Global Information Assurance Certification Paper

Inexpensive Firewall Analysis Project

Daniel DiCioccio Jr.

GSEC Practical Submission

February 23, 2004

TABLE OF CONTENTS

1. Abstract

Firewalls are considered a necessary aspect of defense in depth today. Their logs can offer a great deal of information about the events happening inside and against a network. When these logs are not reviewed, however, the strength of the defense is decreased. This paper details a project that created procedures resulting in a low-cost intrusion detection system built on firewall log analysis by the monitoring group of an actual firm. This project proved to be successful in creating awareness of problems with the network as well as infections and other incidents. It also allowed for the hardening of the firewall rules.

2. Background

The NewAtIt Company had just organized a fully-staffed Information Security Team. Before this, the extent of security measures was minimal. Security Policies had existed but were not properly enforced. An Intrusion Detection System was installed but not functioning properly. The data from the system was incomplete and the system itself was error prone. There was not enough monitoring done to the data collected even if the system had detected something that needed attention. No metrics were being looked at to see what trends in network and internet traffic existed. Security was for the most part not yet a priority for the firm.

When new upper-management came in after two-year reorganization, the priority for information security increased. The new CIO felt that security was an area that needed attention. The budget was allotted, and positions were created and filled. The Information Security team and the processes and defense measures they put into place began to grow. I was one of the new Information Security team members, and was part of the two-member monitoring group within the team.

3. Before

Before the introduction of the full security team, the firm had firewalls in place. The logs from these firewalls were archived and stored for a year according to company policy. While the data was sitting on the logging servers, it was backed up to tape and stored offsite at a data protection warehouse. All the steps necessary to keep the logs safe were taken, but no steps were taken to see what the logs had to say. No one was looking at the logs regularly. Occasionally they were viewed when someone needed to troubleshoot a connection for a server that was just installed, or to determine the volume of internet traffic to a new website, they were able to look at either the live logs or go back to one of the archived dates and analyze them. If there was no need to troubleshoot, the logs were not viewed regularly. There were many risks involved with letting the logs go unviewed. Attacks could occur and no one would know, network segments could cease to communicate correctly and it might be some time before anyone

figured it out. These firewalls were also installed and configured without any security personnel involved. The rules governing these machines of security could have holes in them letting through unwanted traffic.

The first initiative for the monitoring group, as directed by the Information Security officer, was to start daily analysis of these logs. My monitoring supervisor had four years of experience in the Information Security field employed in the banking industry. I however was new to the field of Information Security with a general background in Information Technology.

Unfortunately, the budget for new software purchases was almost nonexistent due to the need to get many other initiatives underway. We could not afford to buy the newest event correlation software or even a better IDS system that worked with our existing firewalls. The team planned to correct the problems and upgrade the current IDS later that year but for now that was a back burner project and the firewall logs had to be deal with presently. The Security Officer also felt that it would benefit me to start with the basics of the protocols and the firewall alerts in order to learn Security from the ground up. This coincided with the need to do things inexpensively.

I was not confident about how we could possibly know what was going on inside of tens or hundreds of millions of lines of event logs every day without software to help us out. I could not visualize how such a task could be carried out by one or two people while also being able to do any other work. I pictured myself sitting in front of a monitor all day long watching text scroll constantly at a fast pace, looking for keywords and text trends. I was surprised when the process eventually unraveled itself and the task seemed possible.

My team had access to the logs and enough drive space to process them. For the next few months, I would learn much about how to turn a mountain of information into reports that would tell us when something was not quite right, and where to start digging around to find out what it was.

4. During
4.1 The Basics

A firewall is a barrier between the outside world and your internal (private) network. It can filter network (IP) traffic, deny certain network requests or transfers, and masquerade externally available IP addresses to internal, private IP addresses. [1] The firewall logs are a text representation of any activity or event that the firewall experiences.

