Tim Newell
SANS CDI West
San Francisco, CA
December 2001

GIAC Certified Intrusion Analyst
Practical Assignment, Version 3.0

Table of Contents

Assignment 1 – Describe the State of Intrusion Detection: "Dragon – Features and Benefits"

Introduction

The intent of this paper is to describe some of the key features and strengths of the Dragon IDS suite from Enterasys Networks. Joni Ramos has written an article entitled "DRAGON – An Intrusion Detection System" that reviews the basic components and features of the Dragon architecture. This paper is available from the SANS Reading Room at http://rr.sans.org/intrusion/dragon.php. Although this paper may overlap somewhat, the emphasis of this article in on:

- describing my impressions of the product as a result of an internal evaluation project and subsequent tracking of development and new reviews;
- providing a brief history of the product; and
- a presentation of specific strengths and advantages.

It is suggested that readers who are not at least somewhat familiar with Dragon or other commercial IDS systems consider reviewing some of the background papers in the SANS Reading Room such as Joni's, as this paper will not go into as much detail on these topics, in order to focus better and minimize duplication.

Background

During the summer of 2000, I was given an assignment to review and evaluate several commercial IDS products for my company. The intent was to identify three of the enterprise-class solutions available, review them, and make some recommendations for two planned IDS deployments. After the initial requirements gathering phase, I spent a lot of time researching the available products. This included reading most of the available product reviews at the time, buyer's guides from the Computer Security Institute and ICSA Labs (now TruSecure), identifying other products through IDS information sites such as Talisker's Intrusion Detection System List (http://www.networkintrusion.co.uk/ids.htm), and going through white papers and brochures from individual vendors. Other influences on some of the criteria included Thomas Ptacek and Timothy Newsham's "Insertion, Evasion, and Denial of Service: Evading Network Intrusion Detection" paper and Marcus Ranum's "Intrusion Detection and Network Forensics" USENIX Tutorial.

Although NIDS (Network Intrusion Detection System) products are what most people first think of when talking about IDS, host-based systems (HIDS – Host Intrusion Detection System) have been around for longer, and offer a complementary set of strengths and weaknesses. A HIDS can often detect types of attacks a NIDS may not be well suited for, as it uses a different perspective and set of inputs. They are also very useful in environments where the effectiveness or coverage of a NIDS is reduced, such as in heavily switched networks. On the other hand, NIDS products can detect attacks or attempted attacks a HIDS might miss; monitor many hosts from one point; and is generally less intrusive to deploy in an environment. Generally, where one approach is weak, the other tends to be strong. Combined, the two IDS approaches can provide maximal coverage, while also providing deployment options to handle a variety of different environments where one particular approach may be ineffective. It is interesting to note that since the time I conducted my initial review, recent acquisitions and partnerships have resulted in most of the major commercial vendors now supporting both techniques, at least to a greater or lesser extent.

One of the key criteria we were looking for was a reliable product that would scale to handle large networks with a distributed management model (various sensors reporting to one or more central management consoles) and support for both network- and host-based detection capabilities. This last requirement quickly eliminated two or three well-known vendors who had strong offerings on the network (NIDS) front, but no host-based (HIDS) component. We wanted a fairly flexible solution that could be used in a variety of situations and client environments, although we recognized that no single IDS product will suite every customer.

The three initial products selected for review didn't include Dragon. At the time it was a relative newcomer to the industry, and its management interface and host based capabilities didn't seem overly strong yet. The emphasis of the review was placed on three of the well-known, mature products.

During the background research, it became apparent from a variety of sources that the IDS industry was generally a bit immature. Each product or suite had its strengths, but also some real weaknesses; no single product was strong, or even solid, in all areas. While this was easy to recognize in an academic or theoretical sense, focusing on some of the products in detail during the review really drove this reality home in a whole new way. It quickly became clear that all three of the products that had been selected suffered from some real weaknesses. Some didn't handle busy networks well; others still hadn't implemented functionality to handle even the more basic techniques described in Ptacek and Newsham's 1998 paper; integration between HIDS and NIDS components was poor in cases; where a product had a strong HIDS component, it tended to be weak on the NIDS front, or vice versa; and access to the details of how signatures operated and what they looked for, or the ability to create customized signatures, was often poor.

At about the same time, Dragon was getting some favorable attention on the IDS mailing lists and newer reviews. Although it had briefly been looked at earlier in the review, it was decided that a closer look was warranted, and so Dragon became the fourth product reviewed. Although it had, and continues to have, weaknesses, Dragon wound up being the selected as a result of the review. Since that time, I have continued to monitor new IDS reviews that have been published, the Dragon and IDS mailing lists, and similar sources. I have continued to be pleased with what I've seen of Dragon's capabilities, development, and support. Some of its weaker areas have improved by a fair amount since 2000, and I have generally become more confident of my initial decision in favor of the product.

General Product History

Ron Gula's company, Network Security Wizards, originally developed Dragon. The initial versions were strictly network-based. Unlike many of the competing products, Dragon was based on a Unix platform. Much of its emphasis was on delivering stability and performance. Reliably detecting a variety of attacks, particularly those using advanced techniques such as those described in the Ptacek and Newsham paper, was a major focus. (The Dragon documentation at the time of the original review put significant emphasis on explicitly describing how the product dealt with such techniques.) Dragon had something of an advantage over some of its more mature competition in this area, as it was being actively developed at about the same time as the paper was published, allowing such issues to be designed into the core product from the start. The more established competition had to deal with reengineering and retrofitting such functionality in many cases, rather than being able to write from scratch.

Overall, Dragon has undergone a very rapid series of development phases and improvements. The development team appears to be consistently and aggressively improving and rounding out the application. Early NIDS-only versions of the product, such as the one reviewed by Greg Shipley during 1999 in "Intrusion Detection, Take Two," (http://www.networkcomputing.com/1023/1023f1.html), supported just a command line interface and simple web-based reporting tool. Neither a central console nor HIDS capability had been developed. Although it received high marks for handling advanced attack techniques under heavy network load even at that time, usability was an issue, as the product required significant comfort with Unix and was relatively cryptic.

During late 1999 and early 2000, several major improvements were made. A centralized management console, the Dragon Server (or Sorcerer at the time), was introduced. Interestingly, the console utilized a web-based interface, as opposed to the Windows-based GUI used by virtually all of the competition. This had some interesting implications for concurrent access to the console by administrators and intrusion analysts, as well as making remote access a non-issue (many of the Windows-based interfaces were quite limited in terms of remote access capabilities). Another difference was the emphasis of the tool on a more forensic type of analysis, providing detailed drill-down capabilities, while largely forgoing the more real-time, flashing-lights style of interface used in other products.

At about the same time Dragon Server was released, the first versions of the Dragon Squire were also being rolled out. Dragon Squire is the HIDS component of the IDS suite. Originally developed for Unix, it was ported to Windows NT/2000 during the original review period. Squire supports signature-based pattern matching for various types of logs, as well as file property and integrity checking (including centralization of file checksums on the Dragon Server). One of the other interesting capabilities of Dragon Squire is its ability to monitor SNMP and syslog output from various infrastructure and security devices, such as routers and firewalls.

In September of 2000, Enterasys Networks acquired Network Security Wizards. Although acquisitions always introduce turmoil and uncertainty, and can even kill product lines at times, there was a positive note to this exchange. Network Security Wizards was a relatively small, self-funded startup company. As such, its development resources and supporting infrastructure were limited. As a large, publicly traded company, Enterasys had significant assets that it could bring to bear in order to further develop and refine the Dragon product line. It appears that this is the course the acquisition has taken, as the product's development continues to be rapid, but the overall polish, process, and consistency of the suite have also been improved significantly.

In August 2001, Greg Shipley and Patrick Mueller released the latest of their well-known IDS product reviews. Entitled, "To Catch a Thief" (http://www.nwc.com/1217/1217f1.html), this review utilized some of the most realistic and strenuous testing conditions I've seen published, and encompassed a variety of different products. Dragon was their product of choice in this version of the review, although they still highlighted several issues they had with it.

Dragon Version 5 was released in October of 2001. This was the first major version release since the Enterasys acquisition. Much of the focus of the new release was on improvements to the underlying architecture. Integration and consistency among the various components was an emphasis. Performance optimizations and scalability were other key goals. A strong focus on providing an effective IDS solution for the enterprise and Managed Security Services Provider (MSSP) markets became particularly evident. This included the introduction of modular components within the architecture, to better support varying deployment requirements, including the ability to "tee" event output from sensors to multiple consoles. This is particularly useful for doing things such as providing a separate read-only console to clients of a managed service offering, while aggregating various customers' data to provide a strategic view of events to the service provider.

Pros and Cons

Now that I have provided some history on my knowledge of Dragon and its development, this section will highlight some of the areas I consider to be particular strengths and weaknesses of the product:

Strengths

- ***Performance*** – Performance, and the ability to handle heavily loaded networks, has been a major strength of Dragon for some time. It has been demonstrated to handle heavily loaded 100 megabit networks for some time, and recent unconfirmed reports indicate new versions are beginning to handle even moderately loaded gigabit networks on a sustained basis. Considering how poorly some of the competition behaves on even moderately busy 100 megabit segments, this is quite impressive.

- ***Handling of Advanced Techniques*** – Dealing with advanced attack techniques, whether involving fragment games, alternate character and protocol encodings, or other approaches, has been a focus of Dragon's development. The current product supports a variety of protocol decoding and normalization options, and options such as robust fragment reassembly have been present since the early NIDS-only product versions.

- ***Signatures and Tuning*** – Signatures are one of the key differentiators of Dragon from many other commercial IDS products. The full content of all signatures is provided with the product. Users can view the signatures themselves in order to gain an understanding of exactly what the signature will match on (and what it won't). This may sound minor to analysts who are largely familiar with Snort, but this is relatively rare on the commercial front, and has been available since before Snort became as popular as it is currently. It was incredibly frustrating to not have access to this kind of detail when assessing other products. Some of them had signatures along the lines of "FTP Server Exploits", with descriptions like "This signature alerts on various FTP server attacks." Huh? Which attacks? How does it detect them? More importantly, which ones *doesn't* it alert on?

  The signature language itself is quite flexible and fairly simple. Users can both customize standard signatures and write their own from scratch. This degree of flexibility, combined with various filtering and decoding options, allow Dragon to be tuned for an environment to a very high degree. One of the enhancements in the current beta software version is the introduction of a new Squire architecture that supports an API through which custom modules can be added, further extending the product's flexibility.

  Signature updates are very regular, with priority updates often being distributed with impressive speed. Signature coverage (and specifically signature counts) seem to have become regarded as the "snake oil" of the IDS marketing world. Different vendors create different types of signatures, and these signatures detect events differently. Directly comparing signature counts is thus very inaccurate and even misleading. However, I feel that it is reasonable to at least make some general observations, especially when you begin to approach order-of-magnitude scale differences in signature counts. Dragon's signature base is very comprehensive (I believe the specific count of standard signatures stands somewhere at or above 1500 currently), incorporating a number of different signature categories.

- ***Analysis Capabilities*** – The Dragon Server interface makes it very easy to drill down into events based on event types or addresses involved. Specific technical details can be examined about individual events, including packet headers and contents. A recently introduced correlation tool is also available, which imports the results of Nessus scans of the protected network into the analysis interface, allowing known vulnerabilities to be correlated against exploit attempts in order to assist in prioritizing and analyzing events.

- ***Vendor Support and Development*** – As was mentioned above, Dragon continues to be actively developed. Vendor staff actively participate in the product mailing list, and frequently update customers on new and upcoming developments and changes. Software updates are quite regular, and typically follow an incremental development model, so that new features and improvements become available fairly quickly. I have found support from both the vendor and the user community to be very good. During the evaluation project, I encountered a number of frustrations when dealing with different vendors, especially when seeking answers to technical questions and clarifications. I found that Network Security Wizards (this was largely prior to the acquisition, but my observations from the product mailing list continue to bear this out) were very responsive, however, and provided solid answers to a wide variety of questions I posed in an extremely short period of time.

- ***Management*** – Management and distribution of signature, policy, configuration, and minor software updates can generally be accomplished via Dragon's web interface (following initial installation of course). Although there is always room for improvement in areas like this, Dragon offers some good options for managing the deployment of changes across both individual and groups of sensors. This is critical for keeping signatures up to date and managing large IDS deployments.

- ***Remote Access*** – Dragon supports a couple of different options that facilitate remote access. Its roots lie in a command-line interface accessed from the physical console or SSH. This kind of interface is extremely efficient for remote access, whether across a LAN, WAN, or even over a dialup connection from home when you get one of those middle-of-the-night pages. The web interface, by its very nature, is also both accessible and efficient in any of these access scenarios as well. Both interfaces also make it possible to have multiple analysts concurrently accessing the console.

Weaknesses

- ***Interface*** – Although the web interface provides some very powerful capabilities, I have noticed different complaints in reviews about the usability of the interface. Some of the navigation options are a bit awkward at first, and some minor browser compatibility issues have been noted. The material presented is quite technical, which may be intimidating to new analysts (although probably desirable to experienced staff).

- ***Dragon Squire*** – Dragon Squire is probably the newest major component of the product suite. Although it has improved and matured significantly since its introduction, it is probably fair to say that it is not as strong within the HIDS realm as the Dragon Sensor is in the NIDS arena.

- ***Pricing Model*** – Different IDS products are licensed in different ways. Some include the management console with the sensor cost, while others don't.

Dragon requires a separate purchase for the management console, although it is at least possible to run a Sensor or Squire without a Server. (Most of the other products require a central console, whether it is bundled or purchased separately.) The individual NIDS and HIDS licenses generally seem to be les expensive than those of other vendors, but the additional cost for the Dragon Server can make Dragon a more expensive capital investment for smaller deployments with only a few Sensors or Squires.

- ***Documentation, Style, Naming*** – I couldn't think of a better term for this point, unfortunately. This is one area where Dragon has improved significantly since the Enterasys acquisition. The documentation and interface I am most familiar with, from the Version 4 series, was very technical and provided a great deal of information about how the system worked. This is the type of material that is very important for a technical analyst. Some of the more minor areas, such as consistency, format, and style, were relatively weak, however. In particular, task-oriented documentation was fairly minimal. Also, the architecture, although modular, had a somewhat excessive collection of pieces, each with its own name and function. Although the names get points for inventiveness and "coolness," there were enough that it became confusing to sort out which piece provided what functionality, especially for new users or potential clients.

- ***Brand Awareness*** – Dragon is generally less well known than some of the other IDS products on the market, although this has changed somewhat over the past year or so. Nonetheless, some organizations that are new to this technology area, or more conservative in their IT strategies, tend to prefer products they have heard more about.

Summary

Dragon has quickly become one of the leading commercial IDS products. It is a powerful tool that delivers a great deal of technical capability. Analysts having a familiarity with Unix-based and/or open source products will find Dragon's approach very familiar and comfortable. (Dragon isn't open source, but has strong ties to a number of open source technologies, and a general "style" that those familiar with this approach may identify with.) As with any of the IDS technologies available, Dragon has both strengths and weaknesses. From my point of view, its strengths tend to coincide with the areas I place the highest priority on, while its weaknesses are generally in areas less important to me. Others will surely have different opinions, and may be able to highlight weaknesses I have overlooked. Regardless, I think it is fair to say that for those looking at purchasing a commercial IDS, Dragon is an alternative worth considering.

References

"DRAGON – An Intrusion Detection System", http://rr.sans.org/intrusion/dragon.php
"Talisker's Intrusion Detection System List", http://www.networkintrusion.co.uk/ids.htm
"Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection", http://www.nai.com/media/ps/nai_labs/ids.ps

"Intrusion Detection, Take Two", http://www.networkcomputing.com/1023/1023f1.html
"To Catch a Thief", http://www.nwc.com/1217/1217f1.html
"Dragon 5, An Intrusion Detection System for the Enterprise", http://www.gartner.com/webletter/enterasys/
"Dragon 5, An Intrusion Detection System for the Enterprise – Hot Sheet", http://www.enterasys.com/ids/hotsheet.pdf
"New Features in Dragon 5.0", https://dragon.enterasys.com/dragon5-GA/docs/dragon5-new.pdf

Assignment 2 – Network Detects

Unfortunately, I was unable to get my own IDS sensor deployed for a significant period of time to capture some useful detects. As I had no dedicated hardware, my regular use laptop was the only system I could use. Since it was in use for other things, and locked down to protect the other data on it, would not see anything beyond initial port scans, and could not leave it running as an IDS long term. I was thus forced to rely solely on the Intrusions mailing list for detects. Given the directive to be very careful about submitting commonly-occurring scans that have been analyzed many times over, my options were somewhat limited (common port scans and sweeps are the majority of the detects posted to the list). I have attempted to extract some of the more interesting posts from the past two months below. It is important to note that much of the initial analysis was already provided on the Intrusions list in the form of follow-up posts, which I have obviously seen. I have tried to credit those who provided this further analysis and direction within the detect analyses.

Detect 1: Port 6112 Scan

Source of Trace

Bill Scherr posted this detect to the Intrusions mailing list (http://www.incidents.org/archives/intrusions/msg03448.html) on January 21, 2002. Although many basic scans for port 6112 have been posted recently, Bill's traces provided greater technical detail than most, as well as useful initial analysis and background.

Detect Generated By

This appears to be a tcpdump capture.

Probability the Source Was Spoofed

Unlikely. The attacker will want to receive the responses from the scan in order for it to be of use, and so spoofing is unlikely.

Description of Attack

A wide TCP sweep across two different subnets within Bill's environment was noted. The scan appears to be a SYN scan for TCP port 6112. This port is associated with the CDE Subprocess Control Service (known as dtpsc – the daemon providing this service is the dtspcd). This service is a standard component of the Common Desktop Environment. This service is run on a wide variety of different Unix platforms. A vulnerability was recently announced in the service, as noted in CERT Vulnerability Note VU #172583.

