 00 00 00 08 00 45 00 05 8C DF 77 40 00 FD 01 12        .....E....w@....
0x0020: 52 80 B7 0A 86 A5 15 56 54 00 00 D6 87 00 F2 89        R......VT.......
0x0030: 68 6D 61 69 6C 74 6F 3A 6F 70 73 40 64 69 67 69        hmailto:ops@digi
0x0040: 73 6C 65 2E 63 6F 6D 20 66 6F 72 20 71 75 65 73        sle.com for ques
0x0050: 74 69 6F 6E 73 20 20 20 20 54 68 69 73 20 49 43        tions    This IC
0x0060: 4D 50 20 45 43 48 4F 20 52 45 51 55 45 53 54 2F        MP ECHO REQUEST/
0x0070: 52 45 50 4C 59 20 69 73 20 70 61 72 74 20 6F 66        REPLY is part of
0x0080: 20 74 68 65 20 72 65 61 6C 2D 74 69 6D 65 20 6E        the real-time n
0x0090: 65 74 77 6F 72 6B 20 6D 6F 6E 69 74 6F 72 69 6E        etwork monitorin
0x00A0: 67 70 65 72 66 6F 72 6D 65 64 20 62 79 20 44 69        gperformed by Di
0x00B0: 67 69 74 61 6C 20 49 73 6C 61 6E 64 20 49 6E 63        gital Island Inc
0x00C0: 2E 20 20 49 74 20 69 73 20 6E 6F 74 20 61 6E 20        .  It is not an
0x00D0: 61 74 74 61 63 6B 2E 20 20 49 66 20 79 6F 75 20        attack.  If you
0x00E0: 68 61 76 65 71 75 65 73 74 69 6F 6E 73 20 70 6C        havequestions pl
0x00F0: 65 61 73 65 20 63 6F 6E 74 61 63 74 20 6F 70 73        ease contact ops
0x0100: 40 64 69 67 69 73 6C 65 2E 63 6F 6D 00 00 00 00        @digisle.com....
0x0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
0x0120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
0x0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
0x0140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00        ................
```

```
0x0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x0180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
[snip]

09/12-03:03:09.091229 EMP.NET.10.134 -> 210.181.96.101
ICMP TTL:253 TOS:0x0 ID:10461 IpLen:20 DgmLen:1420
Type:0  Code:0  ID:29440  Seq:6555  ECHO REPLY
0x0000: 50 AA 00 04 00 19 18 AA 00 04 00 FD 3F AA AA 03     P...........?...
0x0010: 00 00 00 08 00 45 00 05 8C 28 DD 40 00 FD 01 91     .....E...(.@....
0x0020: 3B 80 B7 0A 86 D2 B5 60 65 00 00 B2 CA 00 73 9B     ;......`e.....s.
0x0030: 19 6D 61 69 6C 74 6F 3A 6F 70 73 40 64 69 67 69     .mailto:ops@digi
0x0040: 73 6C 65 2E 63 6F 6D 20 66 6F 72 20 71 75 65 73     sle.com for ques
0x0050: 74 69 6F 6E 73 0A 54 68 69 73 20 49 43 4D 50 20     tions.This ICMP
0x0060: 45 43 48 4F 20 52 45 51 55 45 53 54 2F 52 45 50     ECHO REQUEST/REP
0x0070: 4C 59 20 69 73 20 70 61 72 74 20 6F 66 20 74 68     LY is part of th
0x0080: 65 20 72 65 61 6C 2D 74 69 6D 65 20 6E 65 74 77     e real-time netw
0x0090: 6F 72 6B 20 6D 6F 6E 69 74 6F 72 69 6E 67 0A 70     ork monitoring.p
0x00A0: 65 72 66 6F 72 6D 65 64 20 62 79 20 44 69 67 69     erformed by Digi
0x00B0: 74 61 6C 20 49 73 6C 61 6E 64 20 49 6E 63 2E 20     tal Island Inc.
0x00C0: 20 49 74 20 69 73 20 6E 6F 74 20 61 6E 20 61 74      It is not an at
0x00D0: 74 61 63 6B 2E 20 20 49 66 20 79 6F 75 20 68 61     tack.  If you ha
0x00E0: 76 65 0A 71 75 65 73 74 69 6F 6E 73 20 70 6C 65     ve.questions ple
0x00F0: 61 73 65 20 63 6F 6E 74 61 63 74 20 6F 70 73 40     ase contact ops@
0x0100: 64 69 67 69 73 6C 65 2E 63 6F 6D 00 00 00 00 00     digisle.com.....
0x0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x0120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x0140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x0180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x0190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x01A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
0x01B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
[snip]

09/12-03:04:32.060517 200.51.197.4 -> EMP.NET.10.134
ICMP TTL:245 TOS:0x0 ID:44555 IpLen:20 DgmLen:1420
Type:8  Code:0  ID:25344   Seq:843  ECHO
0x0000: 50 AA 00 04 00 FD 3F AA 00 04 00 19 18 AA AA 03     P.....?.........
0x0010: 00 00 00 08 00 45 00 05 8C AE 0B 40 00 F5 01 B9     .....E.....@....
0x0020: EF C8 33 C5 04 80 B7 0A 86 08 00 0D 7C 00 63 4B     ..3.........|.cK
0x0030: 03 6D 61 69 6C 74 6F 3A 6F 70 73 40 64 69 67 69     .mailto:ops@digi
0x0040: 73 6C 65 2E 63 6F 6D 20 66 6F 72 20 71 75 65 73     sle.com for ques
0x0050: 74 69 6F 6E 73 20 20 20 20 54 68 69 73 20 49 43     tions    This IC
0x0060: 4D 50 20 45 43 48 4F 20 52 45 51 55 45 53 54 2F     MP ECHO REQUEST/
0x0070: 52 45 50 4C 59 20 69 73 20 70 61 72 74 20 6F 66     REPLY is part of
0x0080: 20 74 68 65 20 72 65 61 6C 2D 74 69 6D 65 20 6E      the real-time n
0x0090: 65 74 77 6F 72 6B 20 6D 6F 6E 69 74 6F 72 69 6E     etwork monitorin
0x00A0: 67 70 65 72 66 6F 72 6D 65 64 20 62 79 20 44 69     gperformed by Di
0x00B0: 67 69 74 61 6C 20 49 73 6C 61 6E 64 20 49 6E 63     gital Island Inc
0x00C0: 2E 20 20 49 74 20 69 73 20 6E 6F 74 20 61 6E 20     .  It is not an
0x00D0: 61 74 74 61 63 6B 2E 20 20 49 66 20 79 6F 75 20     attack.  If you
```

```
0x00E0: 68 61 76 65 71 75 65 73 74 69 6F 6E 73 20 70 6C            havequestions pl
0x00F0: 65 61 73 65 20 63 6F 6E 74 61 63 74 20 6F 70 73            ease contact ops
0x0100: 40 64 69 67 69 73 6C 65 2E 63 6F 6D 00 00 00 00            @digisle.com....
0x0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x01A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x01B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x01C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x01D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x01E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x01F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0200: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
[snip]

09/12-03:04:32.064990 EMP.NET.10.134 -> 200.51.197.4
ICMP TTL:253 TOS:0x0 ID:27905 IpLen:20 DgmLen:1420
Type:0  Code:0  ID:25344  Seq:843  ECHO REPLY
0x0000: 50 AA 00 04 00 19 18 AA 00 04 00 FC 3F AA AA 03            P...........?...
0x0010: 00 00 00 08 00 45 00 05 8C 6D 01 40 00 FD 01 F2            .....E...m.@....
0x0020: F9 80 B7 0A 86 C8 33 C5 04 00 00 15 7C 00 63 4B            ......3.....|.cK
0x0030: 03 6D 61 69 6C 74 6F 3A 6F 70 73 40 64 69 67 69            .mailto:ops@digi
0x0040: 73 6C 65 2E 63 6F 6D 20 66 6F 72 20 71 75 65 73            sle.com for ques
0x0050: 74 69 6F 6E 73 20 20 20 20 54 68 69 73 20 49 43            tions    This IC
0x0060: 4D 50 20 45 43 48 4F 20 52 45 51 55 45 53 54 2F            MP ECHO REQUEST/
0x0070: 52 45 50 4C 59 20 69 73 20 70 61 72 74 20 6F 66            REPLY is part of
0x0080: 20 74 68 65 20 72 65 61 6C 2D 74 69 6D 65 20 6E             the real-time n
0x0090: 65 74 77 6F 72 6B 20 6D 6F 6E 69 74 6F 72 69 6E            etwork monitorin
0x00A0: 67 70 65 72 66 6F 72 6D 65 64 20 62 79 20 44 69            gperformed by Di
0x00B0: 67 69 74 61 6C 20 49 73 6C 61 6E 64 20 49 6E 63            gital Island Inc
0x00C0: 2E 20 20 49 74 20 69 73 20 6E 6F 74 20 61 6E 20            .  It is not an
0x00D0: 61 74 74 61 63 6B 2E 20 20 49 66 20 79 6F 75 20            attack.  If you
0x00E0: 68 61 76 65 71 75 65 73 74 69 6F 6E 73 20 70 6C            havequestions pl
0x00F0: 65 61 73 65 20 63 6F 6E 74 61 63 74 20 6F 70 73            ease contact ops
0x0100: 40 64 69 67 69 73 6C 65 2E 63 6F 6D 00 00 00 00            @digisle.com....
0x0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x0190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x01A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
0x01B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00            ................
[snip]
```

2.3.1 Source of trace

This trace was detected at one of my employer's centers. This trace was also from the same data set as the previous two detects.

2.3.2 Detect was generated by

Snort-1.8.1-RELEASE. Please refer to the previous detect for a description of the Snort log format and all relevant fields.in the output.

2.3.3 Probability the source address was spoofed

It is unlikely that these packets are spoofed. This looks like a probe to determine the MTU of the victim host. If this is true, the attacker will need to receive these packets in order to learn the MTU of the victim.

2.3.4 Description of this attack

This is not an attack, it is a function. The idea behind Path MTU Discovery is to optimize network performance by sending the largest possible IP datagrams that will not require fragmentation and reassembly. By using MTU Discovery, an attacker can gather information about the victim network (the MTU) or use the process to attempt a DOS by flooding the network. A description of the DOS is located: http://www.securiteam.com/securitynews/5AP0D2A35U.html

2.3.5 Attack mechanism

At first glance, this looks like an attempt at MTU discovery. Once we look at the actual payload we notice that there is actually a message embedded in the ICMP datagram.

```
[snip]
0x0030: 03 6D 61 69 6C 74 6F 3A 6F 70 73 40 64 69 67 69   .mailto:ops@digi
0x0040: 73 6C 65 2E 63 6F 6D 20 66 6F 72 20 71 75 65 73   sle.com for ques
0x0050: 74 69 6F 6E 73 20 20 20 20 54 68 69 73 20 49 43   tions    This IC
0x0060: 4D 50 20 45 43 48 4F 20 52 45 51 55 45 53 54 2F   MP ECHO REQUEST/
x0070: 52 45 50 4C 59 20 69 73 20 70 61 72 74 20 6F 66    REPLY is part of
0x0080: 20 74 68 65 20 72 65 61 6C 2D 74 69 6D 65 20 6E   the real-time n
0x0090: 65 74 77 6F 72 6B 20 6D 6F 6E 69 74 6F 72 69 6E   etwork monitorin
0x00A0: 67 70 65 72 66 6F 72 6D 65 64 20 62 79 20 44 69   gperformed by Di
0x00B0: 67 69 74 61 6C 20 49 73 6C 61 6E 64 20 49 6E 63   gital Island Inc
0x00C0: 2E 20 20 49 74 20 69 73 20 6E 6F 74 20 61 6E 20   .  It is not an
0x00D0: 61 74 74 61 63 6B 2E 20 20 49 66 20 79 6F 75 20   attack.  If you
0x00E0: 68 61 76 65 71 75 65 73 74 69 6F 6E 73 20 70 6C   havequestions pl
0x00F0: 65 61 73 65 20 63 6F 6E 74 61 63 74 20 6F 70 73   ease contact ops
0x0100: 40 64 69 67 69 73 6C 65 2E 63 6F 6D 00 00 00 00   @digisle.com
[snip]
```

This message seems to clear up the issues associated with this packet. Nevertheless, this packet is unwanted on the network as it is using valuable bandwidth. The question still remains, what is happening here?

Upon further investigation, this still looks like an attempt at MTU discovery – instead of the typical MTU discovery which has "00" (null) as the padding for the payload, this particular attack has a "disclaimer" if you will, associated with it. Also, these packets are originating from different hosts. After a review of the entire day's logs, there were 54 IP addresses that had sent packets identical to the ones listed above. This more strongly associates this trace as an attack, rather than a valid probe from digisle.com.

There are two factors that make this trace difficult to pinpoint as an assured MTU discovery attack. For one, the DF (don't fragment) bit is not set. This bit is usually set in an MTU attack. This allows the attacker to gather information from the router because if the packet is too large to pass, the router will reply with an ICMP type 3 code 4 (fragmentation required, DF set) message if the DF bit is set. If the attacker methodically decrements the packet size, they will know the MTU when they no longer receives error messages. The fact that this is missing from the packet may mean that the attacker is attempting a DOS (as referenced earlier). The second issue, is that the packet size is 1420 as opposed to the more common 1500. The MTU of the victim in this case is actually 4470 (FDDI). It is understandable why the packet isn't 4470 (the attackers MTU is probably not 4470), but why isn't it at least 1500? If this is an attack, the attacker is clearly attempting to make this an uncommon and therefore less detectable attack. Maybe the attacker slightly decreased the packet size so as to avoid IDS's that look for 1500 byte ICMP packets. These two issues, while hypothetically explainable, nevertheless make it difficult to determine exactly what is going on here.

2.3.6 Correlation

There are some references of this type of activity on the internet. I also emailed the organization and received (nearly 2 months later) an email from them describing this activity. The email is listed below. While there is a legitimate excuse for *some* of this activity to occur, the prevalence and the number of hosts generating these packets is odd. The email also says that this traffic should be directed at DNS servers, and the victim host is *not* a DNS server. This traffic is also originating from Korea, which is odd, because neither Sandpiper or Digisle are directly associated with that geographical region. Therefore, this looks an MTU discovery that is cleverly disguised as legitimate traffic. Note that this traffic may be legitimate even though it looks overly prevalent. Either way, it would seem that a good way to (at least) add confusion to a security analyst's detect would be to add comments in the datagram that make the packet look legitimate. It would have been easy to discard these alerts as "warranted" traffic due to the disclaimer in the datagram. This may be something for analysts to keep in mind in the future.

The organizations site: http://www.digisle.com/
Description of MTU discovery: http://www.worldgate.com/~marcs/mtu/

RFC: http://www.faqs.org/rfcs/rfc1191.html
Email from digisle:

Jamil,

I have enclosed the reasoning behind what is going on...

We apologize for any inconvenience caused by pings (ICMP_ECHO packets) coming from our machines. Your server was being pinged as part of our real-time "network weather" mapping system called Best Distributor Selection. BDS is an essential part of Footprint, Digital Island's intelligent network service offering. It is used to optimize performance when your customers access the web resources of our customers.

Many large web publishers, such as AOL, CNBC and Blue Mountain, use our Footprint service to speed up the delivery of their web content. Our system intelligently matches browsers to the servers on our Footprint network that will provide the best performance. The dynamic nature of routing and congestion on the Internet make it necessary for us to constantly update our maps.

Our network was pinging your system because it appeared to be a name server with a sufficient number of resolution requests for our customer web sites to be placed on the list of network nodes to be constantly observed for Internet congestion.

By pinging your name server, we can provide better quality of service to your users when they access the web sites of our expanding customer list. We hope you will consider granting us permission to continue pinging a name server in your domain.

Sandpiper Networks merged with Digital Island in December 1999, which is why some of the machines pinging you were in digisle.net.

At this point you can:

1) Do nothing. Please accept our apologies and be assured that your machines are not being pinged by a hostile party.

2) Tell us if there is an alternate name server in your IP address space that you would like us to ping. We will direct future ping traffic to it.

3) Respond to this message requesting we stop pinging your server. In this event our pinging will cease in several days.

2.3.7 Evidence of active targeting

This attack was not actively targeting this host. Upon review of more logs, it is clear that this probe was attempting to identify the MTU of all hosts in the class B range that I monitor. There were certain hosts that had no probes directed at them, but I believe that there were probes but the IDS did not catch them due to the moderate packet loss indicative of that IDS.

2.3.8 Severity

Direction	Category	Value	Reasons
Attack	Criticality	2	This system is a basic workstation
	Lethality	2	If successful, this attack will determine the MTU of the host, this alone will not do a significant amount of damage but there is the possibility of a DOS.
Response	System	2	The system replied and therefore presumably gave the attacker the information they were requesting.
	Network	1	The router and the firewall both missed this attack and allowed it through to the network.
Severity = (2 + 2) – (2 + 1) = 1			

2.3.9 Defense

To defend against this probe we should disallow all ICMP Echo Requests from external hosts. This action would eliminate these packets before they reached the network.

2.3.10 Multiple Choice Question

What risk do you run if you allow ICMP Echo Requests into your network?

a. ICMP Denial of Service attack
b. Trojan horse installation
c. OS Fingerprinting
d. Session hijacking
e. No benefit

Answer: A. There are many issues with allowing ICMP into the network and it ultimately depends on what the security policy of the network is. The more lax security networks will allow ICMP and run the risk of a DOS knowing that all TCP connections will not break. A more security conscious network policy will disallow all ICMP and know that an ICMP DOS will not be an issue but may have problems with TCP connections dropping due to MTU and MSS issues. It all depends on the policy, but one thing there is no question about – if you allow ICMP, you are more prone to receive an ICMP DOS attack.

2.4.0 Detect 4, Malformed IGMP