[1] Puget Sound Technology, "Glossary of open source, BSD, Linux, Unix, networking, and free software terminology" URL: http://www.pugetsoundtechnology.com/info/definitions/definitions.html (November 20, 2003)

The foundation of the process created was the firewall logs themselves. In order to know what the logs meant, the team first needed to know a little about the meaning of the messages that comprised the logs. Since the infrastructure utilized Cisco PIX firewalls, a basic introduction to the PIX firewall software was necessary for an understanding of how the data was divided. This particular firewall system categorizes and tags every event that occurs on it and every piece of traffic that passes through it. There are seven categories of messages. Each category is determined by severity or importance of the event. Each category has sublevels or specific codes that denote a particular type of event. Cisco provides descriptions for each event code and recommendations for resolving any issues regarding the event. With this data, information security personnel can get a more detailed view of the events occurring on a given firewall and corresponding network segment.

Below is a description for each severity level of each PIX category:

Alert Messages, Severity 1
Critical Messages, Severity 2
Error Messages, Severity 3
Warning Messages, Severity 4
Notification Messages, Severity 5
Informational Messages, Severity 6
Debugging Messages, Severity 7 [2]

Below is an example of a specific system log message, explanation and recommended action for a specific severity level 3 message.

Log Message %PIX-3-305005: No translation group found for protocol

Explanation This message logs when a nat and global command cannot be found for a protocol. The protocol can be TCP, UDP, or ICMP.

Recommended Action This message can be either an internal error or an error in the configuration. [3]

[2] Cisco Systems "Messages listed by Severity Level-Cisco PIX Firewall Software" URL: http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_system_message_guide_chapter09186a008008d278.html (December 5, 2003)
[3] Cisco Systems "System Log Messages-Cisco PIX Firewall Software" URL: http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_system_message_guide_chapter09186a008008d275.html#24101 (December 5, 2003)

4.2 The Requirements

The security team discussed what data had to be present in order for useful analysis. They determined that there would be 51 total pieces of data collected from the logs from each firewall every day. These decisions were based on the PIX severity levels and the need for trend analysis.

The number of occurrences of each severity level (1-7) event would be collected, the total events of each of three protocols TCP, UDP, and ICMP (when denied only), and the total number of all events per day would make up one report. This would be 11 pieces of numerical data. The other data consisted of the top 10 IP's that were denied and the top 10 ports that were denied based on occurrence. This would be the other 40 pieces of data that would make up two other separate reports. The reasoning behind the choice of filtering will be explained later in the analysis section.

We started monitoring only four firewalls that were considered the most critical, but that number grew as the process proved to be useful and beneficial to all areas of the network. At the time only 18 firewalls could be monitored, but there was a network segmentation project in progress that would, considering the company's worldwide network, add many more firewalls.

The final method of how the data is collected and processed will be explained in more detail but a brief description of the metamorphosis that took place to get the procedures to where they are now will give a feeling for how the process grew. The team did not envision the end product in the beginning, but built on to it with every completed step.

4.3 Primitive Data Collection

The data was first collected from the daily firewall logs resident on a mirror of the centralized Linux syslog server where the logs were compressed using gzip. Syslog is a UDP based text message protocol used to collect various forms of system event logs.[4] Gzip or GNU zip is a UNIX compression utility designed to be a replacement for compress. Its main advantages over compress are much better compression and freedom from patented algorithms. [5] Gzip was necessary for space purposes as the size of some of the event logs were 5 GB for one day uncompressed. This process was carried out first thing in the morning by the monitoring team. In the beginning, this was done by hand by entering UNIX commands into the syslog server. The example below shows how the Severity 7 event totals were extracted. The "PIX-7-" string was indicative of a severity 7 event.

[4] Maso, Brian "Track Performance of Distributed Systems" URL: http://archive.devx.com/free/articles/2000/maso01/maso01-1.asp (January 2, 2004)
[5] Jean-loup Gailly, "The Gzip home page" URL: http://www.gzip.org/ (January 3, 2004)

gunzip –c messages.20030808020300 |grep PIX-7-|wc -l

This was quite tedious, especially considering that some of the logs were 10-30 million lines long and the response from any command like this could take up to one whole minute for each. The team did not accomplish much else besides collecting this data for the four firewalls until a more efficient process was devised. Fortunately, this manual process did not last long. However, these commands were the building blocks of the more advanced automated procedures we use today. Moreover, commands like these still needed to be entered manually in order to research an issue.