Source port for the scans consistently 6112 as well. The scanning packets appear to be crafted, as they exhibit static characteristics across multiple hosts, with only occasional changes (which are then repeated for some period themselves). Specifically, sequence numbers, window size, TTL, and IP ID all remain static for a period, and then all change. Following the change, they remain static for a further period.

A sample of the scan consists of:

```
01/19/02 02:20:27.326083 211.39.32.104.6112 > One.Net.Here.162.6112: S
17578451:17578451(0) win 40 (ttl 243, id 41492)
01/19/02 02:20:27.331975 One.Net.Here.162.6112 > 211.39.32.104.6112: R 0:0(0)
ack 17578452 win 0 (ttl 255, id 31204)
01/19/02 02:20:27.336762 211.39.32.104.6112 > One.Net.Here.163.6112: S
17578451:17578451(0) win 40 (ttl 243, id 41492)
01/19/02 02:20:27.341824 211.39.32.104.6112 > One.Net.Here.164.6112: S
17578451:17578451(0) win 40 (ttl 243, id 41492)
01/19/02 02:20:27.344905 One.Net.Here.164.6112 > 211.39.32.104.6112: R 0:0(0)
ack 17578452 win 0 (ttl 255, id 31205)
```

Attack Mechanism

Appears to be synscan based sweep of large address spaces, based on discussion of this pattern and other very similar scans that were being widely experienced at the time. Donald Smith's GCIA practical included a comprehensive analysis of this tool, and his comments during the discussion seemed to quickly identify it as the tool in use.

Although the scan itself is fairly generic, during the time period there had been a major increase in the volume of scans directed at port 6112. The Honeynet project recorded and documented a Solaris 8 system compromise via a vulnerability in port 6112. Although scans appear widespread and common, with scattered reports of successful exploits, the attack vehicle used in the Honeynet compromise does not appear to have surfaced in the public realm yet. However, packet traces provided by the Honeynet project appear to be a fairly typical buffer overflow attack, binding a root shell to a high port.

The rapid increase in activity, combined with the vulnerability and compromise identified by the Honeynet Project, led to the release of a CERT advisory on the issue (reference below).

Correlations

Synscan discussions on the Intrusions mailing list:
http://www.incidents.org/archives/intrusions/msg03520.html

Similar port 6112 scans:
http://www.incidents.org/archives/intrusions/msg03435.html (wide port 6112 scans, does not appear to be synscan)
http://www.incidents.org/archives/intrusions/msg03391.html
http://www.incidents.org/archives/intrusions/msg03768.html
http://www.incidents.org/archives/intrusions/msg03774.html

Donald Smith's GCIA practical and synscan analysis:
http://www.giac.org/practical/donald_smith_gcia.doc

Honeynet packet log and attack description:
http://project.honeynet.org/scans/dtspcd/dtspcd.txt

CERT Advisory: http://www.cert.org/advisories/CA-2002-01.html

CERT Vulnerability Note: http://www.kb.cert.org/vuls/id/172583

Snort signature description for “EXPERIMENTAL CDE dtspcd exploit attempt”: http://www.snort.org/snort-db/sid.html?id=1398

Evidence of Active Targeting

The correlations above, as well as Bill’s description of a wide sweep of his address space in the original detect reviewed, indicate that this is a general scan of a large block of addresses, and not targeted at any single host. It is assumed that positive responses from scanned hosts would likely be followed with attacks similar to the one launched on the Honeynet system.

Severity

Using the formula:
Severity = (Criticality + Lethality) – (System + Network Countermeasures)

We determine the following characteristics:
Criticality: 3 (unknown what hosts, but presumably at least Unix workstations, maybe servers)
Lethality: 1 (scan)
System Countermeasures: 3 (again, environment is unknown, but will take a conservative stance, particularly as this is a new vulnerability and patches may not be deployed)
Network Countermeasures: 2 (IDS in place to detect, but reset packets observed coming from destination IP’s)

Result:
Severity = (3+1) – (3+2) = -1

Although the initial scan is not of critical severity, the weak network countermeasures and new vulnerability are a concern. A follow-up attack would likely be scored as follows:
Criticality: 3
Lethality: 5
System Countermeasures: 3
Network Countermeasures: 2

Severity = (3+5) – (3+2) = 4!

Defensive Recommendations

6112 is part of CDE. Best practices indicate this should be disabled if not in use, and firewalled from outside access. If used, ensure current patches are applied to protect against both internal and external attacks.

Multiple Choice Test Question

Given the following trace:

```
01/19/02 02:20:27.326083 211.39.32.104.6112 > One.Net.Here.162.6112: S
17578451:17578451(0) win 40 (ttl 243, id 41492)
01/19/02 02:20:27.331975 One.Net.Here.162.6112 > 211.39.32.104.6112: R 0:0(0)
ack 17578452 win 0 (ttl 255, id 31204)
01/19/02 02:20:27.336762 211.39.32.104.6112 > One.Net.Here.163.6112: S
17578451:17578451(0) win 40 (ttl 243, id 41492)
01/19/02 02:20:27.341824 211.39.32.104.6112 > One.Net.Here.164.6112: S
17578451:17578451(0) win 40 (ttl 243, id 41492)
01/19/02 02:20:27.344905 One.Net.Here.164.6112 > 211.39.32.104.6112: R 0:0(0)
ack 17578452 win 0 (ttl 255, id 31205)
```

Which hosts were found to have port 6112 active as a result of this scan?
A) One.Net.Here.163 is running dtspcd
B) One.Net.Here.162 and One.Net.Here.164 are running dtspcd
C) Both A) and B)
D) None of the above

Answer: D) The resets from .162 and .164 indicate they are not running the service, but no response is noted from .163 (a syn-ack would be the expected response if it were running the service).

Detect 2: Nessus Scan

Source of Trace

This detect was posted by Jim Slora to the Intrusions mailing list at incidents.org (http://www.incidents.org/archives/intrusions/msg03527.html) on January 28, 2001. Further details were provided in http://www.incidents.org/archives/intrusions/msg03540.html. It initially caught my eye as an intensive, rather unusual scan that was quite atypical from other activity posted on the list.

Detect Generated By

Unknown. Detect appears to be a hand-summarized report extracted from logs.

Probability the Source Was Spoofed

Unlikely. The attacker will want to receive the responses from the scan in order for it to be of use, and so spoofing is unlikely.

Description of Attack

An initial ping was detected from an IP address in China. Approximately three hours later, a fairly intense port scan and subsequent CGI probes were carried out.

Extracts of the actual scan report include:

```
Web Requests:
http://my.ip.Net1.Host1/
http://my.ip.Net1.Host1/qweiop40809440fsfjflr.html
http://my.ip.Net1.Host1/
http://my.ip.Net1.Host1/../../../../../etc/passwd
http://my.ip.Net1.Host1/../../../../etc/passwd
http://my.ip.Net1.Host1/../../../../etc/passwd
…
http://my.ip.Net1.Host1/web_store.cgi
http://my.ip.Net1.Host1/usr/local/apache/share/htdocs/.htaccess
http://my.ip.Net1.Host1/userreg.cgi?cmd=insert</=eng&tnum=3&fld1=test999%0ac
at</var/spool/mail/login>>/etc/passwd
http://my.ip.Net1.Host1/root
http://my.ip.Net1.Host1/quikstore.cfg
http://my.ip.Net1.Host1/pw/storemgr.pw
…
http://my.ip.Net1.Host1/prxdocs/misc/prxrch.idq
http://my.ip.Net1.Host1/prxdocs/misc/prxrch.idq
http://my.ip.Net1.Host1/piranha/secure/passwd.php3
…
http://my.ip.Net1.Host1/domcfg.nsf
http://my.ip.Net1.Host1/database.nsf
http://my.ip.Net1.Host1/cool-logs/mylog.html
http://my.ip.Net1.Host1/cgi-src/phf.c
http://my.ip.Net1.Host1/cgi-src
http://my.ip.Net1.Host1/cgi-bin/zsh
http://my.ip.Net1.Host1/cgi-bin/www-sql
http://my.ip.Net1.Host1/cgi-bin/www-sql
http://my.ip.Net1.Host1/cgi-bin/wwwboard.pl
http://my.ip.Net1.Host1/cgi-bin/wwwadmin.pl
http://my.ip.Net1.Host1/cgi-bin/wrap.cgi
```

```
http://my.ip.Net1.Host1/cgi-bin/wrap.cgi
http://my.ip.Net1.Host1/cgi-bin/wrap
http://my.ip.Net1.Host1/cgi-bin/whois_raw.cgi?fqdn=%0Acat%20/etc/passwd
http://my.ip.Net1.Host1/cgi-bin/whois_raw.cgi
http://my.ip.Net1.Host1/cgi-bin/websendmail
…

Portscan:
79
25
34567
79
137
79
79
23
137
111
445
139
…
```

Obviously, a very noisy and invasive port scan and cgi scan.

Attack Mechanism
Initially sounded like quite an intensive attack.

Others on the Intrusions mailing list mentioned Nessus quickly. Once this was suggested, the pattern immediately became clear (ping, port scan, specific vulnerability checks), having used Nessus frequently. The delay in recognizing it highlights some of the differences in perspectives you have to adjust to when using an IDS instead of assessing.

The three hour delay between initial ping and portscan indicates the first ping may have been a general ping sweep reconnaissance probe, and not part of the Nessus scan, as the initial ping used by Nessus would normally be followed immediately by the full port scan once the target was determined to be responsive.

Chances are port 80 was found open during port scans, which is why the CGI probes were noticed in particular. Either the attacker was focused solely on web vulnerabilities within those supported NESSUS or port 80 was the main port open. Nessus employs some optimizations to only perform detailed vulnerabilities probes based on the availability of prerequisite ports being found open during the initial port scan. Had other ports, such as SMTP or FTP, been found open and the associated modules within Nessus enabled, the detect probably would not have been described as a web or CGI attack.

Note that default nessus behavior is to tell nmap to run a sequential scan, rather than nmap's default randomized scan. However, the attacker may well have just specified a random scan order in this case.

Correlations

Several follow-up emails
http://www.incidents.org/archives/intrusions/msg03529.html
http://www.incidents.org/archives/intrusions/msg03530.html

Evidence of Active Targeting

Likely – ping, scan, and then follow-up vulnerability scan. Although Nessus can scan multiple hosts in parallel, it is not geared toward large-scale bulk scanning. It is more often used as a follow-up tool to more closely analyze a target of interest.

Severity

Using the formula:
Severity = (Criticality + Lethality) – (System + Network Countermeasures)

We determine the following characteristics:
Criticality: 3 (unknown, so go average)
Lethality: 1 (scan)
System Countermeasures: 3 (again, environment is unknown, but will take a conservative stance)
Network Countermeasures: 3 (IDS in place to detect, and it appears that only port 80 was exposed. Presumably this is a web server that is supposed to be running this port, and so the host appears to either be tightly locked down or a reasonably tight firewall is in place)

Result:
Severity = (3+1) – (3+3) = -2

Defensive Recommendations

As this was a scan rather than an outright attack, defensive recommendations focus largely around ensuring that the amount of useful information returned by the scan is minimized, and that visible services are protected against known attacks through careful configuration and maintenance of patches.

Screening unnecessary ICMP traffic, including inbound pings in particular, would go a long way toward discouraging casual attackers searching for targets of opportunity. By default, Nessus will ignore hosts that do not respond to a ping (either standard ICMP ping or nmap TCP / UDP ping), or at least block for long enough, most casual attackers will give up and move on. Similar automated or casual attacks that first confirm a host's availability through the use of pings will also be discouraged.

As usual, any unnecessary services should be disabled. Access to services from the Internet should be restricted to the minimum requirements by a well-maintained firewall. Adequate logging and monitoring to detect scans and attacks should be enabled (as is apparently the case in this example).

The applications in use should be configured according to security best practices. All relevant security patches should be applied, and an efficient, routine patching regimen established to ensure future patches are applied in a controlled but timely manner.

Multiple Choice Test Question

The typical pattern of events for a standard Nessus scan is:

A) port scan, vulnerability probe
B) ping, vulnerability probe, port scan
C) vulnerability probe using all enabled modules

Answer: A)

Detect 3: FTP Warez Probe

Source of Trace

Andrew Daviel posted this trace on January 21, 2002, from his FTP server logs on the incidents.org Intrusions mailing list. The message is archived at: http://www.incidents.org/archives/intrusions/msg03438.html.

Detect Generated By

WU-FTPD FTP server logs.

Probability the Source Was Spoofed

Unlikely. This is a full application-level exchange, and so the TCP handshake has been completed. A full user-level login and subsequent commands have occurred.

Description of Attack

An anonymous FTP session was established:

```
220 obfusc.my.org FTP server (Version wu-2.etc.) ready.
USER anonymous
331 Guest login ok, send your complete e-mail address as
password.
PASS guest@here.com
230-Greetings !
```

Following login, change directory commands for several common directories were issued. These included “/_vti_pvt/”, “/upload/”, “/home/”, “/public/”, “/pub/”, and “/incoming/”. For example:

```
CWD /_vti_pvt/
550 /_vti_pvt/: No such file or directory.
CWD /upload/
550 /upload/: No such file or directory.
…
CWD /pub/
250-This is /pub, the public directory.
…
CWD /incoming/
550 /incoming/: No such file or directory.
…
CWD /pub/incoming/
550 /pub/incoming/: No such file or directory.
CWD /public/incoming/
550 /public/incoming/: No such file or directory.
…
```

Most of these attempts resulted in 550 errors, indicating the requested directory did not exist. The exception to this was the request for “/pub/”, highlighted above. Immediately following this request, an attempt was made to create a new directory:

```
MKD 010118235653p
550 010118235653p: Permission denied on server. (Upload dirs)
```

Fortunately, this request was denied, and the scan continued checking various common directories as described above. Eventually the session was terminated abruptly:

```
CWD / /
550 / /: No such file or directory.
221 You could at least say goodbye.
```

Attack Mechanism

The attack pattern appears to be an FTP scanner of some sort. The long series of directories tested, with no alternate commands or typos, would seem to be scripted activity, regardless of the other indications present. The general intent appears to be to search for a variety of common directories, and test for the ability to write to them (thus the create directory attempt when the "/pub/" was located).

The scan signature is specifically indicative a tool known as "Grim's Ping" (http://grimsping.cjb.net) was used to carry out the attack. This is a well-known tool used to scan for publicly-writeable FTP servers. Such servers are frequently used as warez servers, or drop-off and exchange points for illegal, cracked, and other underground software. The email address used during the anonymous login ("guest@here.com") is a signature of older versions of Grim's Ping. Newer versions typically use an address such as Hgpuser@home.com (specifically, ?gpuser@home.com, with "?" being replaced by an uppercase letter). Another signature of the tool is the directory name used in the create directory command, which is based on a date/time stamp from the client computer (in this case, the year appears to be set incorrectly).

Correlations

FTP scans of this nature are quite common on the Intrusions mailing list. Many are attributed to sites coming from Wanadoo Interactive, an ISP associated with France Telecom. Correlations and further analysis of the original report presented above include messages archived at the following URLs:
http://www.incidents.org/archives/intrusions/msg03440.html
http://www.incidents.org/archives/intrusions/msg03441.html
http://www.incidents.org/archives/intrusions/msg03445.html
http://www.incidents.org/archives/intrusions/msg03098.html

The HoneyNet Project has posted a similar scan as their "Scan of the Month #7" in Novfember. The pattern and final analysis point to the same source, and are available at:
http://project.honeynet.org/scans/arch/scan8.txt

Evidence of Active Targeting

Although not reported within the posting, it is very likely that initial reconnaissance using some of the widespread FTP port scans being currently recorded identified the active FTP server that was attacked. This is speculation, but based on general activity profiles and logical attack behavior.

Bulk scanning for active FTP servers and subsequent follow-up with application-level probes is an easily automated activity. It is unlikely that a malicious party specifically targeted Andrew's FTP server. More likely, it was identified, perhaps automatically, as an active server and tested as a target of opportunity.

Severity
Using the formula:
Severity = (Criticality + Lethality) – (System + Network Countermeasures)

We determine the following characteristics:
Criticality: 4 (network server)
Lethality: 3 (read/write access could be gained to the server)
System Countermeasures: 5 (no writeable directories found)
Network Countermeasures: 2 (IDS in place to detect, but external FTP traffic allowed in)

Result:
Severity = (4+3) – (5+3) = 0

Defensive Recommendations
The attack was unsuccessful, as no writeable directories were identified. Current defenses appear to be adequate. However, the risk of a successful future attack may be further reduced by disabling anonymous access and/or protecting the FTP server in question from general Internet access at the firewall if these capabilities are not specifically required.

Multiple Choice Test Question
Which return code from the FTP server indicates a successful directory change?

A) 404
B) 250
C) 550
D) –1

Answer: B)

Detect 4:

Source of Trace

Posted by Jim Howard to the Intrusions mailing list, with the subject “a strange probe – can anyone identify”. This is a new message at the time of writing, and has not been stored on the mailing list archive (http://www.incidents.org/archives/intrusions/date1.html) yet.

Detect Generated By

Snort scan log. See “Intrusion Detection Snort Style” manual from GCIA training notes, or the Snort User’s Manual.

Probability the Source Was Spoofed

Unlikely, but see below.