```
09/12-13:10:20.634139 61.147.220.23 -> MY.NET.160.242
IGMP TTL:39 TOS:0x0 ID:5341 IpLen:20 DgmLen:572 MF
Frag Offset: 0x0   Frag Size: 0x228
```

```
0x0000: 50 AA 00 04 00 19 18 AA 00 04 00 FC 3F AA AA 03         P..........?...
0x0010: 00 00 00 08 00 45 00 02 3C 14 DD 20 00 28 02 72         .....E..<.. .(.r
0x0020: CA 3D 93 DC 17 A9 9A 25 D4 00 00 00 00 00 00 00         .=.....%........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
[snipped for brevity]

09/12-13:10:20.634987 61.147.220.23 -> MY.NET.160.242
IGMP TTL:38 TOS:0x0 ID:5341 IpLen:20 DgmLen:572 MF
Frag Offset: 0x0   Frag Size: 0x228
0x0000: 50 AA 00 04 00 19 18 AA 00 04 00 FC 3F AA AA 03         P..........?...
0x0010: 00 00 00 08 00 45 00 02 3C 14 DD 20 00 26 02 74         .....E..<.. .&.t
0x0020: CA 3D 93 DC 17 A9 9A 25 D4 00 00 00 00 00 00 00         .=.....%........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
[snipped for brevity]

09/12-13:10:20.635289 61.147.220.23 -> MY.NET.160.242
IGMP TTL:37 TOS:0x0 ID:5341 IpLen:20 DgmLen:572 MF
Frag Offset: 0x0   Frag Size: 0x228
0x0000: 50 AA 00 04 00 FC 3F AA 00 04 00 19 18 AA AA 03         P.....?.........
0x0010: 00 00 00 08 00 45 00 02 3C 14 DD 20 00 25 02 75         .....E..<.. .%.u
0x0020: CA 3D 93 DC 17 A9 9A 25 D4 00 00 00 00 00 00 00         .=.....%........
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
0x00E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00         ................
```

```
0x00F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
0x0100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
0x0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00          ................
[snipped for brevity]
```

2.4.1 Source of trace

This trace was found on my personal network.

2.4.2 Detect was generated by

A Pentium 166 MHz box with two 10mb nics. This box is the gateway to my internal network. The program used to capture this trace was Snort-1.8.1-RELEASE; the system is the gateway to my personal network.

2.4.3 Probability the source address was spoofed

The source address has a high probability of being spoofed. This attack does not require the attacker to receive any information. It is very likely that the attacker has spoofed the source address to avoid detection.

2.4.4 Description of this attack

This attack is based upon a failure of the Windows 95/98/NT/2000 operating systems to handle malformed IGMP packets. If the target is vulnerable to these fragmented packets, a number of things can happen to the system. Some of the notable mishaps resulting from this attack are a blue screen or system crash.

2.4.5 Attack mechanism

This trace is a denial of service attack against my machine. The attacker is attempting to disrupt operations on this system. By flooding the system with a number of malformed IGMP packets (8546 in this case), the attacker can crash vulnerable machines. There are some oddities about these packets. The IP id is the same in all of the packets, this number should not stay the same in sequential packets. According to Richard Steven's book, TCP/IP Illustrated, "The identification field uniquely identifies each datagram sent by a host. It normally increments by one each time a datagram is sent." Therefore, this is clearly not normal IP activity. Also, we can see that these packets are fragmented into 552 byte chunks. While this alone is not extrememly odd, but in addition to this, there are more oddities. The fragmentation offset field in the IP header tells us the offset of this fragment from the beginning of the original datagram. This packet is identifying itself as the first packet because of the frag offset of 0. Why are there 8000+ of these packets that look exactly the same? Well, it MAY be that the sending host is retransmitting this packet, that seems to be a possibility. If we look closer though, that does not seem to be the case. The TTL's of these packets is decrementing. This is not likely to happen naturally, expecially in such a short amount of time (there is very little time between each packet). Also, with such a short time between packets, it would seem that this is most likely not a manual process. Finally, to make these packets even more curious, they are originating from China. China has become notorious in the security community for being a haven for hacker activity. While this has no direct implications on the attack, it should be noted.

2.4.6 Correlations

This attack is described in detail on the following web sites:

1. Bugtraq: http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=info&id=514
2. Here is the official description of the vulnerability from Microsoft: http://support.microsoft.com/support/kb/articles/q238/3/29.asp
3. There are three programs on the Bugtraq site that can be used to exploit this vulnerability: http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=exploit&id=514

   Of the three programs listed on the Bugtraq site, kod.c looks to be the program that creates packets most similar to the trace because of the default TTL and other IP header options. Unfortunately, even though it looks similar, they are not exact, after a review of the source code, while many similarities are present in the header, the program does not include code to decrement the TTL. This is a minor obstacle though, because it would be trivial for someone to modify the source code for their own use.

2.4.7 Evidence of active targeting

After a review of two weeks of logs, there were no other traces of this type. This would lead me to believe that this attack was specifically targeting this host. The only oddity about this assessment is the fact that the victim host is not vulnerable to this attack so it seems illogical for the attacker to target a host with an attack that it is not vulnerable to. When searching for other instances of this attack I looked for the specific signature of the attack, not the source host's address because the address can be easily spoofed in this attack. Ultimately, I do not believe that this attack is an example of active targeting due to the fact that this host is not vulnerable to this DOS because it is a windows specific vulnerability and this host is a BSD system. I assume the attacker is randomly using this program on various addresses.

2.4.8 Severity

Direction	Category	Value	Reasons
Attack	Criticality	4	This system is the gateway to my personal network, it is a very vital system to the proper operation of my network.
	Lethality	3	If successful, this attack can cause the victim to crash, leading to downtime and possible

			data corruption.
Response	System	5	This computer is running the Linux operating system, which is not susceptible to this attack.
	Network	3	The network detected the attack but the firewall did not stop the 8546 packets.
Severity = (4 + 3) – (5 + 5) = -1			

2.4.9 Defensive recommendations

This attack should be turned away by the border router or external firewall. If IGMP is a necessary inbound protocol, then all Windows 95/98/NT/2000 computers should be patched immediately.

2.4.10 Multiple choice question

Is this attack executed manually or with a tool? If a tool has been used how can you tell?

a. This is a manual attack – no tool was used
b. Same IPID for all packets
c. Decrementing TTL
d. Same source and destination IP address for all packets
e. Both b and c

Answer: E. The IPID should not be constant and the TTL should not be decrementing if it continues to be sent by the same host to the same destination. These characteristics could be accomplished manually but if you look at the time when the packets were received, it becomes clear that no human can manually send this many packets in such a short period of time.

2.5.0 Detect 5, @Home Scanning

```
[**] [104:1:1] spp_anomsensor: Anomaly threshold exceeded: 7.5731 [**]
08/24-06:13:35.121270 24.0.0.203:57194 -> MY.NET.160.242:119
TCP TTL:248 TOS:0x0 ID:7179 IpLen:20 DgmLen:44
******S* Seq: 0xD42B95C1  Ack: 0x0  Win: 0x2238  TcpLen: 24
TCP Options (1) => MSS: 1460

[**] [104:1:1] spp_anomsensor: Anomaly threshold exceeded: 7.5731 [**]
08/24-06:13:35.121999 MY.NET.160.242:119 -> 24.0.0.203:57194
TCP TTL:255 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0x0  Ack: 0xD42B95C2  Win: 0x0  TcpLen: 20

[**] [104:1:1] spp_anomsensor: Anomaly threshold exceeded: 7.5075 [**]
08/24-06:13:49.548221 24.0.0.203:62900 -> MY.NET.160.242:119
TCP TTL:248 TOS:0x0 ID:7180 IpLen:20 DgmLen:44
******S* Seq: 0xEA7E96A5  Ack: 0x0  Win: 0x2238  TcpLen: 24
TCP Options (1) => MSS: 1460
```

```
[**] [104:1:1] spp_anomsensor: Anomaly threshold exceeded: 7.5075 [**]
08/24-06:13:49.548685 MY.NET.160.242:119 -> 24.0.0.203:62900
TCP TTL:255 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0x0  Ack: 0xEA7E96A6  Win: 0x0  TcpLen: 20
```

2.5.1 Source of trace

This trace was found on my personal network.

2.5.2 Detect was generated by

The detect was *generated* by the Snort-1.8.1-RELEASE facilities and was *detected* by Spade-092200.1.

2.5.3 Probability the source address was spoofed

The source address has a low probability of being spoofed. The attacker will most definitely want to receive the results of this query.

2.5.4 Description of this attack

This is not an attack, it is a scan which looks for the presence of certain services. At first glance, I thought that it was old worm attack called Happy99. It is a virus that is sent through email and news groups. Once it infects a computer, it opens a window that displays "Happy New Year 1999 !!". While this graphic is displayed, the program installs itself on the victim's computer. Once compromised, the victim computer will then attempt to exploit other machines. Fortunately, after some research I found out otherwise.

2.5.5 Attack mechanism

Initially, I thought this trace was the Happy99 worm attempting to infect my system. I came to this conclusion simply because of the port number. When a system is compromised with this Trojan, every email or Usenet message sent by the host is followed by an attachment containing this program. The curious thing about this attack was the originating address. This particular instance of this attack was originating from authorized-scan1.security.home.net (24.0.0.203). I thought this address was odd and chose to look into it further. Upon investigation, I found that this activity is in relation to @Home scanning their customers to see if they are running USENET services in violation of their contract. So this is actually a false positive, but an interesting one, because I did not know that @Home scanned me. This activity is legal too, because @Home is my Internet Service Provider at my residence. After a review of the logs, it is clear that this was @Home. If we look at the alerts, we can see that the packets destined for my machine are simply TCP SYN packets. I think that the scan is simply attempting to execute a three way handshake. Notice though, I am not running anything on that port because my system replied with a ACK RST which essentially tears down the attempted handshake. The only other thing peculiar with the initiating packets is that there is a TCP option. This option is actually common though, as all it does is set the Maximum

Segmentation Size (MSS), which means that this size is the largest chunk of data that TCP will send to the other end. An MSS of 1460 is a common because according to tests, this provides better performance on an Ethernet network. With all of these points factored together, it makes a firm case that this is in fact @Home scanning my machine.

2.5.6 Correlations

The information about @Home scanning is sparse, but here is what I found:
Good discussion on the topic:
http://groups.google.com/groups?hl=en&threadm=rto3dsoee5og0202c4ihfuu5i6b64vflss%404ax.com&rnum=1&prev=/groups%3Fq%3D%2540Home%2Bport%2Bscanning%26hl%3Den%26rnum%3D1%26selm%3Drto3dsoee5og0202c4ihfuu5i6b64vflss%25404ax.com
Article which briefly references @Home scanning:
http://www.robertgraham.com/pubs/firewall-seen.html

The Happy99 worm is described in detail on the following web sites:

1. http://www.symantec.com/avcenter/venc/data/happy99.worm.html
2. http://www.zdnet.com/zdnn/stories/news/0,4586,2208275,00.html
3. http://www.cert.org/incident_notes/IN-99-02.html

2.5.7 Evidence of active targeting

This is clearly active targeting. @Home is actively targeting all users of their service. I happened to be one of these individuals and therefore was a victim of the scanning. I have read discussions about this activity where @Home was actually scanning IP addresses of individuals who were not @Home subscribers. In my case, @Home actively targeted me because I was a subscriber.

2.5.8 Severity

Direction	Category	Value	Reasons
Attack	Criticality	4	This system is the gateway to my personal network, it is a very vital system to the proper operation of my network.
	Lethality	0	This is simply a scan to see if my system is running USENET services, it poses no threat to my machine or network.
Response	System	4	This scan was detected by my IDS on the machine. The packets were not deterred though, they reached the machine and it replied.
	Network	0	The network did not stop these packets from reaching the host because we can see replies from the victim host being sent to the attacker.

Severity = (4 + 0) – (4 + 0) = 0

2.5.9 Defensive recommendations

There is no way to stop @Home from scanning your machine if you are a subscriber of their service. The contract actually states that they may probe your machine (although the words “port” and “scan” are not in the contract). To deny these specific probes, simply block all incoming traffic to port 119, or if you are running a service on these ports and want to continue their use, simply drop (or optimally, reply with a RST) all packets from security.home.net destined to port 119.

2.5.10 Multiple choice question

What type of scan does @Home use to determine if users are running a service on port 119?

a. NULL scan
b. XMAS tree scan
c. TCP connect
d. SYN scan

Answer: C. @Home simply attempts a three way handshake (SYN, SYN-ACK, ACK) with the service on port 119. The TCP connect scan does just that, if the three way handshake is completed it assumes the service is available. If the host is not offering the service, the three way handshake will never be completed.

3.0 Data Analysis and Assessment

This section will provide a security audit for a University.

3.0.1 Executive Summary

Thank you for providing us with an opportunity to assess and investigate the security of your network. We have analyzed 5 days of IDS logs to assess the security of your University. The data set that was used spans from September 5th 2001 through September 9th 2001. We were given alert, scan, and out of spec logs. Unfortunately, the data does not consecutively span the entire 120 hour frame that the 5 days of logs should account for. There are two significant segments of time that are not present in the log data. The analysis team was also hampered by a limited knowledge of the network information, in terms of architecture and design. Without these fundamental building blocks of analysis, there was a need to make some general assumptions to further improve the accuracy and the validity of our analysis. The assumptions used to ascertain our analysis are as follows:

1. The network being monitored is a University environment – freely accessible information is required
2. This IDS is located at the gateway between the border router and the internal network.
3. Few (if any) access controls are currently implemented
4. There are numerous unique operating systems and flavors on the network. (Windows, Linux, UNIX, BSD, etc.)
5. There are a variety of services that are utilized on the systems within the network (i.e. httpd, smtpd, databases, etc).

While there are many things to be proud of in regards to the security of the University, there are also numerous issues that must be addressed before this network can be considered secure. The remainder of this document will describe the methods, tools, and approach we used to derive our results. We will describe the analysis process, vulnerabilities we encountered, and countermeasures you should take to alleviate these potential hazards. We will begin with a high-level overview of our findings.

Alerts

There were an extremely diverse number of alerts detected by the IDS on the network. This should be expected in a University environment with numerous hosts and relatively lax access controls. A number of the alerts that were detected were false positives so it would be advisable to refine the rule set of the IDS. The more false positives an analyst is approached with, the less time the analyst will have to spend on actual incidents. Also, if an analyst spends too much time seeing false positives, there is a much greater chance that they will miss actual malicious traffic due to such an overwhelming amount of false positives. The primary recommendation to address the current alerts is to upgrade all hosts with the latest software, implement some access controls for unneeded traffic, and (if possible) audit all the hosts on the network. If nothing else, all the hosts that are identified in this report should be audited thoroughly. While we realize that this is a University environment and it should be relatively free of access controls, there is a fine line between information freedom and irresponsibility. By implementing a firewall at the border router and blocking basic ports such as 135-139 and 445, the network security will be enhanced greatly. The primary service that attackers are attempting to exploit is WWW; specifically the Microsoft IIS web server. We recommend that the University lock these hosts down by disabling all unneeded services and also patched and updates all required software on the machines. With these few simple steps, the security of the University will improve greatly.

Scans

Numerous scans were seen on the University network. Some of the scans originated from within the network. This is cause for concern, as some of the internal hosts may be compromised. The logs that we have provided specifically show the internal hosts that are scanning. Beyond that, there are a plethora of scans that are penetrating into the local network. Some attacking hosts have even been scanned over 1000 times. The primary hosts that scanned the network have been researched to find registration information. It is listed in the registration information section. Scans are difficult to stop, and some are very difficult to detect. The scans that were found were not necessarily harmful, but the information the attacker derived from the scan could possibly be used to exploit the machine at a later time. Whenever a scan has been seen, there should be an investigation as to how much data the attacked host(s) revealed. We recommend that the University implement a firewall to deny blatant portscans. Most firewall packages can and will block packets destined to an abundant number of ports or machines over a short period of time. With a firewall in place, the University will not be invulnerable to portscans but will have a moderate defense against them.

OOS data

The out-of-spec data logs were without a doubt the most meager set of data provided in relation to the other logs. Nevertheless, this data provided us with numerous scan and attack attempts. For the most part, the OOS data was reconnaissance work. The attackers sent malformed packets to trick the receiving system into replying with revealing information. A firewall will stop these packets from entering the network. Most firewall packages are built to stop packets just like the ones seen in the logs provided to us by the University. Packets that have conflicting flags and options should not be let into the network, and therefore, a firewall would greatly benefit the University network.

Overall

The University should implement some access controls. There is clear evidence of hosts that are compromised within the network. There is also ample evidence of active targeting. With this nefarious activity pulsing through the network, the chances of a compromise increase dramatically. This document will detail the activity we saw on the network over the 5 day time span. While some of the dates have limited data due to data loss or downtime, there is a plethora of useful evidence which confirms the need for some access controls.

Analysis of Data

3.1.0 Overview of Results:

All statistics in this section were derived from various Snort analysis front ends. Due to the lack of binary log data, the selection of analysis front ends was limited to **snort_stat** and **Snortsnarf**. These programs can read alert data in plain text and do not need binary logs like programs such as **ACID** require. While analysis would be more complete and effective if the binary logs were available, the combination of the previously named programs and manual analysis produced a reliable estimation of the network activity.

NOTE: Due to the length of the processed logs, some charts have been snipped for brevity.

3.1.1 Data Files

The files used to derive this information are as follows:

```
File size     Date/Time of last access     Filename (<alert type>.<date>.tar.gz.txt)
==================================================================
18096964      Oct  2 22:13                 alert.010905.gz.txt
15003667      Oct  2 22:21                 alert.010906.gz.txt
2250025       Oct  2 22:21                 alert.010907.gz.txt
7179780       Oct  2 22:21                 alert.010908.gz.txt
16497681      Oct  2 22:21                 alert.010909.gz.txt
16388         Oct  2 22:21                 oos_Sep.5.2001.gz.txt
14569         Oct  2 22:21                 oos_Sep.6.2001.gz.txt
19852         Oct  2 22:21                 oos_Sep.7.2001.gz.txt
15273         Oct  2 22:21                 oos_Sep.8.2001.gz.txt
7720          Oct  2 22:21                 oos_Sep.9.2001.gz.txt
3523738       Oct  2 22:22                 scans.010905.gz.txt
2131241       Oct  2 22:22                 scans.010906.gz.txt
6170704       Oct  2 22:22                 scans.010907.gz.txt
2997266       Oct  2 22:22                 scans.010908.gz.txt
3847513       Oct  2 22:22                 scans.010909.gz.txt

==================================================================
```

3.1.2 Attack Chart, Top Attack Talkers

The log begins from: 09 05 00:08:03
The log ends   at: 09 09 23:34:43
Total events: 2173
Source IP recorded: 215
Destination IP recorded: 597
Portscan recorded: 2427

Percentage and number of attacks from a host to a destination

```
=============================================================
      #  of
 %   attacks   from             to
=============================================================
48.00   1043     3.0.0.99         10.0.0.1
 6.35   138      164.107.98.247    164.107.3.40
 3.45   75      64.210.135.86     10.0.3.2
 1.06   23      198.180.47.169    198.180.47.156
 0.92   20      3.0.0.0          3.0.0.0
 0.55   12      192.168.1.100     152.7.56.43
 0.55   12      192.168.1.100     24.181.69.9
 0.46   10      64.210.135.86     10.0.3.3
 0.41   9      169.254.101.152   65.88.67.84
 0.37   8      172.139.203.5     213.161.66.183
 0.37   8      64.210.135.86     64.14.124.200
 0.32   7      164.107.98.247    164.107.3.55
 0.28   6      192.168.1.106     24.3.0.36
 0.23   5      169.254.101.152   205.188.46.122
 0.23   5      164.107.98.247    164.107.3.39
 0.23   5      172.128.110.34    24.66.230.197
 0.23   5      172.149.208.213   24.141.213.248
 0.18   4      172.143.120.184   211.196.213.66
 0.18   4      63.168.24.120     192.168.0.2
 0.18   4      169.254.101.152   217.128.205.187
 0.18   4      169.254.101.152   202.167.127.59
 0.18   4      172.168.92.43     63.93.203.10
 0.18   4      172.128.110.34    24.160.204.168
 0.18   4      172.170.142.222   211.196.213.66
 0.14   3      169.254.101.152   205.188.46.120
 0.14   3      172.153.216.67    207.113.114.192
 0.14   3      172.163.138.104   216.34.90.142
 0.14   3      172.150.208.59    212.47.176.170
 0.14   3      192.168.1.100     213.123.167.241
```

0.14	3	172.147.73.119	216.183.18.189
0.14	3	172.148.101.200	204.228.82.198
0.14	3	216.45.78.179	217.5.86.160
0.14	3	192.168.1.100	64.229.110.203
0.14	3	172.149.47.175	66.20.60.90
0.09	2	172.128.110.34	24.17.96.167
0.09	2	172.132.254.40	148.243.183.138
0.09	2	172.139.40.103	148.223.78.226
0.09	2	192.168.1.100	24.80.3.153
0.09	2	169.254.101.152	172.150.6.115
0.09	2	169.254.165.58	169.70.97.19
0.09	2	192.10.14.104	192.10.25.105
0.09	2	169.254.101.152	205.188.46.254
0.09	2	216.45.78.179	63.121.237.202
0.09	2	169.254.134.92	63.102.227.40
0.09	2	216.45.78.179	62.83.102.224
0.09	2	216.45.78.179	32.101.14.141
0.09	2	172.139.44.24	192.168.1.10
0.09	2	192.168.1.106	65.5.120.94
0.09	2	192.168.1.106	24.3.122.153
0.09	2	172.135.77.166	12.31.248.99
0.09	2	169.254.165.58	33.77.131.55
0.09	2	172.129.61.51	205.188.70.102
0.09	2	169.254.165.58	169.148.249.39

[snipped for brevity]

Analysis of attack chart

These statistics show that the most numerous number of attacks are originating from 3.0.0.99 and are directed at 10.0.0.1 with an overwhelming 48% of the attacks logged. This is cause for further investigation to derive what type of activity 3.0.0.99 is throwing at 10.0.0.1. The remainder of the statistics in this chart show a moderate to light amount of activity between the remaining hosts. Do not be deceived though, to exploit a computer it only takes 1 successful attack. We must also consider the fact that we do not know how many attacks passed through the IDS without detection. We will analyze the remaining data before we make any decisions.

3.0.0.99, while generating extensive activity, only talked to one host on one port throughout the course of one day.

```
09/07-23:15:20.348521  [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
09/07-23:15:21.850649  [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
09/07-23:15:38.374010  [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
09/07-23:17:27.864088  [**] UDP SRC and DST outside network [**] 3.0.0.99:137 -> 10.0.0.1:137
```

Neither of these addresses are routable on the Internet. The University IDS, since it is sitting in real address space should not ever see traffic like this. Port 137 is common for

NetBIOS SMB service (http://www.sans.org/newlook/resources/IDFAQ/port_137.htm) for full and proper analysis of this attack we need more information, like the actual packet decodes, so we can analyze the actual traffic.

3.1.3 Portscan Chart, Top Portscan Talkers

Portscans performed to/from HOME_NET

```
====================================
 # of
attacks from
====================================
874    204.50.141.133
193    216.169.181.240
134    217.162.127.5
91     202.204.128.13
91     64.123.43.242
82     213.93.146.50
81     217.80.178.7
36     141.156.45.125
27     205.188.244.57
25     198.186.202.147
24     205.188.246.121
23     64.160.48.11
22     61.134.9.121
20     205.188.233.185
19     61.153.17.244
17     205.188.244.121
16     63.206.137.117
15     199.183.24.194
14     211.73.191.130
14     211.255.136.74
13     205.188.233.121
12     164.106.165.170
11     205.188.233.153
11     128.46.156.155
8      61.153.17.24
8      64.225.140.113
7      130.207.193.70
7      63.196.4.89
7      148.63.224.11
7      151.38.84.194
7      193.137.96.74
6      195.170.5.2
6      217.1.76.109
6      148.63.18.139
```

6	134.197.14.245
5	211.63.185.21
5	216.65.211.218
5	209.193.48.102

[snipped for brevity]

Analysis of portscan chart

We notice that there are numerous portscans against our local network. At first glance, this looks to be an extensive number of portscans to/from our networks. Please note that this chart may be deceiving for the fact that portscans in Snort are determined in two ways; the first is by the number of connections to a host over a certain time period; the second is the number of connections to specific hosts over a certain time period. DNS servers have a tendency to trip the portscan threshold on Snort. Other services that use UDP or require multiple connections in a short time period also cause false-positives. If the IDS is not configured to ignore these hosts, we will receive numerous false-positives in the logs. Nevertheless, there is clearly a lot of portscan activity on this network, so further analysis is necessary to determine the intent of the hosts Snort alerted us to. We should note the primary talkers on this list: 204.50.141.133, 216.169.181.240, and 217.162.127.5.

204.50.141.133 (vickesh01-733.tbaytel.net) is the primary scanning host. This host is only doing NULL scans, which means that the packets the attacker is sending are TCP packets with no flags set. Faud Khan has a brief analysis of null scans in his assessment of this network. The paper is located at **http://www.sans.org/y2k/practical/Faud_Khan_GCIA.doc.** These scans are directed at MY.NET.105.120 port 0, which is indicative of a null scan. This attacker also later uses both a FIN and an XMAS scan against this host. The FIN scan is a TCP scan that uses on the FIN flag to probe the host. The XMAS scan is a scan with all the TCP flags set and usually has various other options set as well. The packet basically looks like a "lit up Christmas tree" hence the name. Refer to section 3.6.0 (redworm) for further analysis of this attacking host.

```
[snip]
09/06-15:31:03.041690  [**] spp_portscan: PORTSCAN DETECTED from 204.50.141.133 (STEALTH)
[**]
09/06-15:15:03.074049  [**] Null scan! [**] 204.50.141.133:0 -> MY.NET.105.120:0
09/06-15:15:04.852792  [**] Null scan! [**] 204.50.141.133:0 -> MY.NET.105.120:0
09/06-15:31:04.557205  [**] spp_portscan: portscan status from 204.50.141.133: 1 connections across 1
hosts: TCP(1), UDP(0) STEALTH
[**]
09/06-15:31:05.975592  [**] spp_portscan: End of portscan from 204.50.141.133: TOTAL time(1s)
hosts(1) TCP(1) UDP(0) STEALTH [**]
09/06-15:31:06.602456  [**] spp_portscan: PORTSCAN DETECTED from 204.50.141.133 (STEALTH)
[**]
09/06-15:15:12.780448  [**] Null scan! [**] 204.50.141.133:0 -> MY.NET.105.120:0
09/06-15:15:13.831650  [**] Null scan! [**] 204.50.141.133:0 -> MY.NET.105.120:0
09/06-15:31:07.710468  [**] spp_portscan: portscan status from 204.50.141.133: 1 connections across 1
hosts: TCP(1), UDP(0) STEALTH
```

```
[**]
09/06-15:31:09.139385  [**] spp_portscan: End of portscan from 204.50.141.133: TOTAL time(1s)
hosts(1) TCP(1) UDP(0) STEALTH [**]
[snip]
```

3.1.4 Top Talkers, Combination of All Logs

Host	Number of Triggers
MY.NET.218.78	32871
MY.NET.201.42	26900
MY.NET.213.6	19914
205.188.246.121	19452
205.188.244.57	14219
205.188.233.185	13849
MY.NET.234.162	12282
134.197.14.245	3646
209.10.56.41	2528

The “Top Talkers” list is a calculation of the hosts that occur most frequently in the logs. The logs that were analyzed to derive the following chart are all of the alert, scan, and OOS data. These logs were run through a series of piped UNIX commands to determine the IP addresses that appeared most often. Without the binary data, it was more difficult to process the logs because more conventional tools such as ACID were not at my disposal. This chart is a good base line to determine the hosts both within and outside the University network that should receive more attention by the security team. These numbers can be used to find hosts that are more susceptible to attack and may also be used determine hosts that are causing a lot of false positives and should therefore be ignored by the IDS.

The host that was queried the most MY.NET.218.78 seems to be relatively safe. The majority of activity was triggered from scanning. There is only one actual attack that was recorded against this host.

```
09/06-09:32:46.240016  [**] High port 65535 TCP - possible Red Worm - traffic [**]
206.222.196.76:65535 -> MY.NET.218.78:1214
```

There was no recorded reply to this attack so this host seems to be unaffected by this attack. For more information on the Red Worm attack refer to section 3.6.0. It should also be noted that this host was triggered a few times because of a watch list.