Perl scripts were written to extract the daily metrics. Perl is a programming language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information.[6] All of the commands that used to be entered separately by hand were combined together and run at once so the team did not have to waste time entering them, nor waiting for responses. A sample of the original script appears below.

```
var1=`gunzip -c $1 |grep %PIX-1- |wc -l`
var2=`gunzip -c $1 |grep %PIX-2- |wc -l`
var3=`gunzip -c $1 |grep %PIX-3- |wc -l`
var4=`gunzip -c $1 |grep %PIX-4- |wc -l`
var5=`gunzip -c $1 |grep %PIX-5- |wc -l`
var6=`gunzip -c $1 |grep %PIX-6- |wc -l`
var7=`gunzip -c $1 |grep %PIX-7- |wc -l`
var8=`gunzip -c $1 |wc -l`
# This code is returning the numbers of each of the 3 protocol denied.
var9=`gunzip -c $1 |grep " Deny " |grep tcp | wc -l`
var10=`gunzip -c $1 |grep " Deny " |grep udp | wc -l`
var11=`gunzip -c $1 |grep " Deny " |grep icmp | wc -l`
# This line calls the top 10 IP denied module.
var12=`perl -w /home/user/top10IPdeny.pl $1`
# This line calls in the top 10 ports denied.
var13=`perl -w /home/user/top10portsdeny.pl $1`
echo $1
echo 'Severity breakdown'
echo '-------------------------'
echo 'Severity 1 - '$var1
echo 'Severity 2 - '$var2
echo 'Severity 3 - '$var3
echo 'Severity 4 - '$var4
echo 'Severity 5 - '$var5
echo 'Severity 6 - '$var6
echo 'Severity 7 - '$var7
echo 'Protocols denied'
echo '-------------------------'
echo 'Total TCP Denied - ' $var9
echo 'Total UDP Denied - ' $var10
echo 'Total ICMP Denied - ' $var11
echo 'Total lines of the log - '$var8
```

[6] "Perl Synopsis" URL: http://www.perldoc.com/perl5.6/pod/perl.html (January 20, 2004)

Here is an example of the top 10 Denied IP's script

```
#!/usr/bin/perl
$file=$ARGV[0];
@lines=qx!gunzip -c $file | grep " Deny "!;
foreach (@lines) {
($foo,$port1,$port2)=split(/\d*\//,$_);
        if ($port1) {
  ($port1,$foo)=split(/ /,$port1);
  $count{$port1}++;
}
        if ($port2) {
  ($port2,$foo)=split(/ /,$port2);
  $count{$port2}++;
}
}
$nr=0;

foreach $key (sort {$count{$b} <=> $count{$a}} (keys(%count))) {
print "PORT $key ^M";
print "$count{$key} ^M";
$nr++;
last if ($nr>=10)
}
```

When complete, the script would use the UNIX sendmail option to email the results to the team's mailbox. The data was then copied and pasted into Microsoft Excel workbooks by hand, from which the reports and graphs of the data were created. Each step was a manual process and still took up most of the monitoring group's day. During this time, when any major incidents occurred, none of the metrics reporting could be done. The amount of time needed for incident response did not allow for this. It was obvious that the team would need to simplify the work of gathering and compiling the data from the firewalls. There was a clear need for more automation.

It was then realized that a crontab job could be set up and the emails could be sent every morning. Cron is a scheduling daemon in UNIX that periodically searches the cron directory for valid jobs that need to be executed.[7] This would save even more time as the email could be finished before the team arrived to work, instead of while they sat and waited for the long jobs to complete manually. The data was still hand entered from there.

4.4 Efficient Data Collection and Reporting

The team soon realized that this data being collected should be populating a database for archival sake. It could also then be more easily extracted and manipulated. This would prove to save the most time as the data could be exported automatically and the reporting could be done automatically. No hand entering would be needed. The Perl script was updated with MSQL code that exported the data into a large SQL database. Database Adminitstrator's then built queries which extracted the preferred data. Reporting packages were used

[7] "Cron" URL: http://www.ss64.com/bash/cron.html (January 21, 2004)

to create the MS Excel workbooks that were copied to the monitoring team's shared directory. The use of these files will be explained in more detail with the processes surrounding the analysis. This transformation from manual script execution to automated file creation took almost 7 months of process building and enhancement between the security monitoring group and the DBA group. In the end was left an efficient process that took large logs of data and made them into something sensible for very little cost. The monitoring team could now do the job that their title referred to, monitoring.