Description of Attack

A series of data with very unusual flag combinations was noted. The pattern involves a remote user accessing a local web server using a standard port combination (>1024 on client side to port 80), followed almost immediately by a second connection between the two hosts. In the second connection, the source port is always 18245, and the destination 21536. Various obviously invalid flag combinations are present in the second connection, and the pattern involves different source hosts on the same subnet:

```
Feb 16 10:21:27 65.128.60.87:1565 -> xx.xx.xx.xx:80 SYN ******S*
Feb 16 10:21:28 65.128.60.87:18245 -> xx.xx.xx.xx:21536 NOACK *2U*PR*F
Feb 16 10:22:51 65.128.60.87:1566 -> xx.xx.xx.xx:80 SYN ******S*
Feb 16 10:22:52 65.128.60.87:18245 -> xx.xx.xx.xx:21536 NOACK *2U*PR**
Feb 16 10:22:52 65.128.60.87:18245 -> xx.xx.xx.xx:21536 INVALIDACK *2*APRSF
Feb 16 10:26:25 65.128.60.87:1570 -> xx.xx.xx.xx:80 SYN ******S*
Feb 16 10:26:25 65.128.60.87:18245 -> xx.xx.xx.xx:21536 NOACK **U*PRS*
Feb 18 08:29:46 65.128.60.64:1115 -> xx.xx.xx.xx:80 SYN ******S*
Feb 18 08:29:46 65.128.60.64:18245 -> xx.xx.xx.xx:21536 NOACK *2U*PR*F
```

Attack Mechanism

Mike Poor (p00r0ne@digitz.org) sent a followup email and indicated that this a problem that has been noted before, and specifically identified Paul Ritchey’s GCIA practical, which provided a detailed analysis of the activity. To summarize, the issue appears to be with malfunctioning network equipment at or near the source. The device is dropping the IP and TCP header data for the subsequent traffic, and so the datagram begins directly at the start of the TCP payload. The unusual flag combinations are actually caused by a binary interpretation of the ASCII payload. Based on Paul Ritchey’s practical, an ASCII decode of the IP and TCP header reveals a fairly typical web request (packet capture and decode quoted from Paul’s practical):

```
04:13:58.916034 209.254.130.32.18245 > AAA.BBB.CCC.DDD.21536
                         4500 01c1 7a33 4000 7006 93d2 d1fe 8220
                         ceb5 d85c 4745 5420 2f75 6262 2f46 6f72
                         756d 372f 4854 4d4c 2f30 3030 3332 362e
                         6874 6d6c 2048
```

```
GET /ubb/Forum7/HTDL/000326.html H
```

Correlations

Mike Poor's follow up email (again, not available to link to in the archives for some reason – there appears to be a several-day lag in archive availability).

This type of behavior was also discussed during the IDS Signatures and Analysis section of the GCIA course with Stephen Northcutt.

Paul Ritchey's practical: http://www.giac.org/practical/Paul_Ritchey_GCIA.doc

One of the earlier discussions of this pattern referenced by both Paul and Mike is: http://archives.unixtech.be/arch055/0229.html, which speculates that the device may be a Nortel CVX.

Evidence of Active Targeting

At first glance, the anomalous behavior appears to be a direct result of the initial web browsing, with the timing being indicative of an automated response. The analysis indicates that this is in fact benign traffic.

Severity

N/A – benign false positive

Defensive Recommendations

N/A

Multiple Choice Test Question

This situation can be characterized by:

A) traffic to a web server from a client using a source port in the 1560 – 1570 range
B) A flag combination of *2U*PR** for traffic immediately following a SYN to port 80 between two hosts
C) Source port of 18245 going to destination port 21536 traffic immediately following a SYN to port 80 between two hosts

Answer: C)

Detect 5:

Source of Trace

John Sage (jsage@finchhaven.com) recently announced the availability of his logs to the Intrusions mailing list. These logs are being posted to his web server, http://www.finchhaven.com/pages/incidents/. In some cases these are analyzed and annotated, while in others they are relatively untouched.

Investigating this site (as it was a convenience instead of posting on the Intrusions list directly), I came across this detect at http://www.finchhaven.com/pages/incidents/021802_1021.html. John was unclear what the traffic was, and so it seemed an excellent trace to analyze.

Detect Generated By

Snort IDS and Linux IPChains firewall logs.

Probability the Source Was Spoofed

The traffic is UDP based, which is generally a good candidate for spoofing. As it appears to be reconnaissance activity, however, it is probably not spoofed, as the attacker would want to see the response traffic.

Description of Attack

A UDP connection to port 6767 on John's system triggered a generic Snort signature "UDP to range 1026-6099". The trace data provided includes:

```
Feb 18 10:21:01 greatwall snort: [1:0:0] UDP to range 1026-60999 {UDP}
 139.92.138.146:1133+-> 12.82.142.34:6767

Feb 18 10:21:01 greatwall kernel: Packet log: input DENY ppp0 PROTO=17
 139.92.138.146:1133+12.82.142.34:6767 L=32 S=0x00 I=64417 F=0x0000 T=112
(#76)
```

Attack Mechanism

The traffic was dropped by the firewall, but is obviously a connection attempt to port 6767 UDP. Lacking input from other IP addresses, it is difficult to say how widespread the pattern is.

An initial search for port 6767 identifies BMC-perf-agent, a system monitoring agent from BMC Software (http://www.bmc.com). I had initially thought this might be similar to out-of-control network discovery activity I've heard ascribed to HP OpenView, a somewhat similar product. Apparently if the user is not careful in configuring the parameters for OpenView's auto-discovery mode, its probes can quickly traverse outside the local network and begin scanning large areas of the Internet.

A more detailed port search using some of the Trojan port lists revealed a more likely explanation. These lists (see Correlations) attribute port 6767 to the "UandMe Trojan" and / or "NT Remote Control". A further search did not reveal much additional information about these Trojans.

At this point, trolling for Trojans or perhaps a simple stray packet appear to be the most likely causes (I tend to think the former, but it could still be benign).

Correlations

UandMe Trojan: http://www.simovits.com/trojans/tr_data/y1825.html
"The Trojan List" http://www.simovits.com/sve/nyhetsarkiv/1999/nyheter9902.html
"Treachery Unlimited Port Lookup Utility" http://www.treachery.net/tools/ports/lookup.cgi

NT Remote Control:
"Trojan Ports Defined by KGB" http://www.textfiles.com/uploads/trojanports.txt

Evidence of Active Targeting

Unknown. Little additional information is available, but it seems to be a general probe of some sort, and is not felt to be a targeted activity.

Severity

Using the formula:
Severity = (Criticality + Lethality) – (System + Network Countermeasures)

We determine the following characteristics:
Criticality: 3 (small personal network, generally unknown)
Lethality: 1 (appears to be a simple probe or misdial)
System Countermeasures: 4 (unknown – didn't reach the host, but given the description of the environment, etc., a Trojan infection seems unlikely)
Network Countermeasures: 5 (IDS in place to detect, firewall blocked)

Result:
Severity = (3+1) – (4+5) = -5

Defensive Recommendations

None. Defenses dropped the activity.

Multiple Choice Test Question

The Snort rule used to detect the attack, based on the information available, is most likely to:

A) Alert on very specific attacks
B) False negative frequently
C) False positive regularly

Answer: C) This appears to be a general "log all high UDP traffic" rule, which would produce many false positives on anything but a very quiet network.

Assignment 3 – "Analyze This" Scenario

Executive Summary

In general, there was a great deal of activity observed that can be classified as either inappropriate or outright malicious. Inappropriate or wasteful activity included:

- chat and instant messaging traffic (incoming and outgoing);
- various peer-to-peer file sharing applications; and
- network gaming.

Malicious activity took the form of things such as:

- a wide variety of incoming and outgoing scans;
- network and host mapping activity;
- possible Denial of Service (DOS) traffic; and
- active exploit attempts.

Additionally, concerns about the current deployment and effectiveness of the security infrastructure (perimeter security, technical guidelines, and IDS configuration) were also noted.

Data Set Analyzed

Data files for December 24 through December 28, 2001, were examined (inclusive). Specifically, this involved the following files:

Alerts:

- Alert.011224.gz
- Alert.011225.gz
- Alert.011226.gz
- Alert.011227.gz
- Alert.011228.gz

Scans:

- Scans.011224.gz
- Scans.011225.gz
- Scans.011226.gz
- Scans.011227.gz
- Scans.011228.gz

OOS:

- Oos_Dec.24.2001.gz
- Oos_Dec.25.2001.gz
- Oos_Dec.26.2001.gz

- Oos_Dec.27.2001.gz
- Oos_Dec.28.2000.gz[1]

List of Detects

Top 20 Alerts by Frequency

The intent of this subsection is to present the top alerts, prioritized based on their frequency. The next subsection highlights several additional alerts that, although less frequent, were of particular concern. Specific alert descriptions and commentary for both sets of alerts are included in the section following that.

Although there are a number of different alerts that occurred within the five day period reviewed, it can be noted that the top three or four alert types account for the majority of all events within the top twenty presented here, and more than half of the alerts overall (there were a total of 256,676 alerts noted in total). The sheer volume of alerts within such a brief time period is cause for concern, and is indicative of serious security concerns within the environment, as well as a likely requirement to further refine and tune the IDS rule set in use to minimize the number of false positives.

Many of the more common alerts in the list below appear to be quite innocuous at first. However, when the volumes of alerts are taken into consideration, the trends are indicative that the activity is out of proportion to what might be considered "normal" traffic patterns. Taken in that context, there appears to be a great deal of malicious traffic, largely taking the form of scanning and reconnaissance activity (DNS scanning and/or zone transfer attempts, proxy scans, queso fingerprints, administratively prohibited messages, syn-fin scans, nmap or hping2 activity, and Windows NULL session scanning, for example). A number of other alerts are also suggestive that there may be questions of inappropriate or excessive use of resources (MSN IM Chat and ICMP source quenches, for example).

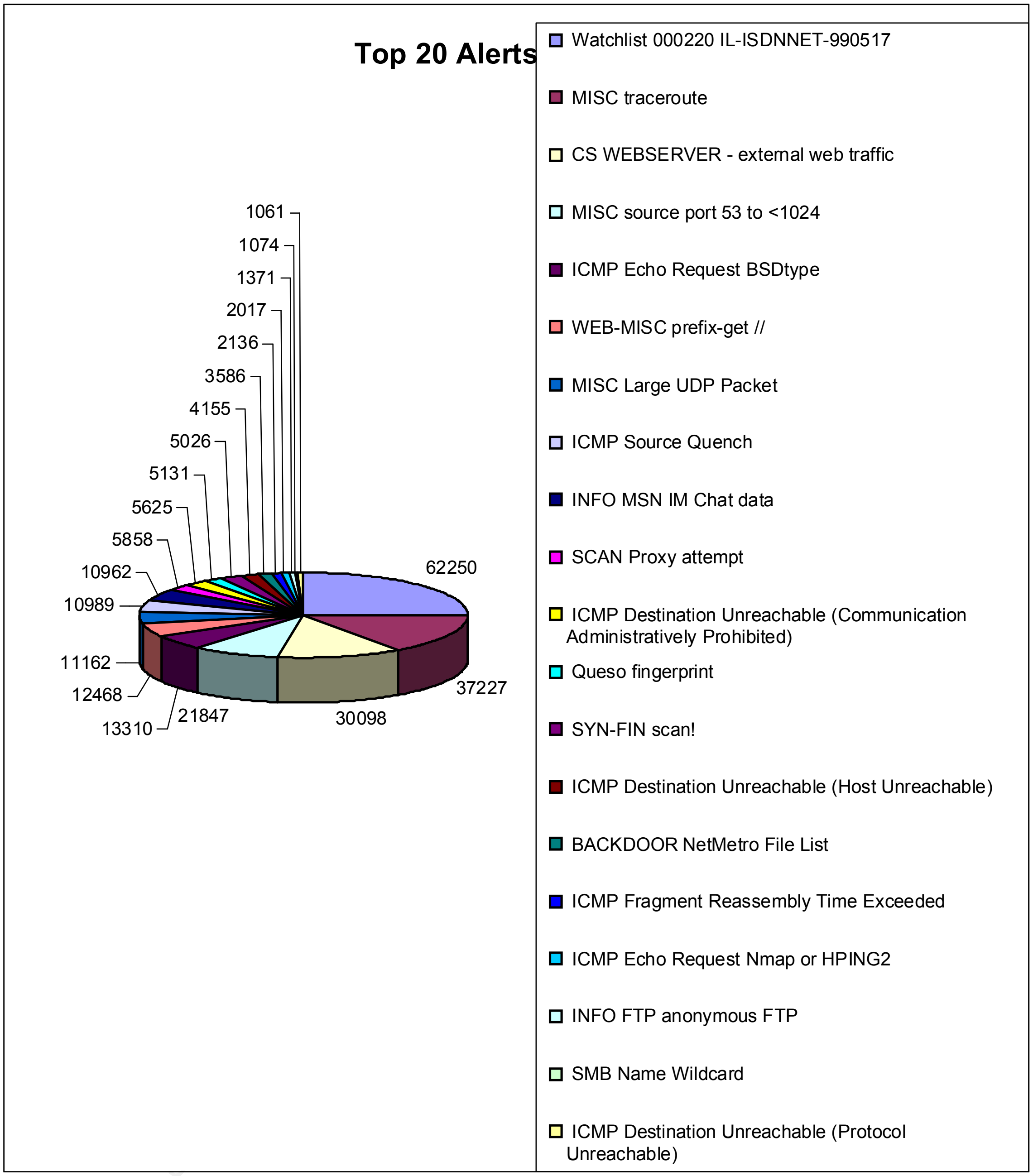
The following table and graph illustrate the top 20 alerts in terms of frequency:

Count	Alert
62250	Watchlist 000220 IL-ISDNNET-990517
37227	MISC traceroute

[1] NOTE – There was no OOS file named "oos_Dec.28.2001.gz". However, the files "oos_dec.28.2000.gz" and "oos_Dec.29.2000.gz" both had modification dates for December of 2001 (and specific days that would indicate they were the files for Dec 28 and 29, 2001, rather than 2000). It is assumed that these are in fact the correct files for 2001, and there was a naming issue in the script that generated them. In any case, the file was empty, and so did not affect the overall results.

30098	CS WEBSERVER - external web traffic
21847	MISC source port 53 to <1024
13310	ICMP Echo Request BSDtype
12468	WEB-MISC prefix-get //
11162	MISC Large UDP Packet
10989	ICMP Source Quench
10962	INFO MSN IM Chat data
5858	SCAN Proxy attempt
5625	ICMP Destination Unreachable (Communication Administratively Prohibited)
5131	Queso fingerprint
5026	SYN-FIN scan!
4155	ICMP Destination Unreachable (Host Unreachable)
3586	BACKDOOR NetMetro File List
2136	ICMP Fragment Reassembly Time Exceeded
2017	ICMP Echo Request Nmap or HPING2
1371	INFO FTP anonymous FTP
1074	SMB Name Wildcard
1061	ICMP Destination Unreachable (Protocol Unreachable)

Top 20 Alerts by Frequency



Other Alerts of Concern

The following table presents a list of additional, less frequent alerts considered by the analyst to be of concern because of their severity or potential implications:

Count	Alert

740	WEB-MISC Attempt to execute cmd
70	EXPLOIT x86 {NOOP, stealth noop, setgid 0, setuid 0}
64	TFTP - Internal TCP connection to external tftp server
22	SNMP public access
8	X11 Outgoing
7	IDS50/trojan_trojan-active-subseven
1	EXPLOIT NTPDX buffer overflow

Additional Alerts of Concern

Descriptions of Alerts

The following table describes the various alerts identified in the two previous subsections (top alerts by frequency and by severity):

Alert	Description

Watchlist 000220 IL-ISDNNET-990517	This custom alert is a general alert that triggers on traffic coming from the netblock 212.179.0.0/16 (IL-ISDNNET-99051). Such signatures are generally intended to keep a close watch on traffic from networks having a history of malicious or unacceptable behavior. The network in question is in Israel.
	
	A brief survey of examples of this alert show a mix of traffic. Some of the traffic appears to be related to Kazaa on MY.NET.70.70 and other internal hosts:
	`Watchlist 000220 IL-ISDNNET-990517 212.179.35.118:60339 -> MY.NET.70.70:1214`
	
	Other alerts show a relationship between some of the "CS WEBSERVER – external web traffic" alerts and this watch list:
	`CS WEBSERVER - external web traffic 212.179.79.2:32282 -> MY.NET.100.165:80`
	`Watchlist 000220 IL-ISDNNET-990517 212.179.79.2:32280 -> MY.NET.100.165:80`
	
	Finally, some of the traffic appears to be possible benign return traffic from internal users' web browsing (pending further information from full packet traces):
	`Watchlist 000220 IL-ISDNNET-990517 212.179.27.164:80 -> MY.NET.98.152:2399`

MISC traceroute	This signature appears to be designed to track traceroute activity. Traceroute is commonly used as a network diagnostic tool, but can also be used by attackers to perform network mapping. The volume of traceroute traffic detected is troubling, as it is far too high for normal

	behavior. A great deal of it is focused on MY.NET.140.9 for some reason, although from a variety of different hosts. Other destination hosts are also involved. The reason for this volume remains unclear. Distributed across a larger selection of hosts, it might be assumed to be particularly heavy network mapping, but the volume of alerts make even this questionable. More likely causes are either a poorly written signature that is generating a large number of false positives, and/or anomalous traffic that is triggering this signature as a side-effect of some sort. Some indications on the Snort-Users mailing list (see Correlations) are that this signature frequently false-positives. Further investigation utilizing the signature source and/or packet traces, is warranted. Sample alert: `MISC traceroute 130.132.252.244:42671 -> MY.NET.140.9:33462`
CS WEBSERVER - external web traffic	Based on inference, alert data, and discussion of this particular signature on the SANS Forum,[2] this custom signature appears to trigger when external addresses access port 80 of MY.NET.100.165. Presumably, this is to alert on outsiders trying to access a web server intended for internal use only. Example alerts include: `CS WEBSERVER - external web traffic 202.56.245.46:62456 -> MY.NET.100.165:80` The signature appears to have triggered for a variety of different source addresses, based on brief inspection of the alert data.
MISC source port 53 to <1024	This is a standard rule that triggers on TCP traffic from source port 53 going to a reserved port (< 1024). The TCP requirement of the signature makes this significant, as such traffic would normally be indicative of either someone attempting a DNS zone transfer from an outside network in order to obtain reconnaissance data, or someone trying to bypass poorly configured simple packet filters using a trusted source port. Most of the alerts of this type reviewed are from port 53 to 53, and so are likely zone transfer requests or DNS server mapping attempts.