```
09/05-13:35:31.202530  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.85.180:2176 ->
MY.NET.218.78:1214
09/05-14:20:54.073052  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.86.6:3108 ->
MY.NET.218.78:1214
09/05-16:08:59.447626  [**] Watchlist 000220 IL-ISDNNET-990517 [**] 212.179.83.99:1056 ->
MY.NET.218.78:1214
```

The only commonality with these alerts is the destination host and port. The common service associated with this port is Kazza using both TCP and UDP. The university must have or have had issues with this Trojan to implement rules to identify it specifically. The MY.NET host never replied to any of the inbound packets, which means that this host seems to be ok. It should be audited regardless though, because of the sheer number of probes it received.

3.2.0 Registration Information About the Five Top Suspects

After analysis of the alert, oos, and scan data, we have compiled a list of suspicious external hosts. We will now find out the registration information of these hosts to see where they are coming from. Note that while some of these IP addresses are suspects due to heightened activity on the network, the information we derive here may not necessarily be the culprit due to circumstances where the attacker spoofs their IP address. It is for this reason that an analyst should never return an attack or begin scanning machines that appear in the logs. We are simply doing registration lookups and gathering contact information. If it becomes evident that these hosts are the actual culprits to the attacks, then we can contact the administrators of the systems at hand. The hosts selected here were chosen due to either an abundance of detects or to suspicious activity that seems malicious.

The format is:
1. IP address of host to get registration information from
2. Nslookup information
3. Whois information on IP
4. Whois information on domain
(Note 1: the first lookup has the commands included, subsequent queries do not)
(Note 1a: some of the hosts that were originally selected would not resolve and were therefore omitted from the list)

205.188.246.121

```
******************************************************************

jamil@hades:~$ nslookup 205.188.246.121

Server:  hades.gaia
Address:  192.168.55.10

Name:    g2lb3.spinner.com
Address:  205.188.246.121

jamil@loki:~$ whois 205.188.246.121
```

```
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 9-Oct-2001 23:15:51 EDT.

jamil@hades:~$ whois spinner.com

Whois Server Version 1.3

   Domain Name: SPINNER.COM
   Registrar: AMERICA ONLINE, INC. DBA AOL AND/OR COMPUSERVE-AOL
   Whois Server: whois.compuserve.com
   Referral URL: http://domain.compuserve.com
   Name Server: DNS-01.SPINNER.NET
   Name Server: DNS-02.SPINNER.NET
   Updated Date: 05-jan-2000


Domain Name: SPINNER.COM

Registrant:
  Spinner Networks, Inc.
    1209 Howard Ave Suite 200
    Burlingame, CA 90410
    US

  Created on..............: Dec 23, 1999
  Expires on..............: Dec 23, 2001
  Record Last Updated on..: Jan 05, 2000
  Registrar...............: America Online, Inc.
                            http://whois.registrar.aol.com/whois/
```

```
 Administrative Contact:
  Domain Administration, Spinner
  Spinner Networks, Inc.
  1209 Howard Ave Suite 200
  Burlingame, CA 90410
  US
  Email. hostmaster@SPINNER.COM
  Tel. 415 934 2700
  Fax. 415 934 2756
 Technical Contact:
  Domain Administration, Spinner
  Spinner Networks, Inc.
  1209 Howard Ave Suite 200
  Burlingame, CA 90410
  US
  Email. hostmaster@SPINNER.COM
  Tel. 415 934 2700
  Fax. 415 934 2756
 Domain servers:
  dns-01.spinner.net
    152.163.159.239
  dns-02.spinner.net
    205.188.157.239
```

134.197.14.245

```
*********************************************************************
Server:  hades.gaia
Address:  192.168.55.10

Name:    comptech.tmcc.edu
Address:  134.197.14.245

University of Nevada, Reno (NET-UNR-DOM)
   Computing and Telecomm/MS292,1644 N.
   Virginia
   Reno, NV 89557
   US

   Netname: UNR
   Netblock: 134.197.0.0 - 134.197.255.255

   Coordinator:
      Wolff, Jeffrey  (JW270-ARIN)  dns@unr.edu
      775-784-1540x268 (FAX) 775-784-4050
```

```
  Domain System inverse mapping provided by:

  NS1.UNR.EDU               134.197.5.1
  NS2.UNR.EDU               134.197.6.1

  Record last updated on 28-Dec-1999.
  Database last updated on 9-Oct-2001 23:15:51 EDT.

Registrant:
Truckee Meadows Community College (TMCC2-DOM)
  7000 Dandini Blvd
  Reno, NV 89502
  US

  Domain Name: TMCC.EDU

  Administrative Contact, Billing Contact:
    Anderson, Cal  (CAI387)  webmaster@TMCC.EDU
    Truckee Meadows Community College
    7000 Dandini Blvd
    Reno , NV 89512
    775 673 8267
  Technical Contact:
    Dunn, Jana  (JD4959)  jana@SCS.UNR.EDU
    University and Community College System of Nevada
    Mailstop 270
    University of Nevada, Reno
    Reno, NV 89557
    (702) 784-6557

  Record last updated on 01-Nov-2000.
  Record created on 15-Jan-1997.
  Database last updated on 9-Oct-2001 22:09:00 EDT.

  Domain servers in listed order:

  TONTO.SCS.UNR.EDU         134.197.10.133
  SCOUT.SCS.UNR.EDU         134.197.212.200
```

216.169.181.240

```
***********************************************************************

Server:  hades.gaia
Address:  192.168.55.10

Name:    bvt181240.ceinetworks.com
```

```
Address:  216.169.181.240

TeleBeam, Inc. (NETBLK-TELEBM-NN0705)
  403 S. Allen St., Suite 112B
  State College, PA 16801
  US

  Netname: TELEBM-NN0705
  Netblock: 216.169.160.0 - 216.169.191.255
  Maintainer: TELB

  Coordinator:
    TeleBeam, Inc.  (TH2-ORG-ARIN)  hostmaster@TELEBEAM.NET
    +1.814.238.0000

  Domain System inverse mapping provided by:

  NS1.PLANETDIAL.COM          216.169.160.2
  NS2.PLANETDIAL.COM          216.169.160.23
  SCE.NACCESS.NET           207.112.232.2

  Record last updated on 18-Apr-2001.
  Database last updated on 9-Oct-2001 23:15:51 EDT.

  Domain Name: CEINETWORKS.COM
  Registrar: NETWORK SOLUTIONS, INC.
  Whois Server: whois.networksolutions.com
  Referral URL: http://www.networksolutions.com
  Name Server: NS1.CEINETWORKS.COM
  Name Server: NS2.CEINETWORKS.COM
  Updated Date: 17-jul-2000

Registrant:
Conestoga Enterprises Incorporated (CEINETWORKS-DOM)
  202 East First Street
  Birdsboro, PA 19508
  US

  Domain Name: CEINETWORKS.COM

  Administrative Contact, Technical Contact, Billing Contact:
    Hostmaster  (HO3961-ORG)  hostmaster@TELEBEAM.NET
    TeleBeam, Inc.
    441 Science Park Road
    State College, PA 16803
    US
```

```
     +1 814.238.0000
     Fax- +1 814.234.4821

   Record last updated on 17-Jul-2001.
   Record expires on 05-Jul-2003.
   Record created on 05-Jul-2000.
   Database last updated on 9-Oct-2001 22:09:00 EDT.

   Domain servers in listed order:

   NS1.CEINETWORKS.COM          216.169.160.2
   NS2.CEINETWORKS.COM          216.169.160.23
```

217.162.127.5

```
*********************************************************************

Server:  hades.gaia
Address:  192.168.55.10

Name:    dclient217-162-127-5.hispeed.ch
Address:  217.162.127.5

% This is the RIPE Whois server.
% The objects are in RPSL format.
% Please visit http://www.ripe.net/rpsl for more information.
% Rights restricted by copyright.
% See http://www.ripe.net/ripencc/pub-services/db/copyright.html

inetnum:      217.162.0.0 - 217.162.255.255
netname:      CH-CABLECOM-20010404
descr:        Cablecom TV Provider
descr:        Provider Local Registry
country:      CH
admin-c:      LM2274-RIPE
tech-c:       WM5132-RIPE
status:       ALLOCATED PA
mnt-by:       RIPE-NCC-HM-MNT
mnt-lower:    AS8404-MNT
changed:      <hostmaster@ripe.net> 20010404
source:       RIPE

route:        217.162.0.0/16
descr:        Cablecom GmbH
descr:        Zollstrasse42
descr:        CH-8021 Zuerich
descr:        SWITZERLAND
```

```
origin:     AS8404
remarks:      ****************************************************
remarks:      For Spam/Abuse, please contact abuse@cablecom.ch
remarks:      ****************************************************
notify:     lir-mnt@cablecom.ch
mnt-by:       AS8404-MNT
changed:      wilson.mehringer@cablecom.ch 20010817
changed:      wilson.mehringer@cablecom.ch 20010831
source:      RIPE

person:      Ludwig Molnar
address:     Cablecom GmbH
address:     Zollstr.42
address:     CH-8021 Zuerich
phone:       +41 1 277 94 07
fax-no:      +41 1 277 93 22
remarks:      ****************************************************
remarks:      For Spam/Abuse, please contact abuse@cablecom.ch
remarks:      ****************************************************
e-mail:      ludwig.molnar@cablecom.ch
nic-hdl:     LM2274-RIPE
notify:     ludwig.molnar@cablecom.ch
mnt-by:       AS8404-MNT
changed:      ludwig.molnar@cablecom.ch 20010906
source:      RIPE
person:      Wilson Mehringer
address:     Cablecom GmbH
address:     Zollstrasse 42
address:     CH-8021 Zurich
address:     Switzerland
phone:       +41 1 277 91 61
e-mail:      wilson.mehringer@cablecom.ch
nic-hdl:     WM5132-RIPE
notify:     wilson.mehringer@cablecom.ch
mnt-by:       AS8404-MNT
changed:      wilson.mehringer@cablecom.ch 20010129
changed:      wilson.mehringer@cablecom.ch 20010404
source:      RIPE
See http://www.nic.ch/terms/aup.html

Domain name:
hispeed.ch

Holder of domain name:
Cablecom Management GmbH
Domain Accounting Team
```

```
Zollstrasse 42
CH-8005 Z<FC>rich
Switzerland

Technical contact:
Cablecom Media AG
Technical Admin Team
Zollstrasse 42
CH-8005 Z<FC>rich
Switzerland

Name servers:
ns.hispeed.ch   [62.2.32.5]
ns1.cablecom.net
ns2.cablecom.net

Date of last registration:
10.04.1998

Date of last modification:
20.04.2000
```

164.107.98.247

```
*********************************************************************
Server:  hades.gaia
Address:  192.168.55.10

Name:    rnos-98-247.resnet.ohio-state.edu
Address:  164.107.98.247

Ohio State University (NET-OHIO-STATE2)
   1971 Neil Avenue  Room 480
   Columbus, OH 43210-1210
   US

   Netname: OHIO-STATE2
   Netblock: 164.107.0.0 - 164.107.255.255

   Coordinator:
      The Ohio State University  (ZT31-ARIN)  zonemaster@net.ohio-state.ed
u
      614-292-5555

   Domain System inverse mapping provided by:

   NS1.NET.OHIO-STATE.EDU       128.146.1.7
   NS2.NET.OHIO-STATE.EDU       128.146.48.7

   Record last updated on 30-May-2000.
   Database last updated on 9-Oct-2001 23:15:51 EDT.