Each day three MS Excel workbook files are produced and delivered to a security-owned directory on the security database server. The three files are the PIX severity category message counts and denied protocol numbers, the top 10 denied IP's, and the top 10 denied ports. These are called FWReport, IPReport and PortsReport respectively. The workbook files are unformatted data that are easily transformed into more useful and presentable reports and graphs with the use of macros.

The data for a single firewall is on one page or tab of the workbook, named after that firewall, for each respective document. The two top 10 denied workbooks have the most recent seven days worth of data for each firewall, while the FWReport has 28 days of data. The top 10 denied reports have seven days of data in order to see how the data has changed over a week and because it is in proper format for deliver to the weekly status meeting. Only one-week's worth of data is needed for each report, and the data is always available on a larger scale if needed. The FWReport has 28 days of data on it for trend analysis. It was decided that an even four weeks was adequate for illustrating any trends and to determine a baseline when looking at any given day's metrics.[8] The reports are run daily for the monitoring group to see the current status of events and to follow up on any anomalies. A new day's worth of data is added and the oldest day falls off the report.

[8] Reavis, Jim "Is your Internet firewall a bottleneck?" 6 September, 1999 URL: http://www.nwfusion.com/newsletters/sec/0906sec1.html (January 25, 2004)

Below you will see a sample view of the FWReport and the PortsReport (The IP report can not be shown due to sensitivity)

FWReport

fw_name	fw_date	log_total	udp	tcp	icmp	sev1	sev2	sev3	sev4	sev5	sev6	sev7
misu04	1/5/2004	75480	56976	0	4231	0	0	1256	61207	11	13006	0
misu04	1/6/2004	73514	51696	0	5582	0	0	1249	57278	5	14982	0
misu04	1/7/2004	72942	53105	0	4172	0	0	1332	57277	18	14315	0
misu04	1/8/2004	71744	54020	0	3028	0	0	1223	57048	15	13458	0
misu04	1/9/2004	76414	56618	0	5533	0	0	1259	62151	14	12990	0
misu04	1/10/2004	68514	49580	0	5532	0	0	1050	55112	0	12352	0
misu04	1/11/2004	69593	51092	0	4984	0	0	1056	56076	0	12461	0
misu04	1/12/2004	72526	53585	0	4835	0	0	1163	58420	14	12929	0
misu04	1/13/2004	69873	52498	0	3198	0	0	1172	55696	10	12995	0
misu04	1/14/2004	70046	55557	0	343	0	0	1202	55900	8	12936	0
misu04	1/15/2004	74709	54965	0	5393	0	0	1189	60358	28	13134	0
misu04	1/16/2004	68344	50028	0	4200	0	0	1158	54228	2	12956	0
misu04	1/17/2004	64485	46629	0	4382	0	0	1043	51011	0	12431	0
misu04	1/18/2004	63803	46328	0	4010	0	0	1057	50338	0	12408	0
misu04	1/19/2004	71039	52122	40	4752	0	0	1178	56914	13	12934	0
misu04	1/20/2004	71226	53237	0	3886	0	0	1182	57123	8	12913	0
misu04	1/21/2004	73855	55597	0	4394	0	0	1199	59991	10	12655	0
misu04	1/22/2004	75309	55553	0	5547	0	0	1195	61100	14	13000	0
misu04	1/23/2004	72036	51958	60	5573	0	0	1304	57591	16	13125	0
misu04	1/24/2004	55460	36291	0	5675	0	0	1087	41966	0	12407	0
misu04	1/25/2004	53849	34826	0	5686	0	0	1044	40512	0	12293	0
misu04	1/26/2004	66609	46835	0	5615	0	0	1161	52450	27	12971	0
misu04	1/27/2004	69170	49835	0	5508	0	0	1174	55343	6	12647	0
misu04	1/28/2004	68060	48531	0	5527	0	0	1166	54058	47	12789	0
misu04	1/29/2004	69485	49901	0	5500	0	0	1192	55401	23	12869	0
misu04	1/30/2004	69096	49472	0	5575	0	0	1163	55047	16	12870	0
misu04	1/31/2004	58460	39378	0	5641	0	0	1035	45019	0	12406	0
misu04	2/1/2004	43964	15319	0	5497	0	0	10708	20816	0	12440	0