[2] http://forum.sans.org/discus/messages/78/1716.html?1012743539

	Note that there appears to be some conflict in the rule format. A review of discussions on the Snort-Users mailing list regarding this signature show copies of the signature using UDP as the protocol, while the current signature description page at www.snort.org indicates TCP. The UDP based signature appears to be notorious for false positives on legitimate traffic, while the TCP version is what is described above. Sample Alert: `MISC source port 53 to <1024 195.145.16.194:53 -> MY.NET.1.4:53`
ICMP Echo Request BSDtype	This is a standard alert on incoming Echo Requests (pings) that appear to have been generated by a BSD-based system, due to the characteristic incrementing data pattern within the body of the packet. Given the apparently open nature of the university's network, incoming ping traffic is likely fairly common. In small quantities, this type of traffic may well be considered fairly benign. It can be indicative of several types of malicious activity, especially given the volume of pings that are occurring for the alert to be ranked as high as it is. Malicious uses can include: • Reconnaissance • Covert channels • Denial-of-Service Tracking such traffic in terms of volumes over time, sources, and destinations can help to clarify its nature. Further analysis would require packet traces to check for signs of covert channels. Example alert: `ICMP Echo Request BSDtype 128.223.4.21 -> MY.NET.70.148`
WEB-MISC prefix-get //	This standard alert triggers when a double-slash "//" is embedded within a GET request. Little documentation on this rule is available, but it appears to attempt to exploit an information exposure vulnerability in some web servers. Presumably, particular web servers reveal more information than they should in response to such a request, probably in the form of an error message. Sample alert:

	WEB-MISC prefix-get // [**] 64.110.96.191:21072 -> MY.NET.253.114:80
MISC Large UDP Packet	This signature triggers on UDP packets having a size greater than 4000 bytes. Some of the discussions found relating to this signature indicate it may be quite common on Windows based networks for ports 137 and 138, but checks of the alert logs seem to indicate other ports in use. Common ports noted include 888 (CDDB or AccessBuilder?), 27005 (gaming traffic – quite possible), 0 (anomalous): MISC Large UDP Packet [**] 66.190.93.40:0 -> MY.NET.87.50:0 MISC Large UDP Packet [**] 66.190.93.40:16638 -> MY.NET.87.50:888 MISC Large UDP Packet [**] 24.76.1.122:27005 -> MY.NET.87.50:888
ICMP Source Quench	Standard rule. Triggers when an external address sends and ICMP Source Quench message to an internal host, indicating that the internal host should throttle its transmissions. Could be a sign of an overloaded external server and/or high bandwidth use by an internal host. Some possibility it could be used to artificially slow down an internal host (DOS). There are some indications this type of message can also be used in host OS fingerprinting if elicited. MY.NET.5.13 is a common source in these alerts. This may be indicative of a capacity / processing issue on this host, or unusually high traffic. Samples: ICMP Source Quench [**] MY.NET.5.13 -> MY.NET.200.20 ICMP Source Quench [**] MY.NET.5.13 -> MY.NET.200.12 ICMP Source Quench [**] MY.NET.5.13 -> MY.NET.200.23
INFO MSN IM Chat data	This is likely similar to the standard "INFO MSN chat access" signature. The intent is to alert on the use of Microsoft's Instant Messenger / Chat application. This signature appears to trigger on traffic to or from port 1863 ("MSNP"), used for this application. A variety of security issues have been identified with various instant messaging or chat applications. The signature seems to be intended to track usage of these programs. Sample: INFO MSN IM Chat data [**] 64.4.12.153:1863 ->

	MY.NET.98.115:1078
SCAN Proxy attempt	This signature detects connection attempts to commonly used web proxy ports (1080, 8080, etc.). Misconfigured proxies can be used to "anonymize" and redirect external web activity (normally they should only proxy requests from internal networks; by handling requests from external networks, the requesting party is hidden from the end receiver.) In some cases, misconfigured proxies may also be used to forward external requests into the [otherwise protected] internal network. These are an extremely common form of scan. Alert example: `SCAN Proxy attempt [**] 24.182.147.53:1783 -> MY.NET.253.105:8080`
ICMP Destination Unreachable (Communication Administratively Prohibited)	This signature triggers on any occurrence of an ICMP Destination Unreachable message with a subtype of "Communication Administratively Prohibited." Such traffic is the result of an unauthorized attempt to access a port or host that is being protected by a router or firewall. The filtering device generates this error code in response to an attempt to access a host and port that are being protected. It is indicative that the receiver of the message may be engaged in unauthorized activity of some sort, and should be further investigated. In this example, it would appear that MY.NET.140.9 tried to access an protected resource: `ICMP Destination Unreachable (Communication Administratively Prohibited) [**] 192.80.43.21 -> MY.NET.140.9`
Queso fingerprint	Queso is a well-known scanning tool that features the ability to fingerprint, or identify, the operating system of its targets. This is done by sending various unusual or "illegal" packets, particularly with odd flags, options, or other parameters, in order to elicit various identifying error responses from the target. This signature indicates that these anomalous packets have been spotted, and so active scanning of the destination is underway. Such alerts should be followed up. `Queso fingerprint [**] 216.52.244.143:33007 -> MY.NET.253.41:25`
SYN-FIN scan!	This is a custom signature written to detect SYN-FIN scans - scans in which both the TCP SYN and FIN flags are set

	simultaneously. Such a combination is illegal and should not occur. However, in some cases this combination can be used to bypass simple filtering devices, and even elicit legitimate responses (i.e. SYN-ACK) from hosts. This is definitely malicious scanning activity and should be investigated. Example: `SYN-FIN scan! [**] 24.0.28.234:22 -> MY.NET.1.2:22`
ICMP Destination Unreachable (Host Unreachable)	This signature triggers on occurrences of an ICMP Destination Unreachable message with a subtype of "Host Unreachable" being returned from an external host to an internal address. Such traffic is the result of an attempt to access a host is currently (network failure or configuration error) or permanently unavailable (i.e. non-existent / unused address). In and of itself this may not be significant, although a cluster of these messages to a particular host may be indicative of network mapping or scanning activity, and such incidents should be further investigated. This message is useful in performing inverse mapping, in which traffic is sent to various behind a router. If the router returns Host Unreachable messages to some of the requests, but not others, it can be inferred that the hosts for which there was no response are active systems. Sample: `ICMP Destination Unreachable (Host Unreachable) [**] 67.201.0.58 -> MY.NET.60.8`
BACKDOOR NetMetro File List	This is a standard Snort rule that watches for traffic from the internal network to an external host on port 5032, which is associated with the NetMetro backdoor / Trojan. The signature is fairly simplistic, as it looks for a very short pattern in any traffic destined for port 5032. As 5032 is a legitimate random ephemeral port, such a signature tends to have a high false positive rate. Jyri Hovila has written a good description of how a similar NetMetro signature operates (and false positives).[3] Despite the incidence of false positives, the hosts involved should be investigated to ensure they haven't been compromised. Example: `BACKDOOR NetMetro File List [**] MY.NET.60.11:20 -> 209.49.12.32:5032`

[3] http://www.geocrawler.com/archives/3/4890/2001/10/50/6941021/

ICMP Fragment Reassembly Time Exceeded	This signature triggers on occurrences of an ICMP Fragment Reassembly Time Exceeded message being returned from a host. Such traffic is the result of fragmented traffic to the network not being fully received within the timeout period, and so reassembly has been unsuccessful. Under normal situations, particularly those where fragmented traffic is unusual (historically this has been the norm, although VPN and other newer types of traffic have a tendency to generate relatively large numbers of fragments), this type of error should be very rare. Particularly when occurring in large quantities, this type of message is indicative of anomalous, and probably malicious, activity. Various Denial of Service attacks and IDS evasion techniques are associated with different types of fragmented traffic. Attackers can also induce this error message by deliberately not sending some of the required fragments to a host in order to perform host discovery and mapping. Example alert: `ICMP Fragment Reassembly Time Exceeded [**] MY.NET.87.50 -> 213.66.178.20`
ICMP Echo Request Nmap or HPING2	This signature alerts on crafted ICMP echo requests typical of scanning tools including Nmap and HPING2. Evidence of such tools in use should be cause for concern, and further investigation is warranted. This alert is quite possibly the initial indication of a scan getting underway, as hosts being targeted are typically pinged prior to the actual port scan to determine their availability. Example: `ICMP Echo Request Nmap or HPING2 [**] MY.NET.83.16 -> 149.1.1.1`
INFO FTP anonymous FTP	This standard alert is triggered when an anonymous login is detected on the FTP control port (TCP port 21). Anonymous FTP servers are frequently misconfigured and subject to abuse as warez or porn sites. FTP servers in general have a poor security track record, having been subject to a wide variety of exploits. Anonymous FTP servers should normally be carefully planned and identified. This signature should normally be tuned to exclude "official" anonymous FTP servers (unless simple usage tracking is desired), but can identify either unauthorized FTP servers, or attackers trying to probe for

	FTP servers providing anonymous services. Further analysis requires additional information about the results of the login attempts, site policies, and network composition. Example: `INFO FTP anonymous FTP [**] 212.95.76.165:1990 -> MY.NET.53.229:21`
SMB Name Wildcard	This is evidence of an attempt to utilize an SMB NULL session to access a Windows resource. Although common on internal networks, such traffic should not be permitted from external hosts. A great deal of information can be enumerated using SMB NULL sessions (i.e. without requiring a valid username and password), including account names, drive shares, and other resources. Traffic of this nature coming in from the external network, whether accidental or malicious, is a concern and should be filtered. Example alert: `SMB Name Wildcard [**] 198.147.81.246:137 -> MY.NET.132.138:137`
ICMP Destination Unreachable (Protocol Unreachable)	This signature indicates an ICMP Protocol Unreachable message was observed. This message is generated when a device receives a request for a network protocol it does not support or allow. It is very unusual in normal traffic, although various reconnaissance techniques make use of this type of error message during information gathering and mapping activities (i.e. elicit responses from hosts during host mapping). Particularly when seen in significant quantities, these messages should be investigated. MY.NET.5.75 appears to be the source of many of these errors (i.e. generates the error code in response to another host's request). This may be indicative that this address is a network device of some sort (router), or perhaps is the target of scanning. Further information on this host in particular would be desirable. Sample: `ICMP Destination Unreachable (Protocol Unreachable) [**] MY.NET.5.75 -> MY.NET.217.126`
WEB-MISC Attempt to execute cmd	Although this string is named slightly differently from the current standard Snort rule set, it appears to be similar in intent to "WEB-IIS cmd.exe access." This rule triggers on an embedded string of "cmd.exe" within a web request.

	Such requests are used by both live attackers and various IIS worms to execute commands on the web server, either as part of a larger exploit or as the result of a back-door installed by a previous compromise. There is some risk of false positives from activities such as downloading Windows service packs and updates, or similar traffic that might legitimately contain the string "cmd.exe." This type of traffic represents a definite threat. Any indication of a positive response to these requests should be immediately investigated, as it would appear to be a clear sign of a compromised system. Example: `WEB-MISC Attempt to execute cmd [**] 216.141.220.247:3572 -> MY.NET.130.86:80`
EXPLOIT x86 {NOOP, stealth noop, setgid 0, setuid 0}	Although the "EXPLOIT" categorization of these alerts is cause for immediate concern, each of these signatures is a "generic" signature intended to detect patterns that are common to a variety of exploits. NOOP instructions are commonly used to pad buffer overflow attempts, for example.[4] As such, this signature may detect a variety of buffer overflow attacks. Unfortunately, all four of these signatures base their criteria on a fairly limited pattern representing machine code or other encodings of the operations in question. Although relatively rare in ASCII- or text-based data, streams containing relatively random binary values are significantly more likely to randomly match these patterns. Given that these signatures are generally using patterns of no more than five to ten bytes, the chances of a match in a sufficiently large amount of binary data are relatively high. Images and files of a similar [binary] nature are particularly well known to cause false positives in these generic signatures. Web and file-sharing traffic in particular, as

[4] See Aleph1's infamous Phrack 49 article, "Smashing the Stack for Fun and Profit," (http://www.insecure.org/stf/smashstack.txt) where he notes, "One way to increase our chances is to pad the front of our overflow buffer with NOP instructions. … If we are lucky and the return address points anywhere in the string of NOPs, they will just get executed until they reach our code."

	well as ftp-data channels, are examples of potential sources for these alerts that should be examined with a somewhat healthy dose of skepticism, tempered by the general paranoia of a security professional. (i.e. They should be examined carefully as real concerns, but with the recognition that certain types of traffic are more likely to false-positive.)
	
	Further information relating to the events surrounding these alerts would, as usual, aid in analysis. Classification of the event as a stimulus or response, packet dumps, review of application activity, and further knowledge of the hosts involved would all aid in further analysis and improve accuracy.
	.
	These three examples are considered likely false positives based on their apparent web-browsing nature, and are representative of the bulk of the alerts recorded:
	`EXPLOIT x86 NOOP [**] 205.138.230.234:80 -> MY.NET.97.216:1178`
	`EXPLOIT x86 setgid 0 [**] 210.112.41.243:80 -> MY.NET.98.155:1294`
	`EXPLOIT x86 stealth noop [**] 207.199.1.201:80 -> MY.NET.111.223:1293`
	
	In this last case, the ports appear to indicate that this traffic is ICQ-related (although other setuid 0 alerts involved web traffic similar to the above examples). This would seem to be more of a concern, and the existence of buffer overflows in mICQ as documented by SecuriTeam at http://www.securiteam.com/exploits/5AP0P1P35E.html makes this particularly unsettling:
	`EXPLOIT x86 setuid 0 [**] 63.240.202.64:4000 -> MY.NET.97.233:1044`

TFTP - Internal TCP connection to external tftp server	TFTP is a very simple protocol that is notorious for poor security. Aside from its use in managing diskless workstations and network infrastructure devices, TFTP is probably best known for its misuse by attackers as a tool for transporting exploits, tools, and data. Use of TFTP should be carefully controlled, and restricted at the network perimeter. Although connecting to an external TFTP server sounds relatively safe (i.e. low risk to the internal network), it may be indicative that a successful compromise has just

	occurred, and the attacker is now connecting back out to download further tools and set up shop. Worms such as Nimda are an example of one type of attacker that behaves in this way, see "CERT Advisory CA-2001-26 Nimda Worm" at http://www.cert.org/advisories/CA-2001-26.html. Example alert: `TFTP - Internal TCP connection to external tftp server [**] MY.NET.98.115:1643 -> 66.69.147.209:69`
SNMP public access	SNMP is a very useful monitoring and network maintenance tool, but it can be a security nightmare. Default community strings (passwords) are well known and frequently unchanged, allowing anyone to access devices and systems using SNMP. At best, SNMP can be used to extract a great deal of information about a device and its configuration. If write (private) access is enabled and using a well-known community string, it may well be possible to remotely reconfigure and/or directly compromise the device. Additionally, the recent CERT advisory (CERT® Advisory CA-2002-03 "Multiple Vulnerabilities in Many Implementations of the Simple Network Management Protocol (SNMP)", http://www.cert.org/advisories/CA-2002-03.html; CVE entries CAN-2002-0012 and CAN-2002-0013) indicates that the use of SNMP should be very carefully restricted and maintained. "Public" is one of the best known and most common default SNMP community strings. Use of this community string is a likely indication of a poorly configured SNMP agent, or someone probing for such an agent. Devices utilizing this community string should have SNMP disabled if not required. Perimeter security controls should also be reviewed to ensure SNMP is not permitted in from external addresses. If required, all community strings changed to hard to guess values, their overall configuration reviewed (if SNMP wasn't tightened, was anything else missed?), access restricted to designated management stations, and similar best practices followed. The source of the traffic should also be investigated for possible malicious intent. Example: `SNMP public access [**] 10.196.5.19:1341 ->`

	`MY.NET.137.7:161`
X11 Outgoing	As with TFTP, opening an X11 session from a compromised host back to an external platform is fairly common behavior for attackers. Aside from such direct malicious implications, X Windows is not encrypted, and poorly secured X11 servers can be easily compromised using X11 keystroke loggers or screen captures of significant information. This alert indicates that some sort of remote X Windows activity was initiated to run an application on an internal server and display it on a remote host, which is troublesome by its very nature. Any such activity should be considered malicious and investigated, although it may hopefully be a security awareness issue that can be more easily addressed. Perimeter controls should also be considered to block this kind of traffic. Example: `X11 outgoing [**] 24.79.242.21:6000 -> MY.NET.130.73:2357`
IDS50/trojan_trojan-active-subseven	This signature indicates that activity associated with the SubSeven Trojan has been detected. This is a major concern, as SubSeven essentially allows an attacker to fully control the compromised host remotely. SubSeven is currently one of the more common Trojans in use. This activity should be investigated, and if the host has indeed been infected, incident response / recovery procedures initiated. Sample: `IDS50/trojan_trojan-active-subseven [**] MY.NET.70.148:1243 -> 204.152.184.75:56442`
EXPLOIT NTPDX buffer overflow	Although only occurring once, this is a very specific exploit alert. Investigation of the alert data revealed that the required ports (NTP – port 123) were in use, and so this is cause for significant concern. A review of the signature details shows the alert is generated in response to an overly long NTP message. Checking the NTP RFC, "RFC 1305 Network Time Protocol (Version 3) Specification and Implementation" at http://www.faqs.org/rfcs/rfc1305.html, it appears that this signature should be accurate for "basic" NTP traffic, which is of a fixed size. There are some cases where the protocol supports longer messages than what the

	signature expects, however. These are for NTP control codes and/or authentication data. It appears that there is sufficient cause for concern to warrant further investigation. Review and decode of the packet dump involved should be sufficient to confirm or discard the event. Alert text: `EXPLOIT NTPDX buffer overflow [**] 207.46.178.10:123 -> MY.NET.190.14:123`

Alert Descriptions

Top 10 Scan Sources

This subsection begins the analysis of the scan data gathered. Most frequent sources and destinations are identified, and finally scan destination ports and types. As with the alert data, the top three or four participants in each of these categories generally represented the majority of the traffic (i.e. were significantly more active or targeted than the others).