Domain Name: OHIO-STATE.EDU
   Registrar: NETWORK SOLUTIONS, INC.
   Whois Server: whois.networksolutions.com
   Referral URL: http://www.networksolutions.com
   Name Server: NS1.NET.OHIO-STATE.EDU
   Name Server: NS2.NET.OHIO-STATE.EDU
   Updated Date: 19-aug-2001
```

Analysis of the registration data:

This data is worth noting for future reference but is not of any real importance at this stage of the analysis process. Some of the hosts that we wanted to find registration information about were not resolvable. This is always a little bit suspicious and should therefore be looked into. The hosts that were resolved seem to be coming from many different geographical regions. By doing these lookups, we can correlate what we know about geography and the signature that triggered the alert to see if the packet was possibly spoofed. For instance, if the attacker is located in China and your network is in Oklahoma City, Oklahoma, it is reasonable to assume that the TTL of the packet will have decremented substantially. If though, the attacker claims to be originating from China in this scenario and the TTL of the packet is 252, it is a pretty safe bet to assume that the packet is spoofed. Registration data is also useful as contact information in the event of a compromise and correlating data indicating a certain domain is responsible, it is trivial to get in contact with the proper individuals.

3.3.0 Day By Day Attack Statistics

Description of fields:

Date: The date the data was recorded
File: Name of file analyzed to derive results
Alerts: Total number of alerts in the file
Earliest: The time and date of the first alert in file
Latest: The time and date of the last alert in the file
Html: The name of the html file contained in the package that displays the information listed here

****

Date: **09/05/2001**
File: alert.010905.gz.txt

Alerts: 129392
Earliest: alert at **00:00:04**.599707 *on 09/05/2001*
Latest: alert at **23:55:26**.655249 *on 09/05/2001*
Html: snarf_090501

Signature (click for sig info)	# Alerts	# Sources	# Destinations
External FTP to HelpDesk MY.NET.83.197	1	1	1
Connect to 515 from inside	1	1	1
ICMP Source Quench	1	1	1
WEB-CGI calendar access	1	1	1
WEB-CGI redirect access	1	1	1
WEB-IIS Unauthorized IP Access Attempt	1	1	1
WEB-IIS view source via translate header	1	1	1
INFO Inbound GNUTella Connect request	1	1	1
INFO - Possible Squid Scan	1	1	1
ICMP Echo Request Delphi-Piette Windows	1	1	1
Virus - Possible MyRomeo Worm	1	1	1
FTP MKD . - possible warez site	1	1	1
WEB-CGI files.pl access	1	1	1
Tiny Fragments - Possible Hostile Activity	2	1	1
spp_http_decode: CGI Null Byte attack detected	2	1	1
WEB-CGI csh access	2	1	1
WEB-MISC L3retriever HTTP Probe	2	1	1
X11 outgoing	2	1	1
WEB-FRONTPAGE author.exe access	2	1	1
x86 NOOP - unicode BUFFER OVERFLOW ATTACK	2	1	1
High port 65535 udp - possible Red Worm - traffic	2	1	1
Probable NMAP fingerprint attempt	2	1	1
ICMP SRC and DST outside network	2	1	1
SMTP relaying denied	3	1	1
SCAN Synscan Portscan ID 19104	3	1	1
SYN-FIN scan!	3	1	1
INFO napster upload request	3	1	1
NMAP TCP ping!	3	1	1
ICMP Destination Unreachable (Protocol Unreachable)	4	1	1
SCAN FIN	4	1	1

EXPLOIT x86 setgid 0	5	1	1
WEB-FRONTPAGE fourdots request	5	1	1
INFO Outbound GNUTella Connect request	5	1	1
INFO - Web Cmd completed	5	1	1
WEB-IIS _vti_inf access	7	1	1
SUNRPC highport access!	8	1	1
MISC Large ICMP Packet	8	1	1
CS WEBSERVER - external ftp traffic	10	1	1
WEB-MISC count.cgi access	10	1	1
WEB-FRONTPAGE fpcount.exe access	11	1	1
WEB-FRONTPAGE _vti_rpc access	12	1	1
External RPC call	12	1	1
beetle.ucs	12	1	1
EXPLOIT x86 NOOP	14	1	1
ICMP Echo Request BSDtype	16	1	1
Port 55850 tcp - Possible myserver activity - ref. 010313-1	16	1	1
EXPLOIT x86 setuid 0	16	1	1
SMB Name Wildcard	17	1	1
spp_http_decode: IIS Unicode attack detected	18	1	1
INFO FTP anonymous FTP	22	1	1
WEB-MISC http directory traversal	22	1	1
High port 65535 tcp - possible Red Worm - traffic	23	1	1
TELNET login incorrect	23	1	1
Queso fingerprint	27	1	1
WEB-MISC 403 Forbidden	50	1	1
SCAN Proxy attempt	65	1	1
ICMP Echo Request Sun Solaris	77	1	1
ICMP Echo Request L3retriever Ping	78	1	1
INFO Outbound GNUTella Connect accept	121	1	1
ICMP Echo Request Windows	146	1	1
FTP DoS ftpd globbing	149	1	1
ICMP Echo Request CyberKit 2.2 Windows	155	1	1
INFO Napster Client Data	191	1	1
TCP SRC and DST outside network	234	74	154
ICMP traceroute	247	1	1
Incomplete Packet Fragments Discarded	284	1	1
ICMP Fragment Reassembly Time Exceeded	285	1	1

ICMP Destination Unreachable (Host Unreachable)	321	1	1
INFO Inbound GNUTella Connect accept	356	1	1
INFO Possible IRC Access	546	1	1
ICMP Destination Unreachable (Network Unreachable)	556	1	1
Watchlist 000222 NET-NCFC	847	1	1
Null scan!	867	1	1
INFO napster login	954	1	1
CS WEBSERVER - external web traffic	1789	1	1
WEB-MISC prefix-get //	2061	1	1
Possible trojan server activity	2188	1	1
MISC traceroute	2189	1	1
Watchlist 000220 IL-ISDNNET-990517	2875	1	1
INFO MSN IM Chat data	2904	1	1
ICMP Echo Request Nmap or HPING2	3043	1	1
ICMP Destination Unreachable (Communication Administratively Prohibited)	3902	1	1
MISC source port 53 to <1024	4577	1	1
MISC Large UDP Packet	14260	1	1
IDS552/web-iis_IIS ISAPI Overflow ida nosize	38676	1	1
WEB-MISC Attempt to execute cmd	44019	1	1

****

Date: **09/06/2001**
File: alert.010906.gz.txt
Alerts: 113061 found
Earliest: alert at **00:00:01**.474597 *on 09/06/2001*
Latest: alert at **17:29:59**.201377 *on 09/06/2001* ***Note the time of the final alert!**
Html: snarf_090601

Signature (click for sig info)	# Alerts	# Sources	# Destinations
WEB-CGI calendar access	1	1	1
SCAN Synscan Portscan ID 19104	1	1	1
WEB-CGI redirect access	1	1	1
Virus - Possible scr Worm	1	1	1
INFO napster new user login	1	1	1
ICMP Redirect (Undefined Code!)	1	1	1
WEB-MISC whisker head	1	1	1

Virus - Possible pif Worm	1	1	1
EXPLOIT identd overflow	1	1	1
RFB - Possible WinVNC - 010708-1	1	1	1
WEB-CGI archie access	1	1	1
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	1	1	1
ICMP Mobile Registration Reply (Undefined Code!)	1	1	1
WEB-COLDFUSION administrator access	1	1	1
INFO Inbound GNUTella Connect request	1	1	1
RPC tcp traffic contains bin_sh	1	1	1
TELNET access	2	1	1
Port 55850 udp - Possible myserver activity - ref. 010313-1	2	1	1
WEB-CGI ksh access	2	1	1
ICMP Destination Unreachable (Fragmentation Needed and DF bit was set)	2	1	1
INFO Outbound GNUTella Connect request	2	1	1
WEB-FRONTPAGE author.exe access	2	1	1
BACKDOOR NetMetro Incoming Traffic	2	1	1
SCAN XMAS	2	1	1
X11 outgoing	3	1	1
Connect to 515 from inside	3	1	1
INFO - Web Cmd completed	3	1	1
WEB-IIS view source via translate header	3	1	1
Virus - Possible MyRomeo Worm	3	1	1
WEB-FRONTPAGE fourdots request	3	1	1
ICMP Destination Unreachable (Protocol Unreachable)	3	1	1
NMAP TCP ping!	4	1	1
INFO - Possible Squid Scan	4	1	1
SMTP relaying denied	4	1	1
INFO napster upload request	4	1	1
EXPLOIT x86 setgid 0	4	1	1
EXPLOIT x86 stealth noop	4	1	1
WEB-CGI scriptalias access	5	1	1
WEB-MISC count.cgi access	6	1	1
Port 55850 tcp - Possible myserver activity - ref. 010313-1	7	1	1
SCAN FIN	7	1	1

CS WEBSERVER - external ftp traffic	7	1	1
EXPLOIT x86 setuid 0	7	1	1
WEB-FRONTPAGE fpcount.exe access	8	1	1
INFO FTP anonymous FTP	8	1	1
Probable NMAP fingerprint attempt	8	1	1
High port 65535 udp - possible Red Worm - traffic	8	1	1
Tiny Fragments - Possible Hostile Activity	9	1	1
WEB-IIS _vti_inf access	9	1	1
SMB Name Wildcard	9	1	1
MISC Large ICMP Packet	12	1	1
ICMP SRC and DST outside network	12	3	4
External RPC call	13	1	1
High port 65535 tcp - possible Red Worm - traffic	15	1	1
TELNET login incorrect	15	1	1
WEB-FRONTPAGE _vti_rpc access	15	1	1
spp_http_decode: IIS Unicode attack detected	17	1	1
beetle.ucs	18	1	1
ICMP Echo Request Windows	21	1	1
ICMP Echo Request Sun Solaris	29	1	1
WEB-MISC 403 Forbidden	29	1	1
Queso fingerprint	29	1	1
WEB-MISC http directory traversal	31	1	1
ICMP Echo Request CyberKit 2.2 Windows	37	1	1
Incomplete Packet Fragments Discarded	41	1	1
x86 NOOP - unicode BUFFER OVERFLOW ATTACK	47	1	1
INFO Outbound GNUTella Connect accept	68	1	1
ICMP Fragment Reassembly Time Exceeded	79	1	1
FTP DoS ftpd globbing	82	1	1
EXPLOIT x86 NOOP	102	1	1
ICMP Echo Request L3retriever Ping	109	1	1
SCAN Proxy attempt	120	1	1
INFO Inbound GNUTella Connect accept	128	1	1
TCP SRC and DST outside network	142	37	97
ICMP traceroute	143	1	1
Watchlist 000222 NET-NCFC	153	1	1
INFO Napster Client Data	166	1	1
ICMP Destination Unreachable (Host Unreachable)	210	1	1

INFO Possible IRC Access	220	1	1
ICMP Destination Unreachable (Network Unreachable)	457	1	1
INFO napster login	1147	1	1
CS WEBSERVER - external web traffic	1232	1	1
WEB-MISC prefix-get //	1461	1	1
ICMP Echo Request BSDtype	1565	1	1
MISC Large UDP Packet	1622	1	1
MISC traceroute	1655	1	1
ICMP Echo Request Nmap or HPING2	1661	1	1
INFO MSN IM Chat data	1759	1	1
MISC source port 53 to <1024	2158	1	1
ICMP Destination Unreachable (Communication Administratively Prohibited)	2944	1	1
Null scan!	3076	1	1
Watchlist 000220 IL-ISDNNET-990517	24472	1	1
IDS552/web-iis_IIS ISAPI Overflow ida nosize	30330	1	1
WEB-MISC Attempt to execute cmd	35244	1	1

****

Date: **09/07/2001**
File: alert.010907.gz.txt
Alerts: 3206 found
Earliest: alert at **00:01:39**.371459 *on 09/07/2001*
Latest: alert at **23:47:28**.014322 *on 09/07/2001*
Html: snarf_090701

Signature	# Alerts	# Sources	# Destinations
Connect to 515 from inside	1	1	1
ICMP SRC and DST outside network	1	1	1
Tiny Fragments - Possible Hostile Activity	2	1	1
NMAP TCP ping!	2	1	1
Possible trojan server activity	3	1	1
High port 65535 udp - possible Red Worm - traffic	5	1	1
Null scan!	5	1	1
WinGate 1080 Attempt	8	1	1
High port 65535 tcp - possible Red Worm - traffic	14	1	1
External RPC call	18	1	1
Queso fingerprint	27	1	1

SMB Name Wildcard	30	1	1
Watchlist 000222 NET-NCFC	39	1	1
TCP SRC and DST outside network	42	13	36
Russia Dynamo - SANS Flash 28-jul-00	79	1	1
Watchlist 000220 IL-ISDNNET-990517	117	1	1
Port 55850 tcp - Possible myserver activity - ref. 010313-1	1276	1	1
UDP SRC and DST outside network	1537	30	196

****

Date: **09/08/2001**
File: alert.010908.gz.txt
Alerts: 51907 found
Earliest: alert at **11:45:03**.153486 *on 09/08/2001* ***Note the time of the first alert!**
Latest: alert at **23:55:21**.804415 *on 09/08/2001*
Html: snarf_090801

Signature (click for sig info)	# Alerts	# Sources	# Destinations
EXPLOIT x86 setgid 0	1	1	1
WEB-CGI scriptalias access	1	1	1
WEB-CGI redirect access	1	1	1
Virus - Possible MyRomeo Worm	1	1	1
SNMP public access	1	1	1
WEB-MISC L3retriever HTTP Probe	1	1	1
DNS zone transfer	1	1	1
IDS50/trojan_trojan-active-subseven	1	1	1
TELNET access	1	1	1
Virus - Possible scr Worm	1	1	1
WEB-IIS scripts-browse	1	1	1
ICMP SRC and DST outside network	2	2	2
WEB-MISC http directory traversal	2	1	1
WEB-IIS _vti_inf access	2	1	1
INFO - Possible Squid Scan	2	1	1
Port 55850 tcp - Possible myserver activity - ref. 010313-1	2	1	1
WEB-CGI rsh access	2	1	1
Port 55850 udp - Possible myserver activity - ref. 010313-1	2	1	1

ICMP Source Quench	3	1	1
FTP CWD / - possible warez site	3	1	1
NMAP TCP ping!	3	1	1
spp_http_decode: IIS Unicode attack detected	3	1	1
ICMP Echo Request L3retriever Ping	4	1	1
INFO FTP anonymous FTP	4	1	1
Tiny Fragments - Possible Hostile Activity	4	1	1
EXPLOIT x86 stealth noop	5	1	1
WEB-FRONTPAGE fpcount.exe access	5	1	1
EXPLOIT x86 setuid 0	5	1	1
beetle.ucs	5	1	1
WEB-FRONTPAGE _vti_rpc access	6	1	1
BACKDOOR NetMetro File List	6	1	1
Possible trojan server activity	6	1	1
WEB-MISC count.cgi access	6	1	1
ICMP Echo Request BSDtype	7	1	1
High port 65535 udp - possible Red Worm - traffic	7	1	1
Russia Dynamo - SANS Flash 28-jul-00	8	1	1
MISC Large ICMP Packet	8	1	1
SMB Name Wildcard	8	1	1
Queso fingerprint	9	1	1
ICMP Destination Unreachable (Protocol Unreachable)	9	1	1
TELNET login incorrect	14	1	1
INFO Possible IRC Access	14	1	1
ICMP Echo Request Windows	17	1	1
Watchlist 000222 NET-NCFC	19	1	1
x86 NOOP - unicode BUFFER OVERFLOW ATTACK	20	1	1
External RPC call	23	1	1
INFO Outbound GNUTella Connect accept	27	1	1
WEB-MISC 403 Forbidden	29	1	1
High port 65535 tcp - possible Red Worm - traffic	29	1	1
FTP DoS ftpd globbing	37	1	1
ICMP Echo Request Sun Solaris	62	1	1
ICMP Echo Request CyberKit 2.2 Windows	67	1	1
ICMP Fragment Reassembly Time Exceeded	91	1	1
TCP SRC and DST outside network	95	33	70

ICMP Destination Unreachable (Host Unreachable)	99	1	1
INFO Inbound GNUTella Connect accept	101	1	1
EXPLOIT x86 NOOP	104	1	1
SCAN Proxy attempt	106	1	1
TFTP - Internal TCP connection to external tftp server	116	1	1
INFO Napster Client Data	139	1	1
Null scan!	142	1	1
Watchlist 000220 IL-ISDNNET-990517	161	1	1
ICMP traceroute	165	1	1
INFO napster login	216	1	1
ICMP Destination Unreachable (Network Unreachable)	249	1	1
Incomplete Packet Fragments Discarded	312	1	1
SMTP relaying denied	364	1	1
CS WEBSERVER - external web traffic	688	1	1
WEB-MISC prefix-get //	866	1	1
MISC source port 53 to <1024	961	1	1
ICMP Echo Request Nmap or HPING2	980	1	1
MISC traceroute	1035	1	1
INFO MSN IM Chat data	1362	1	1
ICMP Destination Unreachable (Communication Administratively Prohibited)	1763	1	1
MISC Large UDP Packet	4324	1	1
IDS552/web-iis_IIS ISAPI Overflow ida nosize	17016	1	1
WEB-MISC Attempt to execute cmd	19945	1	1

****

Date: **09/09/2001**
File: alert.010909.gz.txt
Alerts: 120879 found
Earliest: alert at **00:00:02**.216675 *on 09/09/2001*
Latest: alert at **23:56:09**.920166 *on 09/09/2001*
Html: snarf_090901

Signature (click for sig info)	# Alerts	# Sources	# Destinations
WEB-MISC Lotus Domino directory traversal	1	1	1
WEB-CGI w3-msql access	1	1	1

ICMP Echo Request L3retriever Ping	1	1	1
BACKDOOR NetMetro File List	1	1	1
SNMP public access	1	1	1
Virus - Possible pif Worm	1	1	1
WEB-MISC compaq nsight directory traversal	1	1	1
IDS50/trojan_trojan-active-subseven	1	1	1
WEB-CGI ksh access	1	1	1
EXPLOIT FTP passwd appe path	1	1	1
RFB - Possible WinVNC - 010708-1	1	1	1
RPC tcp traffic contains bin_sh	1	1	1
spp_http_decode: CGI Null Byte attack detected	1	1	1
Port 55850 udp - Possible myserver activity - ref. 010313-1	1	1	1
Back Orifice	1	1	1
SCAN FIN	1	1	1
X11 outgoing	1	1	1
INFO - Web File Copied ok	1	1	1
INFO Inbound GNUTella Connect request	1	1	1
Tiny Fragments - Possible Hostile Activity	2	1	1
WEB-CGI redirect access	2	1	1
ICMP Timestamp Reply	2	1	1
WEB-IIS view source via translate header	2	1	1
INFO - Web Cmd completed	2	1	1
x86 NOOP - unicode BUFFER OVERFLOW ATTACK	2	1	1
WEB-IIS scripts-browse	2	1	1
EXPLOIT x86 setgid 0	2	1	1
WEB-CGI upload.pl access	3	1	1