PortsReport

	1/26/2004	1/27/2004	1/28/2004	1/29/2004	1/30/2004	1/31/2004	2/1/2004
PORT 1	**137**	**137**	**137**	**137**	**137**	**137**	**137**
# Denied	60172	56754	50494	24520	30078	3856	15702
PORT 2	**41524**	**41524**	**41524**	**41524**	**41524**	**1036**	**41524**
# Denied	29	32	35	9	80	1	51
PORT 3	**1604**	**1494**	**445**	**1604**	**1035**	**41524**	**1604**
# Denied	8	14	15	4	25	1	11
PORT 4	**1097**	**1604**	**1604**	**1097**	**1036**	**0**	**22**
# Denied	8	6	10	3	19	0	11
PORT 5	**12345**	**1169**	**1097**	**1040**	**1028**	**0**	**1087**
# Denied	8	6	10	2	6	0	10
PORT 6	**1364**	**1097**	**2796**	**1028**	**1043**	**0**	**1029**
# Denied	7	5	3	2	4	0	8
PORT 7	**21**	**3672**	**2811**	**1037**	**1604**	**0**	**12345**
# Denied	5	4	3	2	3	0	7
PORT 8	**22162**	**1042**	**2805**	**1039**	**1133**	**0**	**1043**
# Denied	5	4	3	2	3	0	6
PORT 9	**1028**	**1028**	**2818**	**1042**	**1048**	**0**	**1046**
# Denied	4	4	3	1	3	0	6
PORT 10	**1046**	**3670**	**1054**	**2794**	**1177**	**0**	**1055**
# Denied	4	4	3	1	2	0	4

The FWReport's data is the source data of another workbook that displays the graphs of that report. The graphs are updated daily when the new FWReport document is available using a small macro that changes the source data file date on all the graphs to read off the most current one, thus changing the range to the 28 corresponding dates on the workbook. Each report and graph workbook is saved for later viewing.

4.5. Analysis

The following processes describe ongoing current procedures. This reporting process was created in order to find things that are abnormal so they can be fixed or at least explained. The three reports created daily usually have data in common, or at least data that was caused by the same issue. Analysis can start on any of the reports, but looking at the other ones during analysis is usually helpful. However, the completely raw data in the logs is also usually necessary to determine the problem.

The secret to knowing where to look first starts with trends. The trends of traffic from day to day and week to week can give the analyst a good idea of when something is wrong. This type of analysis cannot point out every possible attack or problem, as some might not leave a heavy log trail, but it can help the analyst to determine the high priority issues. In addition, any given network will have legitimate error messages. It will also have informational events that are harmless but exist solely to log successful and warranted traffic. What stands out however is how these metrics change from day to day and from week to week. The changes in metrics reveal that something has changed inside or outside the network. Once the metrics are collected and organized, analysis and research can be done to determine what, if anything needs to change in order to harden the infrastructure or to optimize network operation.

Each report is studied every weekday morning for certain characteristics. Each day, the data from the before is analyzed, except for Monday when Friday, Saturday, and Sunday are analyzed. Any anomalies found are researched and recorded and the week's findings are presented to the Information Security Officer for his review. More eyes on a report decrease the possibility of missing something of value.

The following explains the process used to detect irregularities that might call for actions to be taken. At the time of writing, the practice of tracking such irregularities was new, the amount of network traffic was very great and in some cases largely due to many systems creating huge amounts of events due to misconfiguration of services and possibly even absence of correct firewall rules necessary for systems to be communicating correctly with one another. The monitoring team had not had enough time to get the misconfigurations fixed by working with the support groups to decrease traffic to a point where most of it is

legitimate. It was a long-term goal to address these illegitimate traffic causing issues, but for now, the team uses the numbers that exist everyday to create a baseline. While this baseline is not truly “normal” or completely acceptable behavior, it is for the most part constant behavior, as most misconfigurations do not change from day to day. The team continues however to keep any new misconfigurations from being overlooked by catching them and documenting them as soon as they happen.