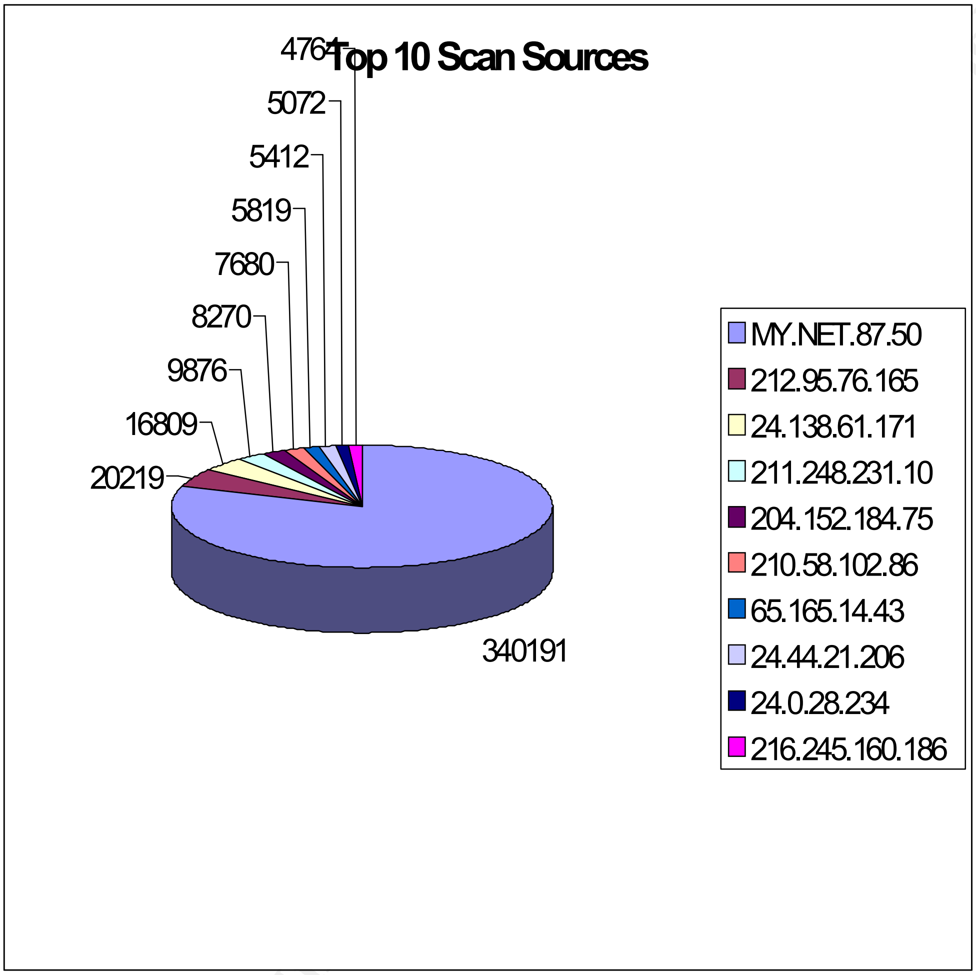
Of particular note when reviewing the top scan sources is the fact that the single most prolific scanner appears to be an internal host, MY.NET.87.50. This is indicative of either a compromised or thoroughly misused (how much of a difference is there?) host. This system should be examined and most likely taken off line for review and cleaning following adequate further investigation.

Several different cable modem subscribers (24.x.y.z) were among the top 10 scan sources, which is consistent with the general reputation these addresses have. Of personal interest is the fact that a quick address lookup for one of the cable modem addresses in question reveals that it is actually my [small] local area (and no, it's not my address).

Count	Source
340191	MY.NET.87.50
20219	212.95.76.165
16809	24.138.61.171
9876	211.248.231.10
8270	204.152.184.75
7680	210.58.102.86
5819	65.165.14.43
5412	24.44.21.206
5072	24.0.28.234

4764	216.245.160.186

Top 10 Scan Sources



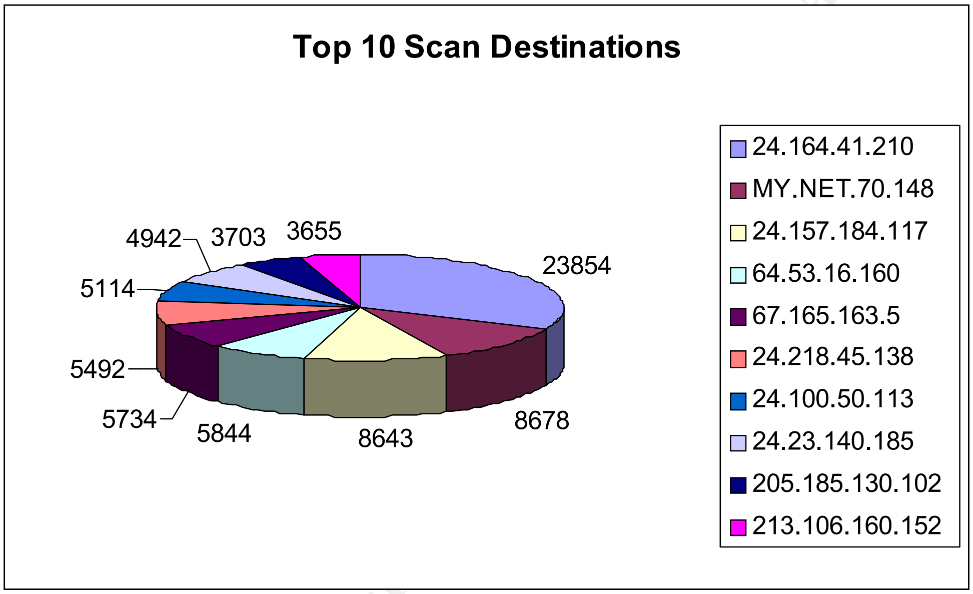
Top 10 Scan Destinations

A review of the top scan destinations show a very large amount of UDP scanning activity. Interestingly, only one of the destinations is an internal host (MY.NET.70.148), indicating that internal hosts (or host, as identified above) have been very active in scanning. Cable modem subscribers using 24.x.y.z addresses are very frequent targets. Although the address blocks involved in the top scan source and destination lists appear to be very similar at first glance, the top 10 in each do not actually overlap.

Count	Destination	Most Common Scan Type
23854	24.164.41.210	UDP

8678	MY.NET.70.148	SYN
8643	24.157.184.117	UDP
5844	64.53.16.160	UDP
5734	67.165.163.5	UDP
5492	24.218.45.138	UDP
5114	24.100.50.113	UDP
4942	24.23.140.185	UDP
3703	205.185.130.102	UDP
3655	213.106.160.152	UDP

Top 10 Scan Destinations



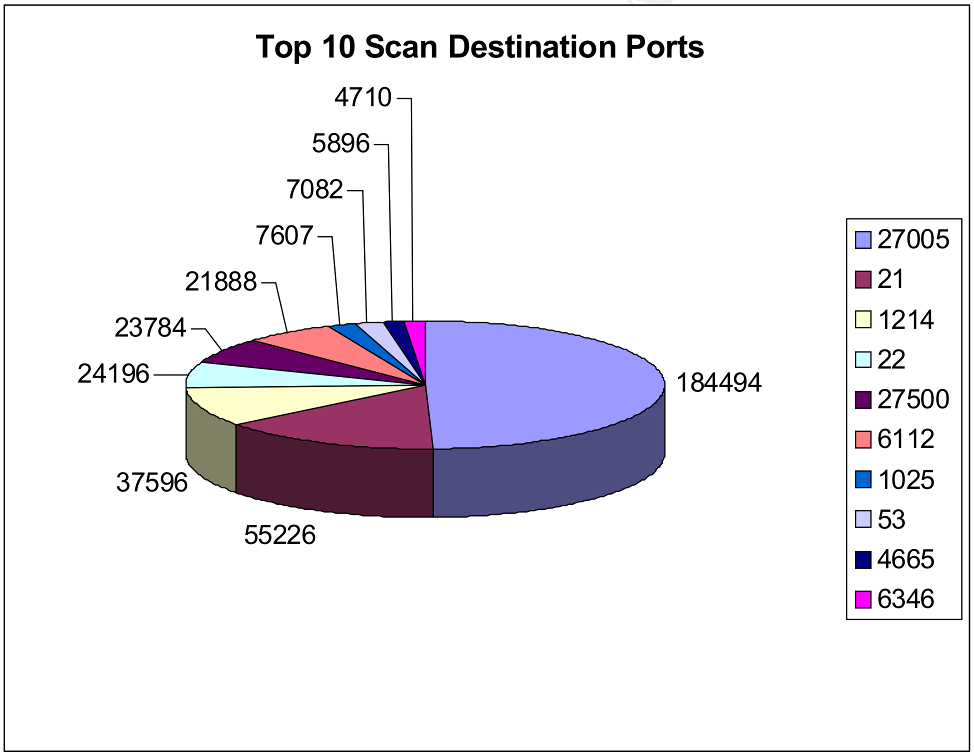
Top 10 Scan Destination Ports

A review of the most frequently targeted destination ports for scans reveals some interesting results. A number of ports on the list are very common mass-scanning destinations, as can be observed at www.incidents.org in the "Top 10 Target Ports" and Intrusions mailing list archives sections. These include FTP, SSH, and DNS in particular, and DTSPC to a lesser extent. Aside from these ports, however, most of the destinations appear to be related to network gaming, file sharing, or Trojans (the standard IANA port lists don't cover most of these ports!).

Count	Port	Protocol	Port Description
184494	27005	UDP	Half-Life client port (FLEX-LM is usually associated with 27005, but that is for TCP – Half-Life client traffic is the only match I could find on

			this port for UDP)
55226	21	TCP	FTP
37596	1214	TCP	KAZAA (file sharing)
24196	22	TCP	SSH
23784	27500	UDP	QUAKEWORLD (game server and/or back door), Elite Force, Alternate Unreal Tournament (i.e. Network Games)
21888	6112	UDP	DTSPC, Battle.Net (Starcraft, etc. – Network Games)
7607	1025	UDP	Blackjack, Trojans: Fraggle Rock, md5 Backdoor, NetSpy, Remote Storm
7082	53	UDP, TCP	DNS (~4200 UDP, ~2900 TCP)
5896	4665	UDP	eDonkey2000 (file sharing)
4710	6346	TCP	Gnutella (file sharing)

Top 10 Scan Destination Ports



Scan Types

A summary of different scan types detected is presented below. UDP scans are by far the most common, but this seems to be questionable based on the author's general observations from various security lists and discussions. Chris

Kuethe's practical notes that he considers UDP and SYN scans to be relatively uninteresting and/or subject to frequent inaccuracies,[5] and has written his scripts (which are among the ones used during this analysis) to ignore such scans by default. It was decided to include them for this paper, as ignoring them completely seemed to discard too much potentially important information. SYN and SYNFIN scans are the next most common, which would seem to be as expected. Other scan types are much more rare.

Count	Type
389414	UDP
141017	SYN
5004	SYNFIN
591	VECNA
168	NULL
57	INVALIDACK
52	NOACK
34	UNKNOWN
21	FIN
5	XMAS
3	NMAPID
3	FULLXMAS

Scan Types

Top 10 OOS Sources

Analysis of the various Out of Specification (OOS) data reveals several things. The top sources are all external addresses. A single source accounts for more than 95% of the overall alerts (7931 alerts for 24.0.28.234 out of a total of 8213 OOS records).

A closer inspection of the traffic from 24.0.28.234 shows that the packets are remarkably similar. All traffic for this IP address occurred on December 25, 2001, between 21:50:46 and 22:12:22 (nearly 8000 records in a 20 minute time period). The basic packet characteristics are largely constant, with little variation (i.e. they are crafted). Specifically, the following characteristics remain constant across all packets:

- Source port: 22
- Destination port: 22
- IP ID: 39426
- Flags: SYN, FIN
- Window Size: 0x404
- TCP Options

[5] http://www.giac.org/practical/chris_kuethe_gcia.html

- TTL

(Some of these parameters will remain constant under normal circumstances, or at least could possibly, but are included for completeness.)

The following fields vary to a certain extent:

- Sequence Number
- Acknowledgement

However, both of these fields remain constant across a number of different alerts (approximately ten to twenty packets in series, or approximately 1 second worth of scanning, have constant sequence and acknowledgement numbers, after which both values change and remain constant for the next series).

The destination of this traffic swept across virtually every subnet, and most hosts within those subnets, of the entire MY.NET.0.0/16 address space. The question arises as to whether all subnets are instrumented with IDS, or possibly screened by a firewall that allows all traffic to other subnets. In a university environment, it is entirely possible that faculty and/or administrative subnets might be screened, while other subnets aren't. Presumably internal firewalls would be used for this, in order to provide protection from internal attacks, etc., but this might still account for the gaps – it might have been possible to only acquire a single perimeter firewall and "make do" with that. Packet loss and other performance issues may also have played a factor. It seems unlikely that these subnets were actually skipped by the attacker, as the detects walk sequentially across large swaths of the entire address space. The target of all of this traffic was port 22, SSH, which has consistently been one of the top scan destinations on www.intrusions.org during this time period (December 2001 through February 2002).

To summarize, the vast majority of the OOS traffic observed was generated by a complete SYN-FIN port scan for SSH across the majority of the entire MY.NET address space, and originating from 24.0.28.234. Packet characteristics (semi-random sequence and acknowledgement numbers, IP ID of 3924, window size of 0x404 or decimal 1028, SYN-FIN scan) are indicative of SynScan (also mentioned above in Assignment 2, Detect 1). Further discussions with Donald Smith, who analyzed SynScan for his GCIA practical (http://www.giac.org/practical/donald_smith_gcia.doc), indicate that this is likely an older version of the tool, as newer ones have semi-randomized the IP ID and TTL as well as the sequence and acknowledgement values. The specific version in use appears to be 1.5 or 1.6 based on these indications. These versions have a static initial TTL of 42, which is a plausible match given the TTL at the destination of 25. The attacker thus appears to be 17 hops away.

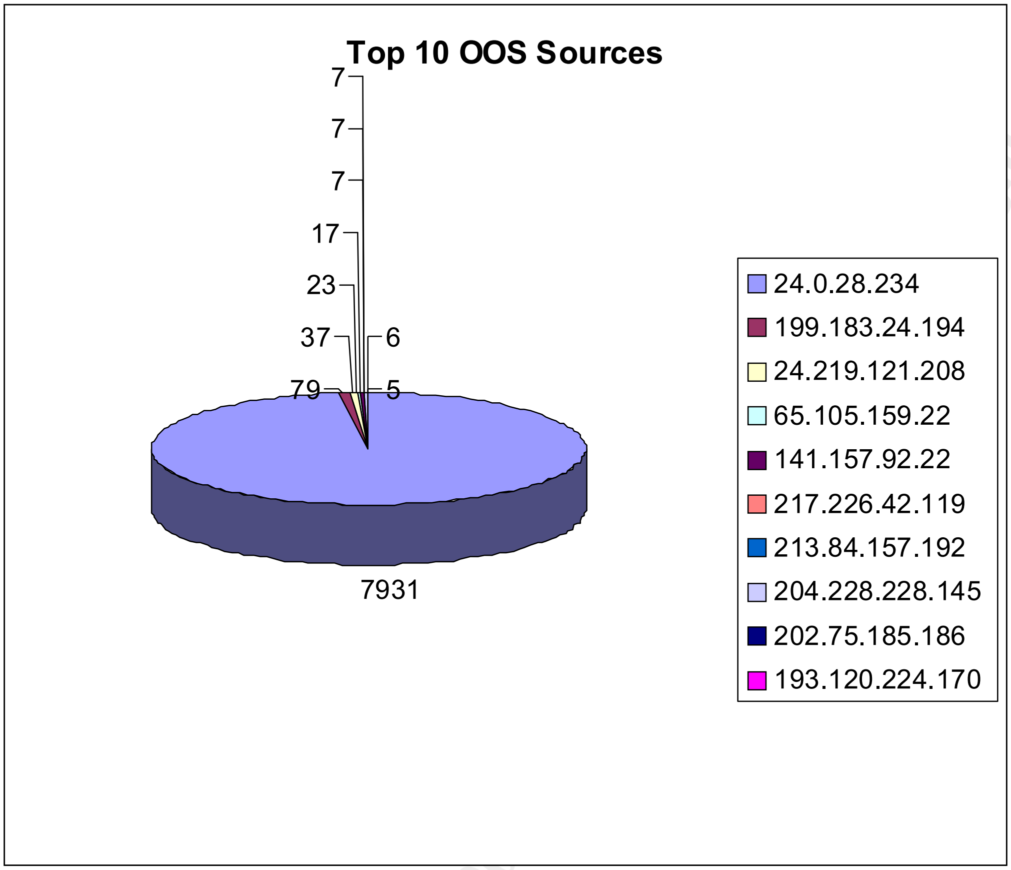
A brief sample of the traffic from 24.0.28.234, showing the transition from one sequence / acknowledgement pair to the next is:

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/25-21:50:46.936960 24.0.28.234:22 -> MY.NET.1.29:22
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x7863007   Ack: 0x6D563A98   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/25-21:50:47.127930 24.0.28.234:22 -> MY.NET.1.38:22
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x7863007   Ack: 0x6D563A98   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/25-21:50:47.240332 24.0.28.234:22 -> MY.NET.1.44:22
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x359223BC   Ack: 0x226ABCA8   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/25-21:50:47.267402 24.0.28.234:22 -> MY.NET.1.45:22
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x359223BC   Ack: 0x226ABCA8   Win: 0x404
00 00 00 00 00 00                                ......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/25-21:50:47.404650 24.0.28.234:22 -> MY.NET.1.52:22
TCP TTL:25 TOS:0x0 ID:39426
**SF**** Seq: 0x359223BC   Ack: 0x226ABCA8   Win: 0x404
00 00 00 00 00 00                                ......
```

The following table and graph summarize the frequency and addresses of the top ten OOS event sources:

Count	Source
7931	24.0.28.234
79	199.183.24.194
37	24.219.121.208
23	65.105.159.22
17	141.157.92.22
7	217.226.42.119
7	213.84.157.192
7	204.228.228.145
6	202.75.185.186
5	193.120.224.170

Top 10 OOS Sources



Top 10 OOS Destinations

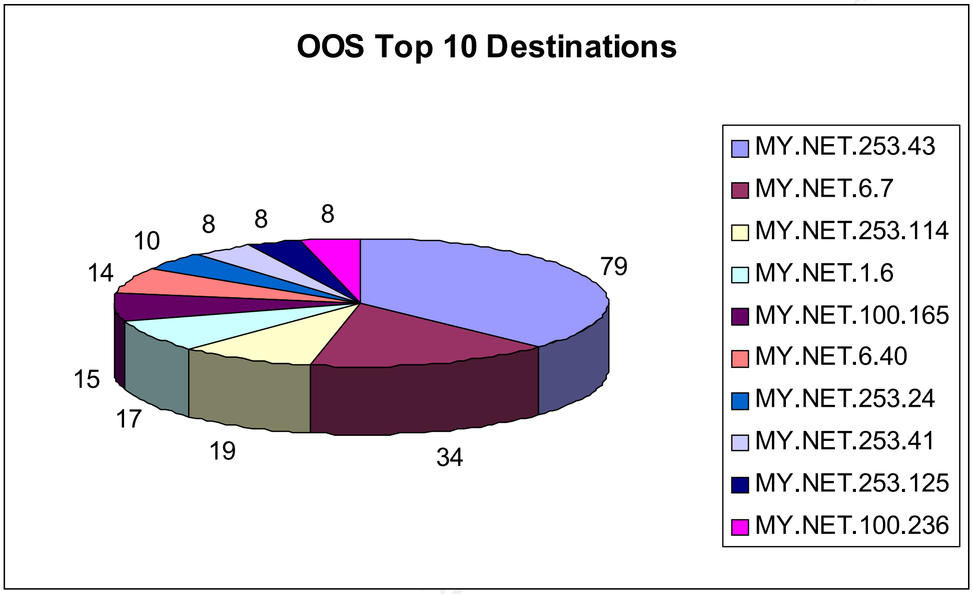
As can be seen from the table, all of the OOS destinations were within the internal network. Interestingly, and perhaps to be expected, many of these top destinations correspond directly to similarly ranked sources in the above table. That is, a review of the data for these particular destinations (excepting traffic from 24.0.28.234, which was a wide sweep across the whole address space) shows that most of the alerts for each destination come from the roughly corresponding entry in the source table. Although not completely accurate, there is a significant one-to-one trend in the data – most of the traffic from any given top source was largely focused on a similarly ranked top destination, and most of the events for that destination originated with the single source (or at least that source is the single most frequent originator). There also seemed to be a trend to focus on one or two specific ports for each destination.

The destination addresses and their frequencies are summarized below:

Count	Destination
79	MY.NET.253.43
34	MY.NET.6.7
19	MY.NET.253.114

17	MY.NET.1.6
15	MY.NET.100.165
14	MY.NET.6.40
10	MY.NET.253.24
8	MY.NET.253.41
8	MY.NET.253.125
8	MY.NET.100.236

Top 10 OOS Destination Addresses



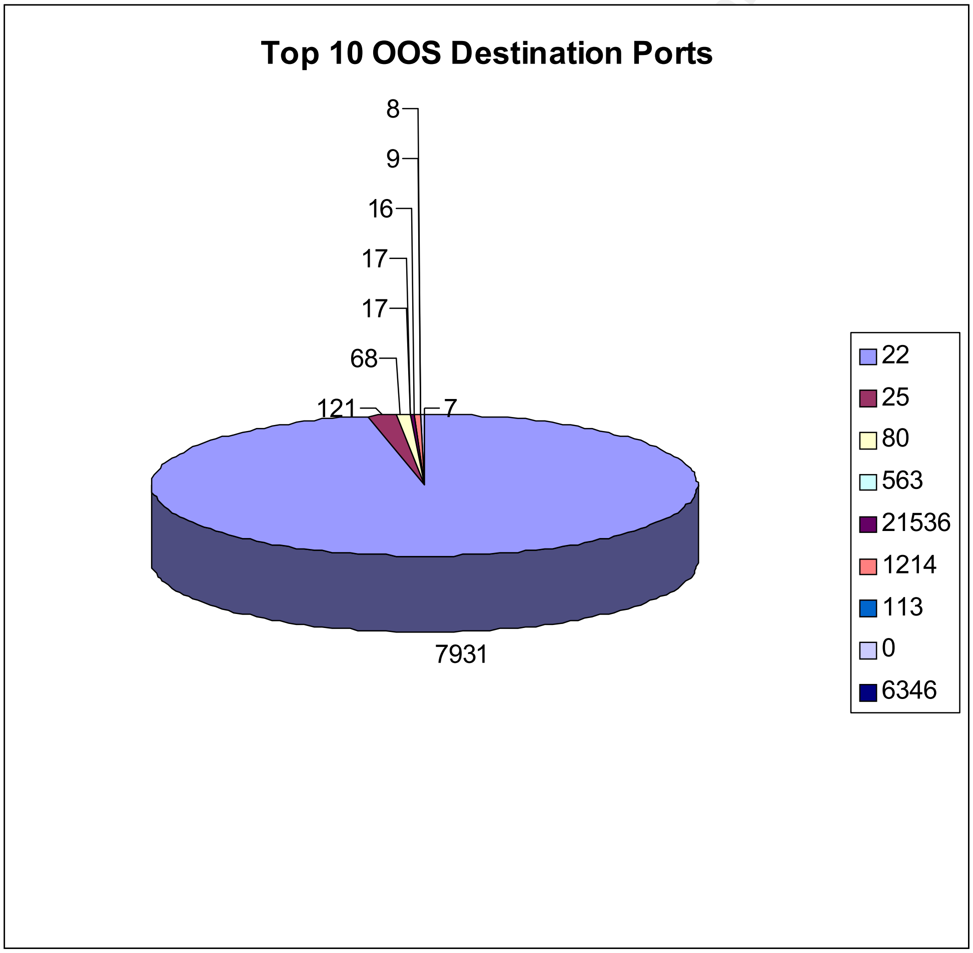
Top 10 OOS Destination Ports

The list of top destination ports for OOS activity is interesting. The SSH activity corresponds to the broad SSH port sweep described above. SMTP and HTTP scanning activity is also very common and can be observed readily at www.incidents.org. The port 563 traffic, which appears to be NNTP over TLS/SSL seems strange. It is one of the ports tested by Ken Kalish's "Remote Security Tester" site, www.mycgiserver.com/~kalish, but it does not seem to indicate why this port is chosen. The FireTower RAPTOR Firewall FAQtory discusses Raptor's HTTP proxy service at http://www.firetower.com/faqs/proxies/httpd/ports-other.html, and indicates that port 563 is a possible HTTP proxy port (this seems to be a more likely target than SNNTP). The "Broken Network Device" detects seem to be fairly well known, and are discussed above in Assignment 2, Detect 4. Kazaa and Gnutella traffic have become familiar during the earlier scan analysis, and appear to be common traffic types on this network.

Count	Port	Protocol	Port Description

7931	22	TCP	SSH
121	25	TCP	SMTP
68	80	TCP	HTTP
17	563	TCP	NNTP over TLS/SSL
17	21536	TCP	Broken Network Device (see Assignment 2, Detect 4)
16	1214	TCP	KAZAA
9	113	TCP	AUTH
8	0	TCP	Reserved / “Ping” – probe for host presence / response, and/or fingerprint target based on response
7	6346	TCP	Gnutella

Top 10 OOS Destination Ports



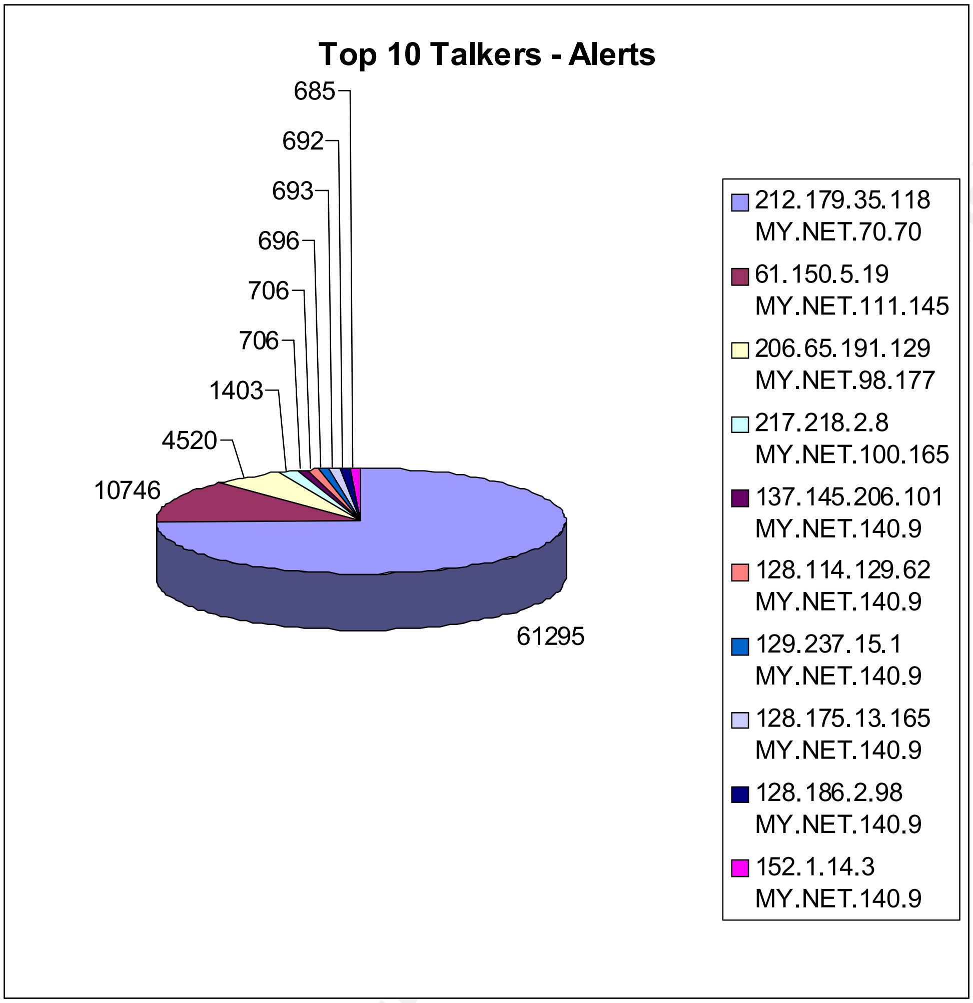
Top Talkers

Alerts

As is to be expected, the alert types for the top ten talkers are all among the top alert types overall. A review of the top alert sources and their associated alert types, as well as the top alert destinations and types using Chris Kuethe's alertcount script (described below, results not included in report for brevity) clearly identify these hosts and alert types as being among the top sources, destinations, and alert types overall. In other words, the same sorts of clustering/concentration on specific source/destination combinations as was described in the Top OOS Destinations discussion can be observed here, although the distribution isn't quite as clear-cut as was observed in the relatively simple OOS traffic patterns. The MISC traceroute traffic concentration on MY.NET.140.9, discussed above in the alert descriptions, is particularly troublesome.

Frequency	Source	Destination	Alert Type(s)
61295	212.179.35.118	MY.NET.70.70	Watchlist 000220 IL-ISDNNET-990517
10746	61.150.5.19	MY.NET.111.145	MISC Large UDP Packet
4520	206.65.191.129	MY.NET.98.177	Queso fingerprint, Null scan!
1403	217.218.2.8	MY.NET.100.165	CS WEBSERVER - external web traffic
706	137.145.206.101	MY.NET.140.9	MISC traceroute
706	128.114.129.62	MY.NET.140.9	MISC traceroute
696	129.237.15.1	MY.NET.140.9	MISC traceroute
693	128.175.13.165	MY.NET.140.9	MISC traceroute
692	128.186.2.98	MY.NET.140.9	MISC traceroute
685	152.1.14.3	MY.NET.140.9	MISC traceroute

Top 10 Talkers - Alerts

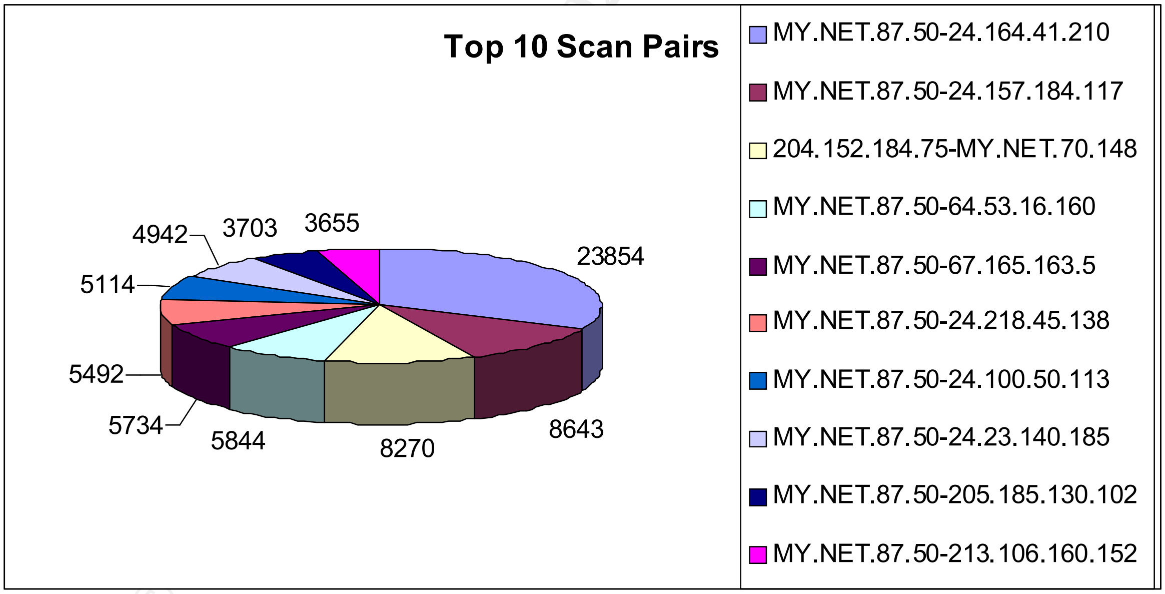


Scans

The top talkers analysis corresponds almost exactly to the top scan sources and destinations presented above. It would appear that many of the conversations were clustered or concentrated between particular hosts, as was observed for the top alert conversations. MY.NET.87.50 was by far the most active scan source overall, and this is immediately apparent from the top talkers list – all but one of the top 10 conversations are from this source. The list of destinations scanned corresponds almost exactly with the top destinations overall (MY.NET.70.148 and 24.157.184.117 are flipped in position), and the counts from these conversations make up the bulk of the total scan counts for these destinations overall (which is reasonable, as the likelihood of other internal scanners targeting the same external hosts would seem unlikely, unless there was some sort of coordination or information sharing going on). The fact that MY.NET.70.148 is the only internal host on the list, and received a relatively higher number of scans overall, seems reasonable. (i.e. The particular scans from 204.152.184.75 do not fully account for all of the scans against this internal host, which makes sense – we would have more overall data for an internal host than the other destinations listed in this table). As usual, a table and graph of the results are presented to help illustrate the behavior.

Count	Source	Destination
23854	MY.NET.87.50	24.164.41.210
8643	MY.NET.87.50	24.157.184.117
8270	204.152.184.75	MY.NET.70.148
5844	MY.NET.87.50	64.53.16.160
5734	MY.NET.87.50	67.165.163.5
5492	MY.NET.87.50	24.218.45.138
5114	MY.NET.87.50	24.100.50.113
4942	MY.NET.87.50	24.23.140.185
3703	MY.NET.87.50	205.185.130.102
3655	MY.NET.87.50	213.106.160.152

Top 10 Talkers - Scans



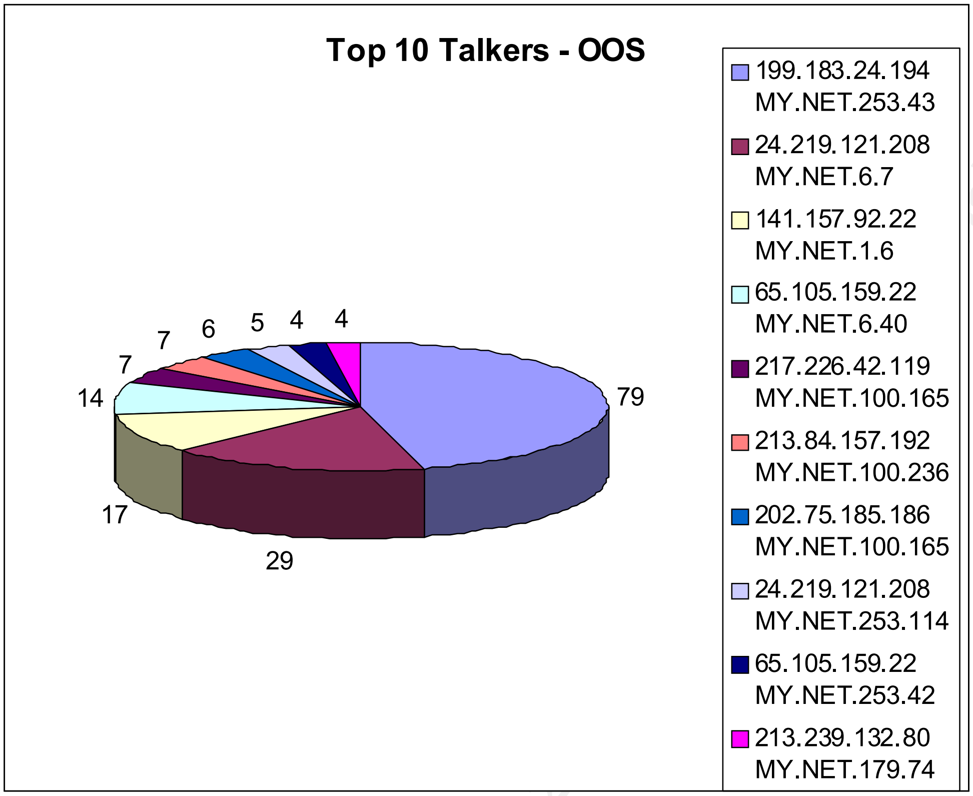
OOS

The OOS top talkers list differs from the individual breakdowns analyzed earlier, as the large-scale port sweep by 24.0.28.234 did not involve any major conversations with individual hosts. This sweep is then factored out by the top talkers analysis, allowing the analysis to focus on the other OOS data. As with the other top talkers, most of the top conversations involved the same hosts, and the majority of the events, noted in the top OOS source and destination analyses above, with some variations in sequencing / rankings. Further analysis of the specific detects represented here reveal some strong similarities and patterns, which are discussed below in the discussion notes table.

Comment ID	Count	Source	Source Port[6]	Destination	Dest Port
A	79	199.183.24.194	32000	MY.NET.253.43	25
A, D	29	24.219.121.208	3200	MY.NET.6.7	80
A	17	141.157.92.22	60000-64000	MY.NET.1.6	563
B, E	14	65.105.159.22	Random High	MY.NET.6.40	25
A, F	7	217.226.42.119	64000	MY.NET.100.165	80
A	7	213.84.157.192	45000	MY.NET.100.236	1214
A, F	6	202.75.185.186	2000	MY.NET.100.165	80
A, D	5	24.219.121.208	3400	MY.NET.253.114	80
A, E	4	65.105.159.22	Random High	MY.NET.253.42	25
C	4	213.239.132.80	50573	MY.NET.179.74	1080

Top 10 Talkers - OOS

[6] This is intended to denote the general range of source ports. The last entry in the table (source IP 213.239.132.80) is the only entry with a completely static source port.



Discussion

Comment ID	Discussion
A	General packet characteristics appear to be normal, and vary as would be expected from one packet to the next. The unusual characteristic here is the flags setting: "21S*****". This could be legitimate traffic using ECN if the TOS field was set appropriately, but none of the OOS top talkers had non-zero TOS fields. Thus, indications are that this data is part of a scanning attempt, and more specifically is probably part of an OS fingerprinting exercise.
B	Same basic characteristics as Comment A, but with some repeating sequence numbers (for the same source and destination ports and addresses). There is some increasing delay between packets, but not a clearly standard TCP incremental back-off. IP ID's are changing across all packets. As the TOS is 0, this is still invalid data and likely part of a fingerprinting exercise, but the repeating sequences were interesting.
C	This data set was a collection of 4 nearly duplicate packets. The packet characteristics were as per Comment A for the most part. The sequence numbers used were identical for all four packets,

	but the IP ID's varied. The TTL was 1, while the others traces all had more normal values. The series of packets does follow the standard TCP incremental back-off values, with time delays between packets of 3, 6, and then 12 seconds. Again, the TOS is 0, so it is invalid ECN data and likely a sign of fingerprinting, but it appears that the fingerprint attempt may have been retransmitted. (?)
D	Note the duplicate source IP address, and similar source ports. The source appears to have been busy, and a check of the Top 10 OOS Source Addresses shows that this source was the third most active OOS source. Target IP's are different, but the target port is the same.
E	As with Comment D, we have duplicate source IP addresses. No specific pattern noted in source ports. Also like Comment D, the target IP's are different, but the destination port is the same. This source was the fourth highest OOS source overall.
F	In this case, the destination IP's are the same, as are the destination ports. The source IP's are from quite different subnets, however (different continents actually – one is German, the other Malaysian). The duplication was initially of interest, and raises some questions about why this single IP address showed up twice in the top OOS data, but a check of the rest of the OOS data for that particular IP shows only two other matches. Overall – probably just a coincidence, but perhaps worth watching in the future.

A sample of the traffic observed follows. This is representative of the basic "A" pattern, with the other data being very similar except for the minor variations discussed.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/27-02:18:39.494002 199.183.24.194:36303 -> MY.NET.253.43:25
TCP TTL:52 TOS:0x0 ID:47734  DF
21S***** Seq: 0xB4259067   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 236136977 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/27-03:42:19.792019 199.183.24.194:47978 -> MY.NET.253.43:25
TCP TTL:52 TOS:0x0 ID:45873  DF
21S***** Seq: 0xEF5E61EA   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 236638939 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/27-05:10:01.280108 199.183.24.194:41363 -> MY.NET.253.43:25
TCP TTL:52 TOS:0x0 ID:14106  DF
21S***** Seq: 0x3B63EEC5   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 237165023 0 EOL EOL EOL EOL
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/27-06:00:41.316855 199.183.24.194:51405 -> MY.NET.253.43:25
TCP TTL:52 TOS:0x0 ID:32161  DF
21S***** Seq: 0xFA30009F   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 237468990 0 EOL EOL EOL EOL

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
12/27-06:10:34.130375 199.183.24.194:57050 -> MY.NET.253.43:25
TCP TTL:52 TOS:0x0 ID:63516  DF
21S***** Seq: 0x1F1D9CD8   Ack: 0x0   Win: 0x16D0
TCP Options => MSS: 1460 SackOK TS: 237528260 0 EOL EOL EOL EOL
```