WEB-FRONTPAGE fpcount.exe access	3	1	1
INFO napster upload request	3	1	1
WEB-CGI rsh access	3	1	1
WEB-FRONTPAGE fourdots request	4	1	1
BACKDOOR NetMetro Incoming Traffic	4	1	1
WEB-CGI files.pl access	4	1	1
WEB-IIS _vti_inf access	4	1	1
WEB-MISC whisker head	4	1	1
NMAP TCP ping!	5	1	1
Port 55850 tcp - Possible myserver activity - ref.	5	1	1

010313-1			
WEB-MISC L3retriever HTTP Probe	5	1	1
beetle.ucs	6	1	1
ICMP Echo Request BSDtype	7	1	1
WEB-MISC count.cgi access	8	1	1
ICMP Echo Request Delphi-Piette Windows	9	1	1
Connect to 515 from inside	9	1	1
EXPLOIT x86 setuid 0	11	1	1
MISC Large ICMP Packet	14	1	1
Queso fingerprint	16	1	1
WEB-FRONTPAGE _vti_rpc access	16	1	1
CS WEBSERVER - external ftp traffic	16	1	1
INFO FTP anonymous FTP	16	1	1
WEB-MISC http directory traversal	17	1	1
SMB Name Wildcard	19	1	1
High port 65535 udp - possible Red Worm - traffic	19	1	1
ICMP Destination Unreachable (Protocol Unreachable)	19	1	1
spp_http_decode: IIS Unicode attack detected	20	1	1
TELNET login incorrect	25	1	1
INFO Possible IRC Access	28	1	1
ICMP Echo Request CyberKit 2.2 Windows	32	1	1
Null scan!	50	1	1
ICMP Echo Request Windows	61	1	1
WEB-MISC 403 Forbidden	66	1	1
FTP DoS ftpd globbing	71	1	1
Watchlist 000222 NET-NCFC	71	1	1
INFO Outbound GNUTella Connect accept	73	1	1
SCAN Proxy attempt	97	1	1
INFO Napster Client Data	101	1	1
TCP SRC and DST outside network	106	50	79
ICMP Fragment Reassembly Time Exceeded	126	1	1
EXPLOIT x86 NOOP	143	1	1
TFTP - Internal TCP connection to external tftp server	171	1	1
External RPC call	176	1	1
ICMP Echo Request Sun Solaris	186	1	1
ICMP Source Quench	219	1	1

INFO Inbound GNUTella Connect accept	260	1	1
ICMP traceroute	269	1	1
ICMP Destination Unreachable (Host Unreachable)	478	1	1
Incomplete Packet Fragments Discarded	515	1	1
INFO napster login	519	1	1
ICMP Destination Unreachable (Network Unreachable)	599	1	1
Watchlist 000220 IL-ISDNNET-990517	654	1	1
SMTP relaying denied	908	1	1
Possible trojan server activity	992	1	1
High port 65535 tcp - possible Red Worm - traffic	1046	1	1
CS WEBSERVER - external web traffic	1540	1	1
WEB-MISC prefix-get //	1644	1	1
MISC source port 53 to <1024	1912	1	1
ICMP Echo Request Nmap or HPING2	1923	1	1
INFO MSN IM Chat data	1930	1	1
MISC traceroute	2413	1	1
ICMP Destination Unreachable (Communication Administratively Prohibited)	2595	1	1
WEB-IIS 5 Printer-beavuh	3007	1	1
MISC Large UDP Packet	11047	1	1
IDS552/web-iis_IIS ISAPI Overflow ida nosize	41076	1	1
WEB-MISC Attempt to execute cmd	43446	1	1

3.4.0 Brief Overview of Alerts

There are 93 unique attacks that were attempted within or against this network. The following is a list of each attack with a general description to follow. Not all 93 signatures are listed because after a thorough analysis of the alerts, it was found that many signatures were actually replicates. Some of the replicates noted were 3 x86 NOOP signatures, 2 NULL scan signatures, 2 Napster signatures, 4 Red Worm > 60000 port activity signatures, 2 traceroute signatures, among others. All signatures that had more than one name but was triggered due to the same malicious activity were deleted from the chart. Section 3.2.0 analyzes five of these signatures in extreme detail.

Name of alert	Description of alert
WEB-MISC Lotus Domino directory traversal	Lotus Domino is a multiplatform web server which integrates messaging and various interactive web applications. It is possible for a remote user to gain access to any known file residing on the Lotus Domino Server 5.0.6 and previous. A

	specially crafted HTTP request comprised of '.nsf' and '../' along with the known filename, will display the contents of the particular file with read permissions. It should be noted that when making this malformed request Internet Explorer removes '.nsf' portion of the URL, obstructing the exploitation of this vulnerability. Successful exploitation of this vulnerability could enable a remote user to gain access to systems files, password files, etc. This could lead to a complete compromise of the host.
WEB-CGI w3-msql access	Under certain versions of Mini SQL, the w3-msql CGI script allows users to view directories which are set for private access via .htaccess files. W3-mSQL converts any form data passed to a script into global Lite variables and these variables can then be accessed by your script code. When an HTML form is defined a field name is given to each element of the form. When the data is passed to W3-mSQL the field names are used as the variable names for the global variables. Once a set of variables has been created for each form element, the values being passed to the script are assigned to the variables. This is done automatically during start-up of the W3-mSQL program.
SNMP public access	Insufficient access control, and allow reading/writing of MIB data with any community password
Virus - Possible pif Worm	Numerous worms use this .pif extension including Sircam.
WEB-MISC Compaq buffer overflow vulnerability	The administration tool is vulnerable to buffer overflow attack techniques employing maliciously-formed user-supplied input. Properly exploited, this vulnerability can allow a remote attacker to execute arbitrary code on the affected system, with the privilege level of the system administrator.
IDS50/trojan_trojan-active-subseven	SubSeven is a trojan for the windows platform. It comes at least in two parts a client and a server. The client is used by the hacker to connect to the victim' s machine. Once the server.exe is installed on the victim's machine the hacker has full access to the victim's machine. http://www.sans.org/newlook/resources/IDFAQ/subseven.htm
WEB-CGI ksh access	Korn shell access, this may or may not be malicious.
RFB - Possible WinVNC - 010708-1	There may be a VNC server or client on the network. WinVNC has multiple exploits associated with it

RPC tcp traffic contains bin_sh	Bin_sh is string which is common to many exploits, it is usually an attempt for an intruder to get a shell
spp_http_decode: CGI Null Byte attack detected	If the http decoding routine finds a %00 in an http request, it will alert with this message. Sometimes you may see false positives with sites that use cookies with URL encoded binary data, or if you're scanning port 443 and picking up SSL encrypted traffic.
Back Orifice	Back Orifice is not a virus. It is in essence a ***remote administration tool***. It gives "system admin" type privileges to a remote user by way of the computer's Internet link. http://www.nwinternet.com/~pchelp/bo/bo.html
SCAN FIN	Portscan that sets only the TCP FIN flag. This scan can produce varied results based on the type of operating system the victim is utilizing/
X11 outgoing	Client inside the network, server (display) outside. This could be an exploited box that is serving an X terminal to an attacker.
INFO Inbound GNUTella Connect request	The mp3 service is requesting a connection. There may be a system with GNUtella on the network.
Tiny Fragments - Possible Hostile Activity	Many small fragments. This is a method that attackers use to bypass firewalls as well as confuse certain TCP/IP stacks.
WEB-CGI redirect access	CGI script that will redirect browsers to a new URL. It can display a page telling the user they are about to be redirected as well as log the redirect. This program can be used maliciously to redirect users to a specific web location.
ICMP Timestamp Reply	The data received (a timestamp) in the message is returned in the reply together with an additional timestamp. The timestamp is 32 bits of milliseconds since midnight UT. The ID and Seq # fields returned to the sender should be unaltered. Type: 14 Code: 0
WEB-IIS view source via translate header	It is possible to force the IIS server to send back the source of known scriptable files to the client if the HTTP GET request contains a specialized header with 'Translate: f' at the end of it, and if a trailing slash '/' is appended to the end of the URL. The scripting engine will be able to locate the requested file, however, it will not recognize it as a file that needs to be processed and will proceed to send the file source to the client.
x86 NOOP - unicode BUFFER OVERFLOW ATTACK	NOOP's are common to find in the payload of packets that are attempting to perform a buffer overflow. In this case, they are masked in Unicode. This is a fairly reliable signature which is rarely a false positive.
WEB-IIS scripts-browse	IIS may return content specified by a malicious third party back to a client through the use of specially formed links.

EXPLOIT x86 setgid 0	Exploit that attempts to utilize the root group id (0).
WEB-CGI upload.pl access	CGI script which allows the uploading of files. This perl script allows directory traversal.
WEB-FRONTPAGE fpcount.exe access	Fpcount.exe is an exploitable program that should not be used. This may be a false positive because the signature simply looks for the “fpcount.exe” string in the payload of the packet.
INFO napster upload request	Napster is an mp3 file trading service.
WEB-CGI rsh access	Rsh is a service which does password authentication through plain text. All passwords are visible to anyone who is sniffing on the network. Highly advisable to disable this service.
WEB-CGI files.pl access	The toolkit contained a script called "FILES.PL" that could be used to view the contents of files or directories on the server by a remote attacker. This is done by passing the parameter "file=<file-or-directory-to-view>" to the script. An attacker could gain information useful in conducting subsequent attacks, or retrieve personal or proprietary information.
WEB-MISC whisker head	Web scanner that has many anti-IDS features. This signature means that this scanner may have been used against the network.
NMAP TCP ping!	Nmap is a popular port scanner and this is one of the methods that it can scan. It is a simple TCP ping. (as opposed to the usual ICMP echo request/reply).
WEB-MISC L3retriever HTTP Probe	Scanner that probes web servers. It may have been used against the network.
beetle.ucs	A CD burning web site? Unknown.
ICMP Echo Request BSDtype	A BSD O.S. Echo Request. Arachnids 152. Type: 8 Code: 0
WEB-MISC count.cgi access	Wwwcount (count.cgi) is a very popular CGI program used to track website usage. In particular, it enumerates the number of hits on given webpages and increments them on a 'counter'. In October of 1997 two remotely exploitable problems were discovered with this program. The first problem was somewhat innocuous in that it only allowed remote users to view .GIF files they were not supposed to have access to. This may be dangerous if the site contains sensitive data in .GIF files such as demographic/financial data in charts etc. The second and most serious problem is a buffer overflow in QUERY_STRING environment variable handled by the program. In essence a remote user can send an overly long query to the program and overflow a buffer in order to execute their own commands as whatever privilege level the program is running as.

Connect to 515 from inside	This is a connection to the LPD service from within the network.
MISC Large ICMP Packet	This signature can be cause by a variety of sources. It is primarily triggered upon an MTU discovery attempt.
Queso fingerprint	Queso is an operating system fingerprint program. This means that this program was detected while it scanned a host on the network.
INFO FTP anonymous FTP	There was an anonymous login to an FTP server. This signature is informative more than it shows a specific hacking attempt. This should only cause alarm is there should not be an anonymous FTP server on the network.
WEB-MISC http directory traversal	A web server has been used to traverse the hosts' directories. This may or may not be an incident to investigate.
High port 65535 udp - possible Red Worm – traffic	The "Code Red" worm is self-replicating malicious code that exploits a known vulnerability in Microsoft IIS servers (CA-2001-13).
ICMP Destination Unreachable (Protocol Unreachable)	This is an administrative alert. It may or may not be worthy of investigation. Type: 3 Code: 2
spp_http_decode: IIS Unicode attack detected	A flaw exists in the handling of .asp requests. Typically when a request is made for an .asp file, IIS will identify that it is a script and run it as such. However if the host is formatted with a FAT file system and a request is made with an .asp Unicode encoded file extension, IIS may not handle the request properly and return the source code of the file.
TELNET login incorrect	Multiple telnet incorrect logins could mean that an intruder is attempting to brute force the password of an account on a telnet server.
INFO Possible IRC Access	Internet Relay Chat program access. This means that IRC has been detected, merely informational.
Null scan!	This is a TCP scan that has no flags set. This can cause some TCP/IP stack implementations to disclose information about open ports on the system.
ICMP Echo Request Windows	A ping from a Windows machine.
WEB-MISC 403 Forbidden	Attempt to access a web page that is "Forbidden" by the administrator. This may or may not be harmful. It may be accidental.
FTP DoS ftpd globbing	Globbing generates pathnames from file name patterns used by the shell, eg. wildcards denoted by * and ?, multiple choices denoted by {}, etc.

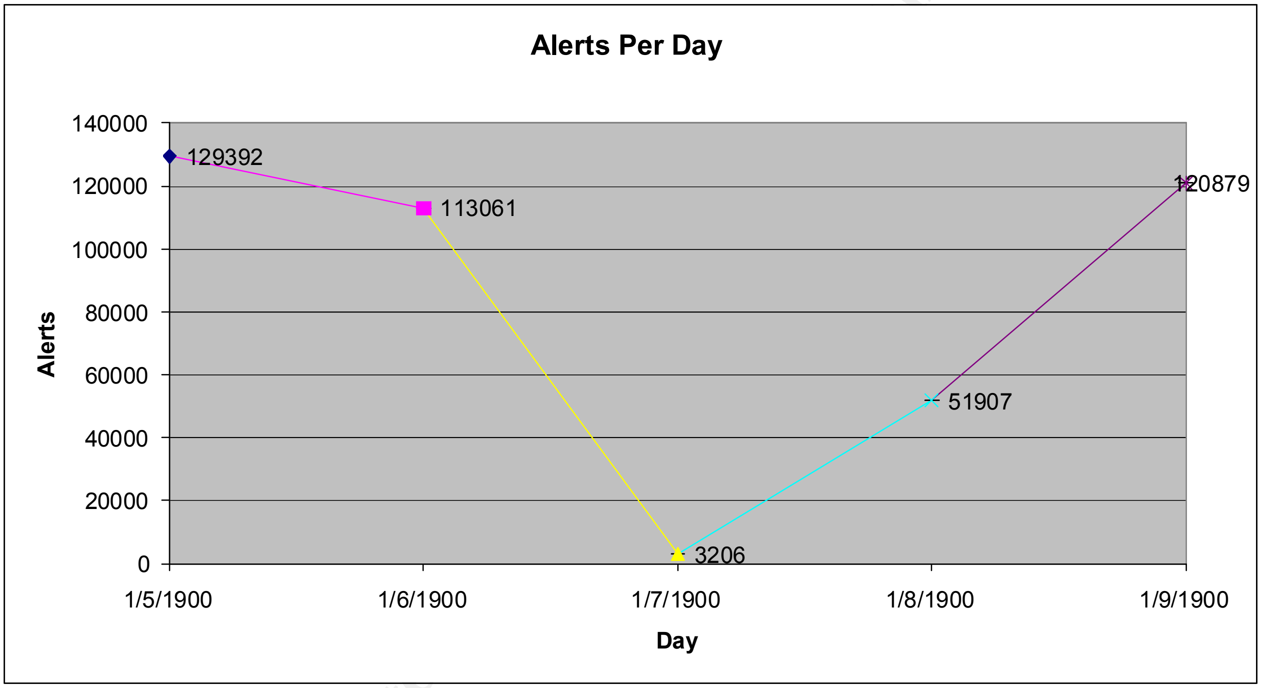
	The vulnerable FTP servers can be exploited to exhaust system resources if per-user resource usage controls have not been implemented.
INFO Outbound GNUTella Connect accept	Another GNUtella signature, this time it is for an outbound request.
INFO Napster Client Data	Another Napster signature, this time it is for a Napster client.
TCP SRC and DST outside network	The IDS should only capture data that is coming to or from the local network. If data is neither originating nor destined for the local network, the data must be spoofed, which is a tell-tale sign of malicious activity or a misconfigured router.
ICMP Fragment Reassembly Time Exceeded	This may be informational and not malicious. There is a 60 second grace period for fragment reassembly and this alerts us to that scenario. Type: 11 Code: 1
External RPC call	Attempted use of an RPC service from a remote location. The RPC services are listed as one of the top 10 SANS most vulnerable services.
ICMP Echo Request Sun Solaris	Sun specific ping. This is an informational alert. Type: 8 Code: 0