4.5.1 Firewall Data Graphs Analysis

The FWReport Graphs are line graphs showing the 11 points of data using a different color for each for all of the 28 days. The graphs for the most part have a baseline range of event-counts for each category that is considered normal. The graph is studied for deviations in each of the points of data surrounding the date in question. High severity events and anything else that is unusual is also flagged. When data from the date being scrutinized deviates from this baseline, the occurrence is flagged by marking the deviation on the graph with a circle. After researching it, a textbox is added describing what the deviation was.

From that point, research is done on the event log from the dates in question, and possibly others surrounding it. This is important to get a feel for specifics on why the anomaly exists and what it is. This will be visually demonstrated later on. Three different scenarios could result from my research depending upon the reason behind the irregularity.

- It may be determine that the irregularity was caused by something that was acceptable and a known issue. For instance, a scheduled scan was conducted from an outside vendor, which the group warrants. The exception is marked as such on the graph.
- It may be found that the situation is no longer occurring and seems reasonable but warrants confirming the cause of events. A technician responsible for the situation will be contacted and a confirmation that some testing or otherwise normal activity was completed on that particular day is requested. The exception would be noted on the graph with a mention of the follow-up pending confirmation.
- If the events are continuing and a trend is evident, a trouble ticket is opened for the support group to check into the effected systems. This is mainly done for confirmation that something is not wrong on an application level and causing an outage. The ticket number is placed in the textbox on the graph along with a description of the problem. The severity of the issue will affect the severity of the ticket, thereby affecting the speed at which the technician responds to it.

These practices ensure that the firewall data is being monitored regularly for strange occurrences. Oftentimes, when a virus or worm breaks, outside notifications of such and alarms from anti-virus software alert the Monitoring group. The graph analysis usually shows this activity the next day. However, the

graphs can be used after the outbreak in order to track the volume of systems still infected and continue to respond to the issues until the infection is under control. The Nachi Worm affected many laptops, some of which were not patched immediately and had shown up when reconnected to the network anywhere from days to months later. These infections were quite visible with the daily graph analysis. Any non-publicized propagating malware on or scanning the network would be detected by at least the next day.