Source Address Analysis

The following source addresses were selected for further analysis:

Address	Reason for Selection
24.0.28.234	Top OOS source, one of the top ten scanners, conducted very brazen SSH port sweep across the address space.
61.150.5.19	Second-highest top talker for alerts (already identified highest alert talker because of the Watchlist alert type)
63.240.202.64	One of the sources of "EXPLOIT x86" style alerts that looks less likely to be a false positive (i.e. this one may be a real attack, and a serious one).
207.46.178.10	Source of the "EXPLOIT NTPDX buffer overflow" alert, one of the possible deliberate attacks noted.
212.95.76.165	Top external scan source.

The registration information is presented below. Information was retrieved using the Dshield / Incidents.org IP information lookup utility (http://www.dshield.org/ipinfo.php). In addition to the basic whois address information, this tool summarizes interesting information about the IP being researched, including the number of events for that particular IP that are present in the Dshield database, notes about messages to the contact email address bouncing, etc.

<table>
<tr><th>Address</th><th>Registration</th></tr>
<tr><td>24.0.28.234</td><td>
<table>
<tr><td><b>IP Address:</b></td><td>24.0.28.234</td></tr>
<tr><td><b>HostName:</b></td><td>dhcp-24-0-28-234.corp.home.net</td></tr>
<tr><td><b>DShield Profile:</b></td><td>
<table>
<tr><td>Country:</td><td>US</td></tr>
<tr><td>Contact E-mail:</td><td>noc-abuse@noc.home.net (bounced)</td></tr>
<tr><td>Total Records against IP:</td><td>12</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

Number of targets:	11
Date Range:	2001-12-18 to 2001-12-18

Ports Attacked (up to 10):

Port	Attacks

```
@Home Network (NETBLK-ATHOME)
   450 Broadway Street
   Redwood City, CA 94063
   US

   Netname: ATHOME
   Netblock: 24.0.0.0 - 24.23.255.255
   Maintainer: HOME

   Coordinator:
      Operations, Network  (HOME-NOC-ARIN)  noc-
abuse@noc.home.net
      (650) 556-5599

   Domain System inverse mapping provided by:

   NS1.HOME.NET                24.0.0.27
   NS2.HOME.NET                24.2.0.27

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

   Record last updated on 10-Apr-2000.
   Database last updated on  19-Dec-2001 19:56:11 EDT.

----------

@Home Network (NETBLK-HOME-CORP-1)
   425 Broadway
   Redwood City, CA 94063
   US

   Netname: HOME-CORP-1
   Netblock: 24.0.16.0 - 24.0.31.255

   Coordinator:
      Operations, Network  (HOME-NOC-ARIN)  noc-
abuse@noc.home.net
      (650) 556-5599

   Record last updated on 09-Apr-1998.
   Database last updated on  19-Dec-2001 19:56:11 EDT.

The ARIN Registration Services Host contains ONLY
```

	Internet Network Information: Networks, ASN's, and related POC's. Please use the whois server at rs.internic.net for DOMAIN related Information and whois.nic.mil for NIPRNET Information.
61.150.5.19	**IP Address:** 61.150.5.19 **HostName:** 61.150.5.19 **DShield Profile:** Country: Contact E-mail: Total Records against IP: Number of targets: Date Range: to Ports Attacked (up to 10): **Port Attacks** % Rights restricted by copyright. See http://www.apnic.net/db/dbcopyright.html % (whois7.apnic.net) inetnum: 61.150.0.0 - 61.150.31.255 netname: SNXIAN descr: xi'an data branch,XIAN CITY SHAANXI PROVINCE country: CN admin-c: WWN1-AP tech-c: WWN1-AP mnt-by: MAINT-CHINANET-SHAANXI mnt-lower: MAINT-CN-SNXIAN changed: ipadm@public.xa.sn.cn 20010309 source: APNIC person: WANG WEI NA address: Xi Xin street 90# XIAN country: CN phone: +8629-724-1554 fax-no: +8629-324-4305 e-mail: xaipadm@public.xa.sn.cn nic-hdl: WWN1-AP mnt-by: MAINT-CN-SNXIAN changed: wwn@public.xa.sn.cn 20001127 source: APNIC
63.240.202.64	**IP Address:** 63.240.202.64 **HostName:** 63.240.202.64 **DShield Profile:** Country: US

Contact E-mail:	dns@CERF.NET
Total Records against IP:	
Number of targets:	
Date Range:	to

Ports Attacked (up to 10):

Port	Attacks

```
AT&T CERFnet (NETBLK-CERFNET-BLK-5)
   P.O. Box 919014
   San Diego, CA  92191
   US

   Netname: CERFNET-BLK-5
   Netblock: 63.240.0.0 - 63.242.255.255
   Maintainer: CERF

   Coordinator:
      AT&T Enhanced Network Services  (CERF-HM-ARIN)
dns@CERF.NET
      (619) 812-5000

   Domain System inverse mapping provided by:

   DBRU.BR.NS.ELS-GMS.ATT.NET  199.191.128.106
   CBRU.BR.NS.ELS-GMS.ATT.NET  199.191.128.105
   DMTU.MT.NS.ELS-GMS.ATT.NET  12.127.16.70
   CMTU.MT.NS.ELS-GMS.ATT.NET  12.127.16.69

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

   Record last updated on 06-Aug-2001.
   Database last updated on  18-Feb-2002 19:56:12 EDT.

The ARIN Registration Services Host contains ONLY
Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for
DOMAIN related Information and whois.nic.mil for NIPRNET
Information.
```

207.46.178.10

IP Address: 207.46.178.10

HostName: 207.46.178.10

DShield Profile:

Country:	US
Contact E-mail:	noc@microsoft.com
Total Records against IP:	114

Number of targets:	77
Date Range:	2002-02-06 to 2002-02-06

Ports Attacked (up to 10):

Port	**Attacks**

```
Microsoft (NETBLK-MICROSOFT-GLOBAL-NET)
   One Redmond Way
   Redmond, WA 98052
   US

   Netname: MICROSOFT-GLOBAL-NET
   Netblock: 207.46.0.0 - 207.46.255.255

   Coordinator:
      Microsoft  (ZM39-ARIN)  noc@microsoft.com
      425-936-4200

   Domain System inverse mapping provided by:

   DNS1.CP.MSFT.NET            207.46.138.20
   DNS2.CP.MSFT.NET            207.46.138.21
   DNS1.TK.MSFT.NET            207.46.232.37
   DNS1.DC.MSFT.NET            207.68.128.151
   DNS1.SJ.MSFT.NET            207.46.97.11

   Record last updated on 20-Jun-2001.
   Database last updated on  12-Jan-2002 02:32:38 EDT.

The ARIN Registration Services Host contains ONLY 
Internet
Network Information: Networks, ASN's, and related POC's.
Please use the whois server at rs.internic.net for
DOMAIN related Information and whois.nic.mil for NIPRNET
Information.
```

212.95.76.165

IP Address: 212.95.76.165

HostName: ip-76-165.evc.net

DShield Profile:

Country:	FR
Contact E-mail:	netmaster@sdv.fr
Total Records against IP:	7
Number of targets:	7
Date Range:	-none- to -none-

Ports Attacked (up to 10):

Port	**Attacks**

```
% 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:      212.95.72.0 - 212.95.79.255
netname:      EV
descr:        Est-Videocommunication
descr:        26 Boulevard du president Wilson
descr:        67954 Strasbourg Cedex
descr:        France
descr:        Ip Block #1 provided by SdV
country:      FR
admin-c:      GB8119-RIPE
tech-c:       SG727-RIPE
status:       ASSIGNED PA
notify:       ripe-dbm@sdv.fr
mnt-by:       SDV
mnt-lower:    SDV
changed:      salim@sdv.fr 20020204
source:       RIPE

route:        212.95.64.0/19
descr:        FR-SDV
descr:        SdV Plurimedia IP-Block #1
origin:       AS8839
cross-mnt:    SDV
mnt-by:       SDV
changed:      salim@sdv.fr 19991209
source:       RIPE

person:       Gaston Burger
address:      EST VIDEOCOMMUNICATION
address:      42 route de Bischwiller
address:      67300 SCHILTIGHEIM
phone:        +33 3 88 76 44 60
fax-no:       +33 3 88 76 44 69
e-mail:       gaston@evc.net
nic-hdl:      GB8119-RIPE
changed:      ripe-dbm-updates@nic.fr 20000825
source:       RIPE

person:       Salim GASMI
address:      SDV PLURIMEDIA
address:      15, rue de la nuee bleue
address:      67000 STRASBOURG
address:      France
phone:        +33 3 88 75 80 50
fax-no:       +33 3 88 23 56 32
```

	e-mail: netmaster@sdv.fr nic-hdl: SG727-RIPE mnt-by: RAIN-TRANSPAC changed: ingo@rain.fr 20000309 source: RIPE

Comments on the resulting address lookups:

Address	Comment
24.0.28.234	@Home cable modem subscriber. This address range is generally well known as a major source of malicious activity, and so the results are not surprising.
61.150.5.19	Chinese address. China is another fairly common source of activity.
63.240.202.64	AT&T CERFnet Internet services. Not overly conclusive, as this could be just about anyone using this AT&T service.
207.46.178.10	Microsoft. ☺ Although possible this is an attack from Microsoft, even the paranoid side of me considers this questionable. This does not mean it couldn't happen, or that the attack is necessarily a false positive. As NTP uses UDP, which is easily spoofed, the alert might have been a legitimate attack using spoofed address. In such a case, Microsoft would be a likely spoofed source selection of many attackers, given Microsoft's unpopularity and "preferred evil monopoly" status with such groups.
212.95.76.165	Video communications company in France. Some records against this IP exist in the DShield database, which may indicate a history of activity, although there are not a lot of entries.

Correlations

The basic format for this list of correlations is based on Lloyd Webb's GCIA practical (GCIA 0422, http://www.giac.org/practical/Lloyd_Webb_GCIA.doc), as are the initial base of correlations. Unless otherwise noted, the references are derived from various GCIA practicals, and simply the name of the student, their GCIA number, and the URL are provided, in order to keep the correlations brief. Secondary correlations are generally provided for less common activity.