ICMP Source Quench	This alert is triggered when one of the hosts in a connection cannot handle the amount of data being sent to it from another host. A source quench tells the offending host to reduce the amount of traffic it is sending. Type: 4 Code: 0
ICMP traceroute	Informational. Traceroute traces the route a packet will take to a particular destination. Traceroute will initially send a packet with a TTL of 1 to the ultimate destination and await an error response from the host the packet timed out at. It will continue to increment the TTL by 1 until it finally reaches the destination host.
ICMP Destination Unreachable (Host Unreachable)	Informational. ICMP error message saying that the destination cannot be reached. Type: 3 Code: 1
ICMP Destination Unreachable (Network Unreachable)	Informational. The router cannot reach the desired network. Type: 3 Code: 0
SMTP relaying denied	An attempt to relay mail from an SMTP server failed and the server replied with this message. This is usually a good message because open mail relaying will lead to the blacklisting of the particular SMTP server.
WEB-MISC prefix-get //	This string is associated with numerous exploits including:

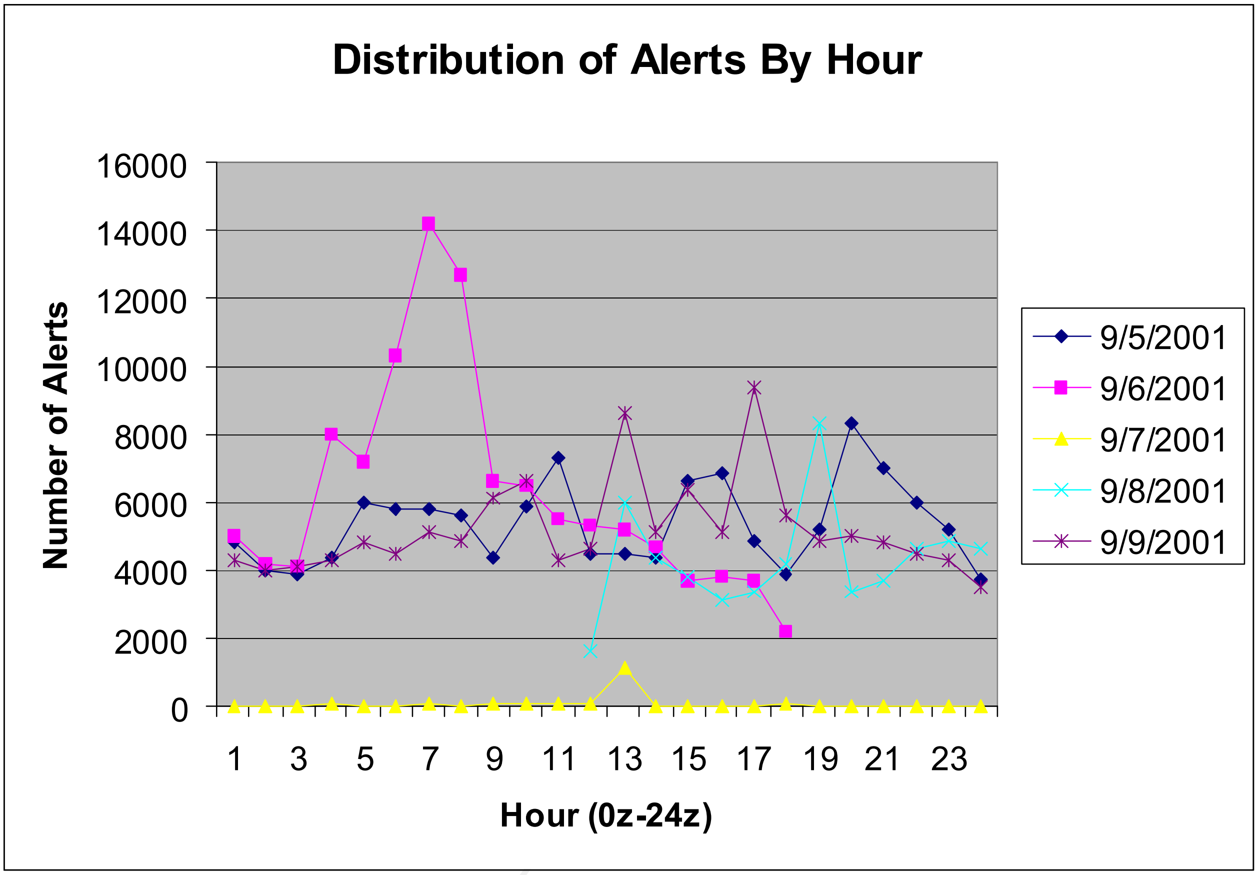
	ht://dig Remote Command Execution Vulnerability ht://dig Arbitrary File Inclusion Vulnerability AOL Instant Messenger 'aim://' Buffer Overflow Vulnerability Trend Micro Interscan Applet Trap '//' Bypass Vulnerability
MISC source port 53 to <1024	Port 53 is the reserved address for nameserver activity. A DNS server should send data (source) on port 53 to a port above 1024 (destination). Sometimes on older BIND implementations, both the source and destination are 53 and therefore this leads to a false positive.
ICMP Destination Unreachable (Communication Administratively Prohibited)	Informational. This is an alert that is generated when the network has specific restrictions on the traffic. This ICMP message is returned to a host who attempts to direct traffic to a restricted location. RFC 1812: http://sunsite.dk/RFC/rfc/rfc1812.html Type: 3 Code: 13
MISC Large UDP Packet	This could be a sign of a UDP flood. If many large UDP packets are sent to a host it can cause a DOS. Another possibility is that the UDP session is actually a covert channel used by an attacker to communicate with a compromised host. This warrants investigation.

3.5.0 Graphing Trends

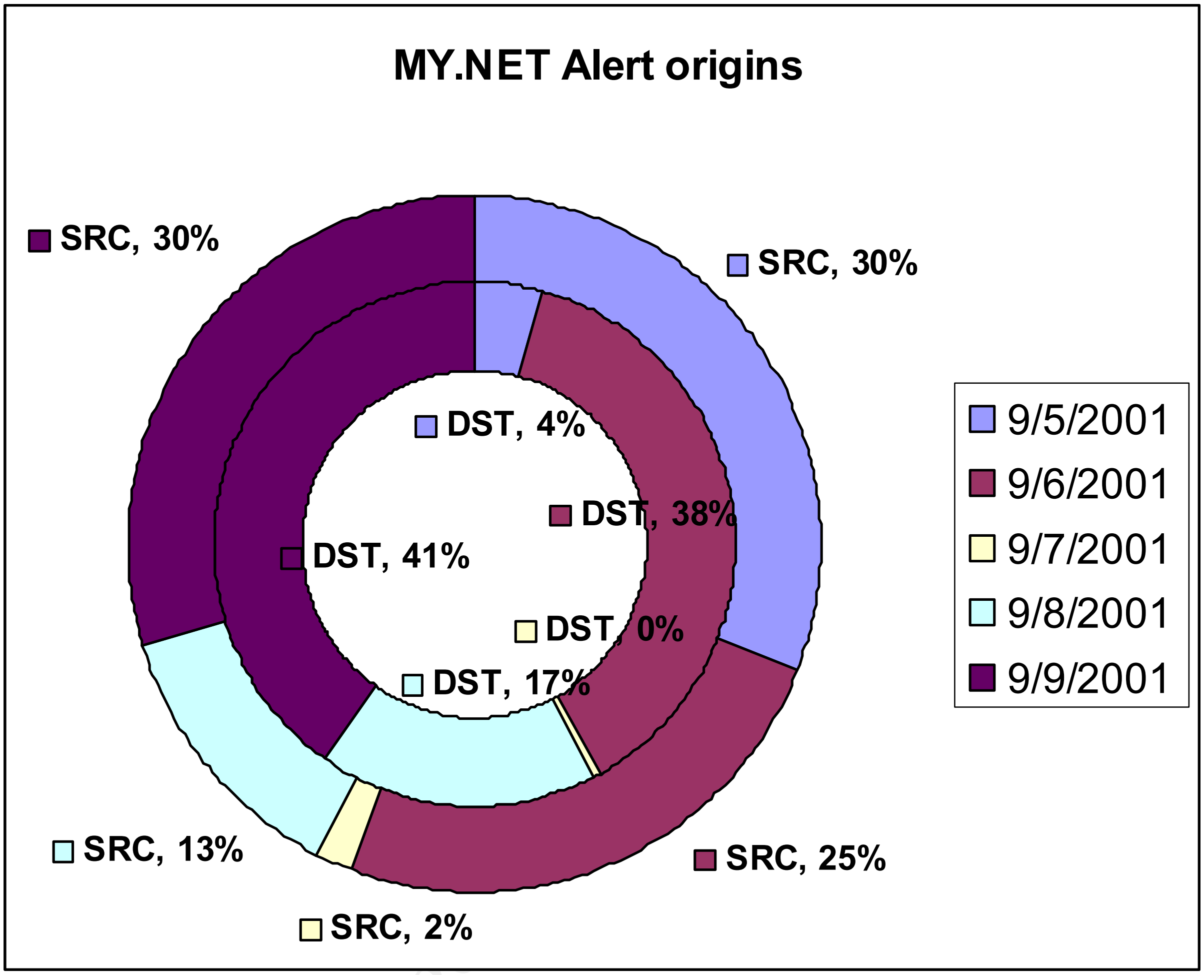
Noting the number of alerts per day is a significant advantage because it shows us which days are more prone to attack. By isolating certain dates, we can further analyze the data by pinpointing certain time frames that excess activity was witnessed.

NOTE: If viewing this on a computer, sometimes you must wait a little while for the image to appear – it seems Word has issues displaying graphs. There are three graphs below.





The graph displaying distribution of alerts by hour is beneficial in that it helps us notice what periods of time the network saw heightened malicious activity. We can gather a number of beneficial artifacts from this graph. For one, we note the immense activity from 6 am until around 830 am on the morning of 9/6/2001. The logs from this time period should be examined further. We also note that the data from the 6th stops at 1730z. While we do not know the source of this problem, we can graphically see it depicted here. We can see though, that the trend at that time was a relatively dramatic decrease in network alerts. Another data outage is also easily identifiable when looking at this chart; the morning of the 8th, from 0z to around 12z noon there is not data available. A general look at the entire graph shows many trends which we should use to develop a "norm" for the traffic. There is a general decrease in alerts from about 20z to 330z. The majority of the alerts we are seeing are from about 9z through 20z. The peak alert period was 7z, 9/6/2001; the trough alert period for recorded data was 2z 9/7/2001. In the future, it would be extremely beneficial to continue to develop an hour by hour alert graph like this to ease the process of analyzing large amounts of data.



This graph represents the origin of the alerts detected in relation to the MY.NET network (the University's network). All references to "SRC" indicate that the MY.NET network was the source of the alert. All "DST" references (inner circle) indicate that the attack originated from an external network and was destined for the MY.NET network. By looking at this graph, we can infer that on the 5th and 9th there were numerous outbound packets that triggered alerts. Also note that for the majority of dates, the ratio of inbound/outbound attacks is relatively even except on the 6th 152% difference (13 percentage points) and the 5th with 750% difference (26 percentage points).

==

3.6.0 Analysis of Alerts

There are a total of 93 unique alert types for this five day time span. Each of these alerts was triggered up to 44019 times per day. This information can now be correlated with specific IP addresses and activities to determine what activity is legitimate and which activity is illicit. Keep in mind that many of these alerts can and may be false positives due to the general nature of the signature which triggers the alert.

This section will thoroughly document and describe the results of the analyzed logs. The format will be like so:

● **Name**: The name of the alert
● **Occurrences:** Total number of detects and a day by day breakdown of individual occurrences.
● **Description:** A description of the attack.
● **Correlations:** Documentation related to the aforementioned attack.
● **Countermeasures:** Methods to defend against the attack.
● **Action:** The reason this alert was selected for further review and a look at the direct impact of the attack upon the University's network.

These are the top 5 alerts/scans/out of spec traces that have been selected due to sheer number of occurrences, severity of the attack, or uniqueness. For a general description of each of the 93 unique attacks, refer to section 3.1.5.

Name	WEB-MISC Attempt to execute cmd
Occurrences	142654 total Date -- Alert 9/5/01 -- 44019 9/6/01 -- 35244 9/7/01 -- 0 9/8/01 -- 19945 9/9/01 -- 43446
Description	By supplying /msadc in the URL, it is possible to "escape" from the web root directory, and reach other directories that are not usually accessible through normal HTTP requests. This attack has recently been incorporated and identified with the Code Red attack too. **Sample exploit:** ---runaway.sh---- #!/bin/sh lynx -dump http://$1/msadc/..\%c0\%af../..\%c0\%af../..\%c0\%af.\./winnt/system32/cmd.exe\?/c\+$2+$3+$4+$5+$6+$7
Correlations	Nancy L. Feder has a wonderful analysis of this attack here: http://www.sans.org/infosecFAQ/threats/SADMIND.htm Another description of the attack is located here:

	http://www.securiteam.com/exploits/6F00M2000A.html
Countermeasures	Update Windows and Solaris machines with latest patches. Disable sadmind by editing /etc/inetd.conf. Update and patch Microsoft IIS servers.
Action (Moderate Alert)	This attack can cause a lot of damage, but for the most part, it is a false-positive because most of these alerts were triggered by the attacking host scanning for new victims. The scans were unsuccessful. The reason this attack was selected for review is because of the severity of the attack and the number of times it was attempted on the network. This attack was attempted 142654 times in the 5 day time period that the logs were analyzed. The severity of this attack is extreme as a successful attack will give the attacker root level access. Some examples of the attempts against the University follow: 09/05-06:04:59.213133 [**] WEB-MISC Attempt to execute cmd [**] 203.224.12.142:2431 -> MY.NET.86.164:80 09/05-06:05:02.147913 [**] WEB-MISC Attempt to execute cmd [**] 203.247.220.188:1198 -> MY.NET.149.85:80 09/05-06:05:02.599043 [**] WEB-MISC Attempt to execute cmd [**] 130.251.22.228:1982 -> MY.NET.54.156:80 09/05-06:05:03.544593 [**] WEB-MISC Attempt to execute cmd [**] 217.230.139.20:4369 -> MY.NET.195.19:80 09/05-06:05:03.857163 [**] WEB-MISC Attempt to execute cmd [**] 211.90.164.34:3380 -> MY.NET.85.197:80 These attacks both originate and are destined for various hosts. These attacks were extrememly prevelant as well. All vulnerable hosts should be immediately patched to assure no systems are compromised by this attack.

Name	Possible trojan server activity
Occurrences	3191 total Date/Alerts 9/5/01 -- 2188 9/6/01 -- 0 9/7/01 -- 3 9/8/01 -- 7 9/9/01 -- 993
Description	There are two hosts inside the MY.NET network that show signs of being compromised by the Subseven

	Trojan. Heavy traffic from the two systems on port 27374 (the common Subseven port) indicates that they may be compromised. SubSeven is often used as a Trojan Horse, which allows an intruder to deliver and execute any custom payload and run arbitrary commands on the affected machine. This control includes the ability to read, modify, and delete confidential information. Additionally, the intruder may use the affected computer as a launching point for additional attacks (namely, denial of service). Example of detect: 09/05-14:53:32.255960 [**] Possible trojan server activity [**] MY.NET.235.14:6346 -> 149.2.31.6:27374
Correlations	SANS: www.sans.org/newlook/resources/IDFAQ/subseven.htm Good write-up of Subseven: http://www.sans.org/y2k/practical/Robert_Mcmillen_gcih.doc CERT: http://www.cert.org/incident_notes/IN-2001-07.html Others: http://www.ntsecurity.net/Panda/Index.cfm?FuseAction=Virus&VirusID=616
Countermeasures	Install latest virus detection software, block port 27374 at the firewall.
Action (High Alert)	MY.NET.207.42 and MY.NET.235.14 are probably compromised. They show extensive traffic to hosts 207.69.129.186 and 149.2.31.6. The 207.42 host was transmitting data on the 9th, the 235.14 host was on the 5th as well. There were several false positives generated by this signature, primarily when a host arbitrarily allocated port 27374 for a communication channel with a service like WWW or DNS. 09/05-06:48:25.126786 [**] Possible trojan server activity [**] 64.37.200.46:27374 -> MY.NET.1.9:53 It would be advisable to check both of the MY.NET machines for a compromise and follow the countermeasures listed above. There may be more hosts compromised by this Trojan, these two hosts are the

	primary candidates because they had the most numerous amount of alerts in respect to their addresses. MY.NET.235.14 09/05-14:54:09.092070 [**] Possible trojan server activity [**] MY.NET.235.14:6346 -> 149.2.31.6:27374 09/05-14:54:09.135987 [**] Possible trojan server activity [**] MY.NET.235.14:6346 -> 149.2.31.6:27374 MY.NET.207.42 09/09-15:45:23.113368 [**] Possible trojan server activity [**] MY.NET.207.42:1214 -> 66.31.207.63:27374 09/09-15:45:23.114610 [**] Possible trojan server activity [**] MY.NET.207.42:1214 -> 66.31.207.63:27374 This attack was selected because of the severity of the signature, the number of alerts (over 3000), and the fact that the hosts listed here have a high possibility of being compromised. The severity of this attack is simple, if these hosts are found to be using the subseven Trojan, they are compromised, and the user has administrator privelidges. There are many signs that the above listed hosts have been compromised and should immediately be taken off-line and audited.

Name	spp_http_decode: IIS Unicode attack detected
Occurrences	58 total Date -- Alerts 9/5/01 -- 18 9/6/01 -- 17 9/7/01 -- 0 9/8/01 -- 3 9/9/01 -- 20
Description	This attack exploits a known problem with the Microsoft ISS web server. The attacker can pass commands and change directories at will by using Unicode. By encoding the '/' character in UTF8 (which results in the 2 byte value 0xc0af), IIS fails it's safety check to properly canocalize the URL, leaving the UTF8 characters in the filename. However, when IIS passes the filename to the underlying OS, the OS interprets the

	UTF8 characters, and therefore serves up a different file than IIS was expecting. The Unicode exploit is present in a variety of worms. The most prevalent worm that incorporates this attack currently is the Nimda worm. Example of trace: 09/08-15:50:07.914997 [**] spp_http_decode: IIS Unicode attack detected [**] 64.12.97.8:33832 -> MY.NET.253.125:80
Correlations	This is one of the SANS top 20 vulnerabilities: http://www.sans.org/top20.htm Roy Hutchison has a good analysis of this exploit here: http://www.sans.org/y2k/practical/roy_hutchison_GCIH.zip Bugtraq: http://www.securityfocus.com/bid/1806 Arachnids: http://www.whitehats.com/info/IDS452 A good write up on how Unicode can be used to subvert IDS's: www.securityfocus.com/focus/ids/articles/utf8.html
Countermeasures	Patch all Microsoft IIS web servers immediately.
Action (High Alert)	This attack is very devastating if successful. From the log analysis, it looks as if a couple of the systems on the network are probably compromised because they are attempting to attack other machines. The hosts that were identified as possibly compromised are MY.NET.98.148 09/09-10:42:29.375057 [**] spp_http_decode: IIS Unicode attack detected [**] MY.NET.98.148:1511 -> 217.146.193.5:8080 09/09-10:42:29.375057 [**] spp_http_decode: IIS Unicode attack detected [**] MY.NET.98.148:1511 -> 217.146.193.5:8080 09/09-10:42:29.375057 [**] spp_http_decode: IIS Unicode attack detected [**] MY.NET.98.148:1511 -> 217.146.193.5:8080. MY.NET.20.10 09/09-15:40:20.267950 [**] spp_http_decode: IIS Unicode attack detected [**] MY.NET.20.10:56213 -> 211.218.150.128:8080 09/09-15:40:20.267950 [**] spp_http_decode: IIS Unicode attack detected [**] MY.NET.20.10:56213 -> 211.218.150.128:8080

	This attack was chosen because of the two hosts that are possibly compromised and the fact that this attack is devastating in terms of repercussions. If successful, this attack will give the attacker administrator rights. These systems should be checked for integrity immediately. Please follow the suggested countermeasures listed above.

Name	High port 65535 tcp - possible Red Worm - traffic
Occurrences	1168 total Date -- Alerts 9/5/01 -- 25 9/6/01 -- 23 9/7/01 -- 19 9/8/01 -- 36 9/9/01 -- 1065