4.5.2 Top 10 Denied Traffic Analysis

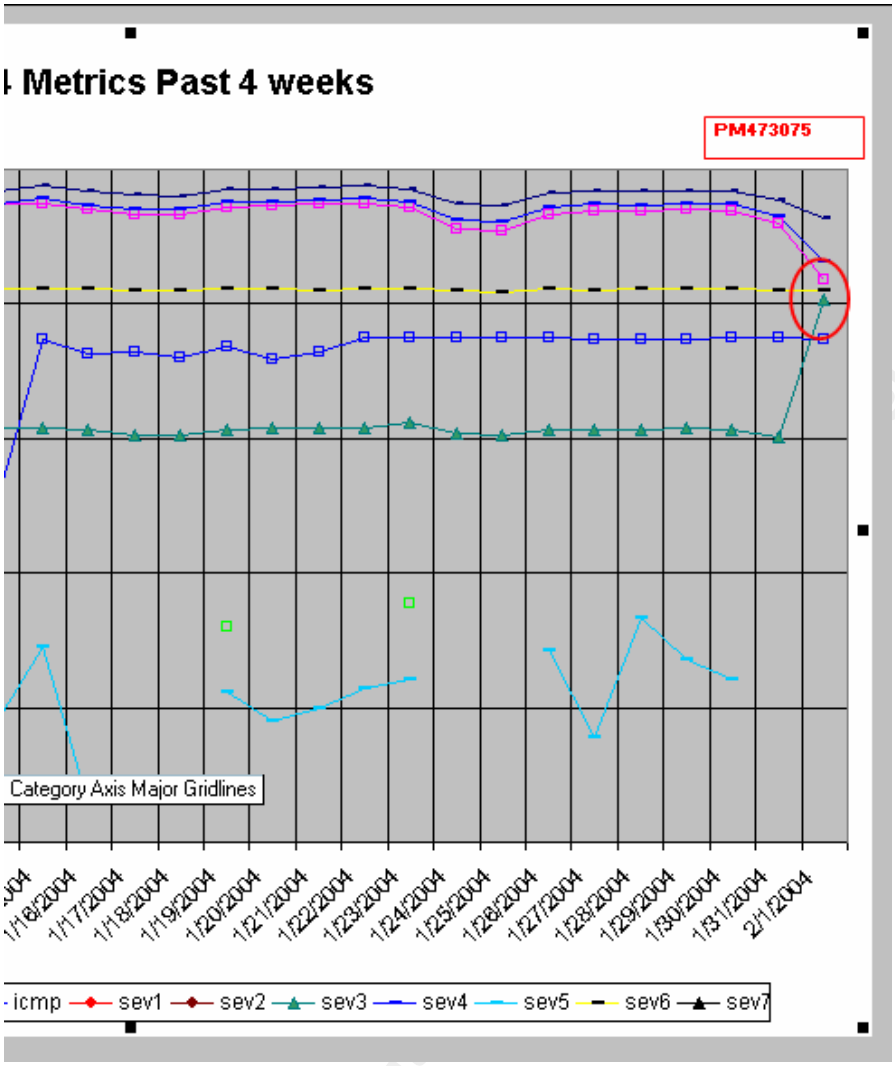
The other two reports can be looked at directly and research can be done to each item of data individually. For instance, with the top 10 denied IP report, it should be determined why the number one most denied IP is so popular. Is it an outside website that employees are attempting to visit? Is it an outside address that was attempting to port scan our network? Was it an internal address that is trying to communicate with an server that the firewall won't allow? If so, why is it trying to communicate with the other server? What port is it trying to use? Is this port also on the top 10 denied ports list?

The type and number of questions depends on the data given. Whether it is an inside or outside IP, whether it was the source address or the destination address, and what port or ports it was attempting to communicate with are all information that needs to be determined. As with the firewall report, The same type of research must be done on the top 10 denied reports.

4.6 Research on Anomalies

The research done on anomalies is basically filtering out from the logs what I need from that which I do not need. UNIX grep commands are a very good way to isolate the exact traffic you need to find.[9] They can either isolate lines that have a particular string of text, or they can isolate a section of lines that does not have a particular string of text, or a combination of both of these. You can create a rather lengthy command, but in the end you will have found what you are after. Below is an example of where log analysis would be performed. Please note that all IP's have been masked and any sensitive information has been taken out.

[9] McCarty, Bill "The Truth About Text" URL: http://www.linux-mag.com/2001-01/newbies_02.html (February 2, 2004)



Note in the graph, the green line with the triangle markers represent severity level 3 events. The increase from January 31 to February 1 raises a flag. Taking into account the logarithmic scale of the graph, the increase is almost 10 times the normal amount. For this amount of change in volume the difference in data should be fairly easy to determine. Sometimes the difference is simply an increase in the same traffic that was occurring the day before and the day before that. This type of increase is the most tedious as much analysis must be done before arriving at the determination that there was no new source of traffic.

Once the date and the type of traffic that needs to be isolated is determined, the next step is to access the log for analysis. I use a free telnet/ssh client called PuTTY. Putty allows me to access, via ssh, any UNIX-based box that I have an account on from my Windows OS desktop.[10] Once logged into the box where the logs reside, I can run commands to see what has changed from January 31 to

[10] "PuTTY: A Free Telnet/SSH Client" URL: http://www.chiark.greenend.org.uk/~sgtatham/putty/ (February 3, 2004)