Alert	Correlation
Watchlist 000220 IL-ISDNNET-990517	Webb, Lloyd. GCIA 0422. URL http://www.giac.org/practical/Lloyd_Webb_GCIA.doc
MISC traceroute	Comment on false positives: Ginnetty, James. Snort-Users Mailing list. "[Snort-Users] Newbie Question." URL

	http://archives.neohapsis.com/archives/snort/2001-03/thread.html#235 SANS Institute Course Notes – GCIA Track for SANS CDI West, 2001.
CS WEBSERVER - external web traffic	French, Jamie. SANS Forum discussions. "Analyze this Detects" URL http://forum.sans.org/discus/messages/78/1716.html?1012743539
MISC source port 53 to <1024	Current signature information: Snort Signature Database. "MISC source port 53 to <1024." URL http://www.snort.org/snort-db/sid.html?id=504 Discussion of signature issues and false positives with UDP version of signature: Snort-Users Mailing List. "[Snort-users] MISC source port 53 to <1024 question." URL http://archives.neohapsis.com/archives/snort/2001-10/thread.html#253
ICMP Echo Request BSDtype	Snort Signature Database. "ICMP PING BSDtype." URL http://www.snort.org/snort-db/sid.html?id=368 SANS Institute Course Notes – GCIA Track for SANS CDI West, 2001.
WEB-MISC prefix-get //	Bauduin, Raphael. Snort-Users Mailing List discussion. "[Snort-Users] rule meaning." URL http://archives.neohapsis.com/archives/snort/2000-10/0331.html
MISC Large UDP Packet	Yuen, Rick. GCIA 0435. URL http://www.giac.org/practical/Rick_Yuen_GCIA.doc
ICMP Source Quench	Arkin, Ofir. "ICMP Usage in Scanning: The Complete Know-How." Version 3.0. URL http://www.sys-security.com/archive/papers/ICMP_Scanning_v3.0.pdf SANS Institute Course Notes – GCIA Track for SANS CDI West, 2001.
INFO MSN IM Chat data	Snort Signature Database. "INFO MSN chat access." URL http://www.snort.org/snort-db/sid.html?id=540 General article on dangers of instant messaging software such as MSN IM and AIM: Stiennon, Richard. Gartner Viewpoint. "Commentary: AIM's ever-present risk." URL http://news.com.com/2009-1023-800719.html

SCAN Proxy attempt	Rubio, Reuben. GCIA 0432. URL http://www.giac.org/practical/REUBEN_RUBIO_GCIA.doc Lamb, Jason. Intrusions Mailing List discussion. "Proxy Scanning Activities." URL http://www.incidents.org/archives/intrusions/msg03757.html
ICMP Destination Unreachable (Communication Administratively Prohibited)	Arkin, Ofir. "ICMP Usage in Scanning: The Complete Know-How." Version 3.0. URL http://www.sys-security.com/archive/papers/ICMP_Scanning_v3.0.pdf SANS Institute Course Notes – GCIA Track for SANS CDI West, 2001.
Queso fingerprint	Webb, Lloyd. GCIA 0422. URL http://www.giac.org/practical/Lloyd_Webb_GCIA.doc
SYN-FIN scan!	Webb, Lloyd. GCIA 0422. URL http://www.giac.org/practical/Lloyd_Webb_GCIA.doc
ICMP Destination Unreachable (Host Unreachable)	Arkin, Ofir. "ICMP Usage in Scanning: The Complete Know-How." Version 3.0. URL http://www.sys-security.com/archive/papers/ICMP_Scanning_v3.0.pdf SANS Institute Course Notes – GCIA Track for SANS CDI West, 2001.
BACKDOOR NetMetro File List	Hovila, Jyri. Snort-Users Mailing List Discussion. "[Snort-Users] BACKDOR ??" URL http://www.geocrawler.com/archives/3/4890/2001/10/50/6941021/
ICMP Fragment Reassembly Time Exceeded	Arkin, Ofir. "ICMP Usage in Scanning: The Complete Know-How." Version 3.0. URL http://www.sys-security.com/archive/papers/ICMP_Scanning_v3.0.pdf SANS Institute Course Notes – GCIA Track for SANS CDI West, 2001.
ICMP Echo Request Nmap or HPING2	Webb, Lloyd. GCIA 0422. URL http://www.giac.org/practical/Lloyd_Webb_GCIA.doc
INFO FTP anonymous FTP	Goodwin, PJ. GCIA 0305. URL http://www.giac.org/practical/PJ_Goodwin_GCIA.doc
SMB Name Wildcard	Webb, Lloyd. GCIA 0422. URL http://www.giac.org/practical/Lloyd_Webb_GCIA.doc
ICMP Destination	Arkin, Ofir. "ICMP Usage in Scanning: The Complete Know-How." Version 3.0. URL http://www.sys-

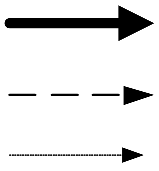
Unreachable (Protocol Unreachable)	security.com/archive/papers/ICMP_Scanning_v3.0.pdf SANS Institute Course Notes – GCIA Track for SANS CDI West, 2001.
WEB-MISC Attempt to execute cmd	Snort-Users mailing list archive. Thread '[Snort-users] "Attempt to execute cmd" Surge!' URL http://archives.neohapsis.com/archives/snort/2001-08/thread.html#185 Snort Signature Database. "WEB-IIS cmd.exe access." URL http://www.snort.org/snort-db/sid.html?id=1002
EXPLOIT x86 {NOOP, stealth noop, setgid 0, setuid 0}	Snort-Users mailing list archive. Thread '[Snort-users] "SHELLCODE x86 NOOP" from presumably non dangerous addresses'. URL http://www.geocrawler.com/mail/msg.php3?msg_id=7204949&list=4890 Aleph1. Phrack 49. "Smashing the Stack for Fun and Profit." URL http://www.insecure.org/stf/smashstack.txt Intrusions Mailing list discussion "re: Experimental shellcode" URL http://www.incidents.org/archives/intrusions/msg03756.html
TFTP - Internal TCP connection to external tftp server	Snort Signature Database. "WEB-MISC tftp attempt" URL http://www.snort.org/snort-db/sid.html?id=1068. Mentions use of TFTP to transfer additional tools to a compromised host. CERT. "CERT Advisory CA-2001-26 Nimda Worm." URL http://www.cert.org/advisories/CA-2001-26.html Miller, Nate. "Microsoft IIS Unicode Exploit." URL http://www.lucent.com/livelink/197020_Whitepaper.pdf
SNMP public access	Webb, Lloyd. GCIA 0422. http://www.giac.org/practical/Lloyd_Webb_GCIA.doc
X11 Outgoing	Snort Signature Database. "X11 Outgoing." URL http://www.snort.org/snort-db/sid.html?id=1227
IDS50/trojan_trojan-active-subseven	Wagner, George. "Intrusion Detection FAQ: SubSeven Trojan V1.1". URL http://www.sans.org/newlook/resources/IDFAQ/subseven.htm (February 19, 2002). Woodroffe, Allan. GCIA 0433.

	URL http://www.giac.org/practical/Alan_Woodroffe_GCIA.doc
EXPLOIT NTPDX buffer overflow	Description of vulnerability, comments on spoofing opportunities: SecuriTeam NTP Advisory. “NTPD vulnerable to a remotely exploitable buffer overflow (readvar).” URL http://www.securiteam.com/unixfocus/5PP032K40A.html

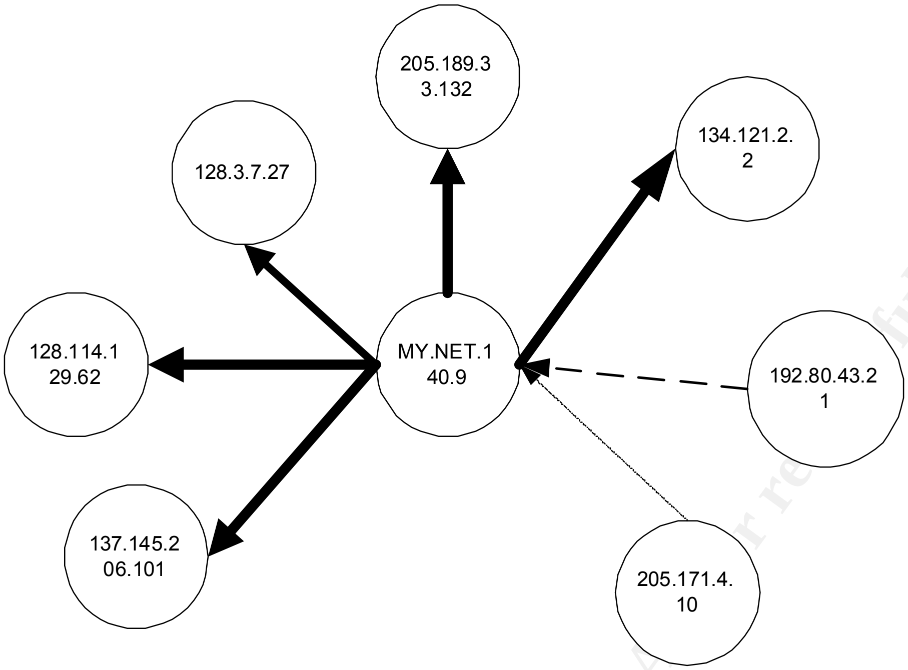
Link Graph

The following link graph is representative of the relationships between MY.NET.140.9 and some of the other hosts with which it generated alerts. This particular host generated a number of MISC Traceroute alerts to a variety of different external hosts. The quantity of alerts with each relationship was fairly significant. Additionally, several ICMP error messages were returned to the host. These included both admin prohibited and host unreachable messages. Three types of lines are used to illustrate the interactions. The arrow on the line denotes the direction of the alert from source to destination, which the weight (darkness) of the line is generally intended to convey the frequency of alerts. Solid lines denote “MISC Traceroute” alerts, dashed lines “ICMP Destination Unreachable (Communication Administratively Prohibited)”, and dotted lines “ICMP Destination Unreachable (Host Unreachable).”

Sample Lines:



Link Graph:



Areas of Concern

Many of the individual areas of concern have been discussed during the various analysis sections above. Some of the key concerns are summarized here:

- Perimeter security – Given the level of anomalous or malicious activity traversing the network, it seems apparent that there is very little in place in the way of perimeter security controls. This increases the risk to all hosts on the network tremendously.
- Internal hosts initiating scans – The top scan source by far was an internal host. This host in particular is a major concern, and may well be compromised. A number of other internal hosts were also involved in scanning activity, and should be investigated.
- File Sharing – Many of the alerts and scans noted were related to file sharing activity (kaxaa, gnutella, edonkey2000, etc.). At best, many of these applications may be considered wasteful of bandwidth and generally inappropriate usage of resources. Worse, they often constitute sources of virus or Trojan infection, information compromise, and security vulnerabilities. Many of these applications are designed to circumvent security controls, and thus by this very nature are a concern.
- Gaming – In addition to file sharing, gaming activity and scanning appeared prevalent. As with file sharing, this is wasteful of resources and may be considered inappropriate use, in addition to the vulnerabilities that have been identified in various network gaming applications.
- IDS tuning – Many of the alerts identified would appear to be potential or likely false positives. Tuning an IDS is a vital part of a successful

deployment, in order to avoid flooding the analyst with more data than can be handled. Unnecessary false positives increase the risk of a true attack going unnoticed among all the background noise. Although false positives are always a problem with IDS, even when well tuned, this particular data set seems to indicate that very little tuning has gone on. Additionally, the IDS configuration should be examined to determine whether logging of packet contents is a practical step that can be taken to provide better quality information to intrusion analysts, rather than the current simple alerts.

- Technical precautions – Several technical precautions and specific perimeter security measures for things such as SNMP, FTP, and X11 were identified during the analysis discussions.

Defensive Recommendations

Defensive measures for a university network are a difficult issue. There is frequently little understanding or appreciation of the risks posed to the university, and to the larger community, by both faculty and administration. Faculty are often extremely vocal in their demand for free access in the interests of freedom of information, "research," and similar needs. Although these are important tenets of the academic community, and Western society in general, it should be recognized that basic security precautions facilitate these goals by protecting research, availability of information and resources, and generally improving stability. This introduction should not be taken as too harsh a criticism of universities, but is intended to highlight some of the issues that have been experienced by the author in working with universities to try to implement basic security measures.

The recommendations being made in this section, then, are tempered by some of these issues. The intent of the recommendations is to make some basic improvements that will be more likely to be acceptable to a university community than the stronger measures typically deployed in the corporate world, particularly if strong backing from the institution's administration is not present.

Recommendations:

- Establish strong policies surrounding appropriate usage and security in general. Ensure that topics such as misuse of resources, malicious activity, user responsibilities, and potential consequences are clearly communicated to all users.
- Establish some basic perimeter security measures. External connections should pass through a well-maintained and -configured firewall. A default deny policy for incoming traffic should be put in place. Ideally, such a policy would also be applied to outgoing traffic, but this may be too difficult to obtain university buy-in on. Blocking incoming traffic will filter much of the noise from the network, and should not cause as much impact to

faculty and staff. Filtering of "known bad" outgoing traffic as a minimum precaution for the other direction is hopefully a compromise that will be accepted, and will help to further reduce the basic noise and malicious activity on the network, allowing intrusion analysts to focus more heavily on the truly dangerous activity. Providing for research and educational requirements within this policy will be essential to the success of the initiative, although difficult to balance at times. The capability should be in place on the firewall to quickly filter malicious traffic, whether incoming or outgoing, regardless of the basic configuration. (i.e. Given that significant allowances will likely have to be made for various types of traffic, there will likely be incidents associated with these. Being able to respond and block this activity in the event of an incident will be important.)

- Technical Guidelines – Technical guidelines for the deployment of common network infrastructure and shared systems should be established as a minimum, in order to ensure basic security precautions are taken on as many systems as possible. Depending on who controls system deployments within the university, it may or may not be possible to enforce these guidelines on all installations.
- Patching and Maintenance – Best practices require that regular system patching and maintenance be carried out, particularly with respect to security updates. Particularly given the current weak perimeter security controls and large amount of malicious internal traffic, ensuring systems are kept up to date and technical guidelines are followed is an essential security precaution.
- Logging – Ensure adequate logging is configured on network infrastructure devices and servers, in order to supplement and validate the detection capabilities of the IDS. The use of a secured, central log server is highly recommended, as is the use of time synchronization software such as NTP to ensure system clocks are all coordinated.
- Tune IDS – As discussed in the Areas of Concern section, the current IDS installation does not appear to be adequately tuned, and is in need of attention in order to ensure analysts are able to detect and focus on truly anomalous or dangerous traffic.
- Additional specific minor recommendations have been made in the analysis discussions, above.

Analysis Process

Initially I had hoped to be able to load the data set into a database, preferably the standard Snort database schema, in order to efficiently analyze it and perform various trending functions. It was hoped that a console such as ACID could be used to assist with this. It quickly became obvious that without the actual packet captures from which the alerts were generated, this was not going to be practical.

It was hoped there would be some tools available to bulk-load basic alert data in to the database, but a web search identified only other queries for this functionality, with no replies. Some consideration was given to hacking the database output plugin from snort to replay alert data and use the existing database logging capabilities, but this was impractical given time constraints. Perl or shell scripts to reformat the data into SQL insert scripts was also given, but again, time was an issue. In either case, there certainly would have been data missing that normally would have gone into the database schema, but which wasn't present in the brief alert output format. Although this data might have been "dummied up," this just added another layer of complexity and database analysis.

The focus of the analysis process then turned to reviewing the work of other students more closely, and reusing the various scripts and tools they had developed. SnortSnarf was the tool that seemed best able to parse and utilize the alert format we were given, but was also clearly a problem for most users, due to its memory and processing requirements. Some not-so-brief experiments with several days' worth of data illustrated this. I was able to process the alerts for individual days, however, and this provided a useful tool for exploring the relationships among various hosts and alerts within a particular day.

The bulk of the analysis was performed by reusing the scripts of several previous GCIA students. Essentially, a collection of different scripts was obtained and experimented with, and the majority of the tools at least tested out to better understand their approach and the analysis they provided. Specific students' work that was utilized included:

- Mike Bell
- Paul Asadoorian
- Chris Kuethe

Lorraine Weaver and Lenny Zeltser's utilities were each reviewed, but not applied heavily during the analysis. Lorraine's scripts seemed to overlap with some of the others. Although Lenny's Berkeley database scripts seem useful (and in hindsight, may have been the better way to go), I was unfamiliar with the underlying tool and decided not to try to learn it at the time.

These scripts produced various summaries and reports that provided the basis for further analysis. Basic Unix commands and shell scripts (awk, grep, etc.) were used to further pull apart the data files and extract information of interest. Excel was used heavily to graph the data for easier comprehension, and its auto-filter capability was also very useful during the detailed analysis phase.

Part of the review phase, after the summaries and major trends had been reviewed, was to focus on some of the less common events, under the assumption that some of the active exploits and particularly nasty traffic would be less prevalent than general scans and anomalies.

Throughout the process, not having access to traffic history data, packet dumps, or information about the environment and hosts involved was a frustration.

Part of the early preparation was reviewing the practicals of a number of previous students. During this phase, in addition to looking for analysis techniques and approaches, I also noticed a couple of reports that featured presentations I particularly liked. I found that Lloyd Webb's report provided a great deal of information in a very clear, concise format. I also liked the graphs that David Hed had produced using Excel, especially with the ability to retrieve additional details from the graphs themselves. These two approaches were borrowed in preparing this report, although probably carried to an extreme that lost much of the conciseness aspect.

Various web sites were used throughout the analysis. Some of the more major ones included:

- www.snort.org/snort-db
- www.google.com
- www.incidents.org, and particularly the Intrusions mailing list
- www.portsdb.org and various other online port databases
- Archives of the Snort-Users mailing list, typically searched using Google.

These sites in particular were typically open constantly during the analysis and report preparation phases.

List of References

Tools and scripts from various GCIA practicals:

Paul Asadoorian, GCIA 0337, URL http://www.giac.org/practical/Paul_Asadoorian_GCIA.zip
Chris Kuethe, GCIA 303, URL http://www.giac.org/practical/chris_kuethe_gcia.html
Mike Bell, GCIA 0318, URL http://www.giac.org/practical/Mike_Bell_GCIA.doc
Lloyd Webb, GCIA 0422, URL http://www.giac.org/practical/Lloyd_Webb_GCIA.doc
David Hed, GCIA 0438, URL http://www.giac.org/practical/David_Hed_GCIA.zip

"RFC 1305 Network Time Protocol (Version 3) Specification and Implementation" URL http://www.faqs.org/rfcs/rfc1305.html

SANS Institute Course Notes – GCIA Track for SANS CDI West, 2001 (various).

Aleph1. Phrack 49. "Smashing the Stack for Fun and Profit," URL http://www.insecure.org/stf/smashstack.txt