Description	The "Code Red" worm is self-replicating malicious code that exploits a known vulnerability in Microsoft IIS servers. The "Code Red" worm attempts to connect to TCP port 80 on a randomly chosen host assuming that a web server will be found. Upon a successful connection to port 80, the attacking host sends a crafted HTTP GET request to the victim, attempting to exploit a buffer overflow. Once the system is compromised, it is self-propagating and will send out the "HTTP GET" request to randomly selected hosts. Example alert: 09/09-02:15:43.987044 [**] High port 65535 tcp - possible Red Worm – traffic[**] MY.NET.222.185:80 -> 194.175.74.65:65535
Correlations	There are many good write ups about this worm: http://www.incidents.org/react/code_red.php http://www.cert.org/advisories/CA-2001-23.html http://www.cert.org/advisories/CA-2001-19.html http://www.symantec.com/avcenter/venc/data/codered.worm.html http://www.caida.org/analysis/security/code-red/ http://www.eeye.com/html/Research/Advisories/AL20010717.html

	http://www.incidents.org/react/code_redII.php http://archives.neohapsis.com/archives/incidents/2001-08/0092.html
Countermeasures	As always, apply all Microsoft patches. Any system that is running IIS is vulnerable to this attack. Also, all Cisco 600 series DSL routers are vulnerable. All unexploitable web servers will probably log the request.
Action (Moderate-High Alert)	This signature has picked up a lot of false positives when hosts arbitrarily allocate a high port for services like SMTP and DNS. Nevertheless, there are some signs of compromised hosts on this network. There are also many indications of external hosts that have been compromised. 164.106.165.170 has probably been compromised and is randomly scanning some of the MY.NET hosts. This can be proven with this excerpt from the Sept 9th logs. [**] 164.106.165.170:65535 -> MY.NET.7.65:8080 09/09-19:31:51.584083 [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 -> MY.NET.7.141:8080 09/09-19:31:51.966607 [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 -> MY.NET.7.179:8080 09/09-19:31:51.993278 [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 -> MY.NET.7.182:8080 09/09-19:31:52.163610 [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 -> MY.NET.7.199:8080 09/09-19:31:52.383103 [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 -> MY.NET.7.221:8080 09/09-19:31:56.165547 [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 -> MY.NET.9.91:8080 09/09-19:31:56.913856 [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 -> MY.NET.9.166:8080 09/09-19:31:56.944497 [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 -> MY.NET.9.169:8080 09/09-19:31:56.973717 [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 -> MY.NET.9.172:8080 09/09-19:31:57.295406 [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 -> MY.NET.9.204:8080 09/09-19:31:57.394085 [**] High port 65535 tcp - possible Red Worm - traffic

	[**] 164.106.165.170:65535 -> MY.NET.9.214:8080 09/09-19:31:57.686198 [**] High port 65535 tcp - possible Red Worm - traffic [**] 164.106.165.170:65535 -> MY.NET.9.243:8080 The internal host MY.NET.222.185 may also be compromised. This host is triggering the alert signature on outbound traffice which means that either this host has been compromised and is attempting to propogate, or the receiving host (194.175.74.65) dynamically allocated port 65535 for WWW. The host MY.NET.222.185 should be audited immediately. 09/09-02:15:43.987044 [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.222.185:80 -> 194.175.74.65:65535 09/09-02:15:44.320861 [**] High port 65535 tcp - possible Red Worm - traffic [**] MY.NET.222.185:80 -> 194.175.74.65:65535 We also notice that a host we referenced earlier (the top scanner), used this attack on the system it was scanning. This looks like active targeting because the attacker began by learning about the system through scanning, and then attempts an attack on it. The logs from the 6th show the attempt more clearly: 09/06-12:33:33.817479 [**] SCAN FIN [**] 204.50.141.133:9388 -> MY.NET.105.120:5679 09/06-12:55:09.616606 [**] SCAN XMAS [**] 204.50.141.133:63921 -> MY.NET.105.120:8130 09/06-12:58:13.241230 [**] Probable NMAP fingerprint attempt [**] 204.50.141.133:4965 -> MY.NET.105.120:3658 **09/06-14:44:11.652641 [**] High port 65535 tcp - possible Red Worm - traffic [**] 204.50.141.133:55727 -> MY.NET.105.120:65535** This attack was chosen due to the high level of evidence that a host has been attacked as well as the prevalence of this attack on the network.

Name	External RPC call
Occurrences	242 total Date -- Alerts 9/5/01 -- 12 9/6/01 -- 13 9/7/01 -- 18

	9/8/01 -- 23 9/9/01 -- 176
Description	These alerts were triggered when a host that is external to our network sent a packet to port 111 of any of our internal hosts. This alert is notifying us of a scan. By analyzing the alerts and the timing of the alerts, we can quickly infer that this is a scan. 09/08-15:08:32.538384 [**] External RPC call [**] 209.209.13.57:4644 -> MY.N ET.132.84:111 09/08-15:08:34.546667 [**] External RPC call [**] 209.209.13.57:1058 -> MY.N ET.133.218:111 09/08-15:08:34.548508 [**] External RPC call [**] 209.209.13.57:1090 -> MY.N ET.133.250:111 09/08-15:08:35.525629 [**] External RPC call [**] 209.209.13.57:4642 -> MY.N ET.132.82:111 09/08-15:08:36.547909 [**] External RPC call [**] 209.209.13.57:2014 -> MY.N ET.137.153:111 09/08-15:08:36.547985 [**] External RPC call [**] 209.209.13.57:1999 -> MY.N ET.137.138:111 09/08-15:08:36.548131 [**] External RPC call [**] 209.209.13.57:1967 -> MY.N ET.137.106:111 09/08-15:08:37.554722 [**] External RPC call [**] 209.209.13.57:2054 -> MY.N ET.137.193:111 Note how the source address is the same in all of these alerts. Also note how fast each of these systems were queried. This is clearly a scan to find out which hosts on the network have the RPC services available. The reason the attacker is scanning specifically for port 111 (RPC services) is that many of the services offered by RPC are traditionally susceptible to exploitation. Services such as statd, nis, nfs, mountd, and others are very tempting to an attacker because they often times are easily compromised. There are many tools which are readily available on the net which can exploit these services.
Correlations	This service is listed as one of the SANS "Top Ten": http://www.sans.org/topten.htm This service is listed as on the SANS "Top 20": http://66.129.1.101/top20.htm

	Bugtraq list of exploits for RPC: http://www.securityfocus.com/cgi-bin/vulns.pl?keyword=rpc&section=keyword A site which has some RPC exploit programs: http://www.phreak.org/archives/exploits/unix/rpc-exploits/ Chris Kuethe’s GCIA practical refers to many false positive for RPC as well http://www.sans.org/y2k/practical/chris_kuethe_gcia.html#2.6 Robert Sorenen’s description of this alert http://www.sans.org/y2k/practical/Robert_Sorensen_GCIA.htm#ss-5
Countermeasures	RPC services should be disabled if not needed. If these services are required for the local network, a firewall should block all external hosts from querying port 111. If the RPC service is needed externally, the service should have the latest patches applied to it and an IDS should be in place to monitor activity to the service. RPC is a dangerous service and should be monitored heavily.
Action (Moderate)	There are hosts that are scanning the network for RPC. Some of the scans look like they are actively targeting certain hosts. If this is true, then great caution must be taken to make sure the hosts are not vulnerable to exploitation. It is difficult to defend RPC because it houses so many services, but if it must be run, patching the system must be a priority. There are many attempts to access RPC services on the network. These attempts were never followed by a reply from any of the internal hosts so it is hopeful and likely that none of the attempts were successful. 09/09-06:59:47.113618 [**] External RPC call [**] 128.174.115.14:2889 -> MY.NET.135.107:111 09/09-06:59:47.117532 [**] External RPC call [**] 128.174.115.14:2912 -> MY.NET.135.130:111 09/09-06:59:47.128145 [**] External RPC call [**] 128.174.115.14:2940 -> MY.NET.135.158:111 09/09-06:59:47.129454 [**] External RPC call [**] 128.174.115.14:2948 -> MY.NET.135.166:111 The reason this alert was selected was due to the severity of the attack if successful and also the variety of hosts that were probed. Many UNIX hosts run RPC services that are vulnerable (like statd) and therefore these scans may eventually find a system on the university network that is

	running an exploitable service. Great care needs to be taken to make sure no vulnerable RPC services are running on the University computers.

3.7.0 Summary

There is an abundance of malicious traffic flowing through the Universities network. We have seen numerous portscans, root level attacks, and out of spec data. Much of the traffic is harmless in that the systems that are being attacked are not susceptible to the exploits being forced upon the system. There are many signs that show some compromised systems on the network though. The lack of access controls and sporadic logging of IDS data hinders the security prowess of this network. By implementing a firewall scheme, refining the IDS rule set, and maintaining IDS uptime (and integrity), the University network will be much more secure. Granted, this is a University, and it is presumably difficult to implement numerous access controls, but by simply enacting a few rules on a firewall, the network would be much safer. By cleaning the IDS rule set, the security analyst will have more time to research hostile traffic with malicious intent rather than wasting valuable man hours investigation what turns out to be a false positive. By improving the rule set, the IDS will also be more efficient by dropping less packets. Uptime is also a concern. While we saw a plethora of attacks and scans on the network, there are immense time gaps on two of the five days of logs. Without any data for these periods, it is impossible to determine what happened on the network and what attacks/scans were attempted. Uptime for the University IDS is essential. Finally, there should be policies developed which would require University computer staff to update the software and services on the network. If the systems are patched and fully up to date, it is difficult for an attacker to penetrate a system.

With these recommendations followed out, the University network will be much more secure and efficient. Thanks again for allowing us to assess the security of this network. It has been difficult but fulfilling work. We hope that you continue to be safe.

3.8.0 Description of Analysis Process (Informal)

The process that I used to derive my results was closely tied to the flow of this document. Other than the initial “overview” section, I for the most part did each section progressively. To begin I did a brief overview of the data by manually looking through the text. My next step was to research tools that could help me analyze the data. I found a couple (snortsnarf and snort_stat) on the Snort web site. I had trouble finding tools that could process the plain text logs. At work, I use tools like ACID, mySQL, and Demarc to analyze the data, but those tools require the binary logs. In this case, I only had access to the text logs so I had to use new tools. I also had to write a few perl scripts of my own because of the limited capabilities of the text processing tools I mentioned earlier. I also utilized many of the standard Unix tools such as wc, cat, grep, less, uniq, etc. Once I processed the data, I began to look for trends. There was an immense amount of data to analyze, so again, I felt I was at a disadvantage not having access to the binary logs. I manually calculated and plugged data into the excel workbook to develop the graphs. I wanted to find correlations between certain hosts and malicious activity. I was also interested in the time frames of increased activity. I was looking to find a spike in alerts at a late time period but I found none. In fact, most of the time scale information was to be expected. The only oddity was the sheer lack of alerts on the 7th. It took a long time to develop the timescale graph and I didn't feel it helped out in my analysis as much as I would have hoped. Next, I wanted to see how many of the attacks were originating from within the network and how many were external. Once these graphs were developed and the other data was processed (registration information, top talkers lists, alerts, etc) I felt I was ready to begin the real analysis. I concatenated all the logs and pushed them through Snortsnarf. I then had a huge list of alerts on a day by day basis, which was extremely beneficial. I must say thanks to Jim Hoagland, Stuart Staniford and the Silicon Defense team for their great contributions to the security community. I used both Spade and Snortsnarf extensively in compiling this practical. Once I had the alerts in a presentable format, I continued by researching each and every one of the alerts. While I had heard about most of the attacks, I did not know exactly what they were, so it was a great learning experience to research so many exploits and attacks. Once I completed that phase (and let me tell you, it took a long time to look up all those alerts), I began to try to correlate the more severe attacks with the hosts that I had seen before in the logs. I chose my "top 5" based on these characteristics. While I wanted to be diverse in the specific attacks I chose to investigate, I found that as an analyst for hire, it would be in the best interest of the University to explore the alerts that could potentially cause them the most damage, rather than research a less common less severe alert which would be more exciting to me personally. Once I completed it, I felt strongly that the detects I chose to analyze were worthy and would have benefited the University if it were a real security assessment. Overall, the assessment took me much longer than I had anticipated. It was a very gratifying experience and I think that it is a wonderful learning experience. It forces the analyst to think out of the box and to utilize techniques that they otherwise may not have used (in my case by using those text processing tools instead of ACID or something). I also think that the practical as a whole is structured so that it requires the analyst to explore and attempt to learn about many aspects of the profession. Between the security assessment, the signature analysis, and the research document, this practical covers nearly every facet of the intrusion detection analyst profession, and maybe more.