February 1. I first determine the traffic that was present on January 31. I first access the logs for this particular firewall. I then run commands on the data from January 31 to determine what type of traffic was occurring that day. In this case it was simple. The traffic from January 31 was from one internal server trying to contact another internal server via udp port 137 (Netbios) unsuccessfully. The command
gunzip –c messages.200402010408 |grep PIX-3-|more
results in the following: (only an indicative section)

```
Jan 31 04:22:47 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.94/137 dst inside:XXX.XX.179.197/137
Jan 31 04:22:48 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.94/137 dst inside:XXX.XX.179.197/137
Jan 31 04:22:50 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.94/137 dst inside:XXX.XX.179.197/137
Jan 31 04:22:52 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.94/137 dst inside:XXX.XX.179.197/137
Jan 31 04:22:52 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.94/137 dst inside:XXX.XX.179.197/137
Jan 31 04:24:58 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.94/137 dst inside:XXX.XX.179.197/137
Jan 31 04:24:59 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.94/137 dst inside:XXX.XX.179.197/137
Jan 31 04:25:01 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.94/137 dst inside:XXX.XX.179.197/137
Jan 31 04:25:03 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.94/137 dst inside:XXX.XX.179.197/137
Jan 31 04:25:04 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.94/137 dst inside:XXX.XX.179.197/137
```

This command responds with the lines of the log that are severity 3 messages from the January 31 log. From the full results, I note that the lines above make up all of the severity 3 lines of the log. If I wanted to guarantee this I could use the "–v" option of grep on the IP returned above. This allows grep to not include the IP you are already aware of. This can be repeated until there is a good understanding of what the log has to say. The end command for this exception was-
gunzip –c messages.200401310408 |grep PIX-3-|grep –v XXX.XX.81.94|more
which returns nothing, as expected.

In this situation, the given firewall does not have a great amount of traffic and these results are not considered normal log activity. In other words, most logs have many issues causing a specific severity message, not just one. In those cases, more filtering is necessary to find all of the constituents of the severity level totals.

Once all of the IP's are determined, the same analysis can be done to the data from the day in question, in this case February 1. The command-

gunzip –c messages.200402010408 |grep PIX-3-|grep –v XXX.XX.81.94|more

returns the following:

```
Feb  1 04:03:20 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.154/137 dst inside:XXX.XX.179.197/137
Feb  1 04:03:21 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.154/137 dst inside:XXX.XX.179.197/137
Feb  1 04:03:23 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.154/137 dst inside:XXX.XX.179.197/137
Feb  1 04:03:24 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.154/137 dst inside:XXX.XX.179.197/137
Feb  1 04:03:26 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.154/137 dst inside:XXX.XX.179.197/137
Feb  1 04:03:27 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.154/137 dst inside:XXX.XX.179.197/137
Feb  1 04:03:29 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.154/137 dst inside:XXX.XX.179.197/137
Feb  1 04:03:30 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.154/137 dst inside:XXX.XX.179.197/137
Feb  1 04:03:32 misu04 %PIX-3-305005: No translation group found for udp src dmz1:XXX.XX.81.154/137 dst inside:XXX.XX.179.197/137
```

It can be concluded that this new IP is the cause of the new traffic. Since it might not be the only one, the process of elimination could be used again to determine this.

gunzip –c messages.200402010408 |grep PIX-3-|grep –v XXX.XX.81.94|grep –v XXX.XX.81.154|more

Which returns nothing as expected.

Since the two boxes are both internal and it is obviously a configuration error, this issue is not a security risk and therefore can be documented in a low severity work order ticket in order to be fixed. The servers in question either need to be reconfigured or a rule on the firewall is incorrect.

5. After

The monitoring team started with only firewall logs and created a simple Intrusion Detection System as well as a network error detection system. This process allowed for a way to baseline, analyze and use the data that had always been collected to determine when something was wrong whether inside the network or coming from the internet. The procedures are not to detect everything, it is not a full function IDS system, but it is a very good start at monitoring the network for anomalies.

Looking back at how there was no process in place for log analysis, the process described here is simple yet quite remarkable. The monitoring team is now more aware of the traffic that flows through the firewalls every day, and now has an opportunity to start making sure that traffic is legitimate and not just causing increased network latency. There is a way to track infected boxes, and servers that are not configured correctly. The firewall rulesets can be measured for effectiveness and changed according to arising needs. The Information Security Team will benefit from this analysis in many ways.

Recently it was discovered that the logs could be monitored on a more granular level by filtering out each individual message type rather than by severity message level. While there would be intensive work to get the scripts changed around, have the data formatted on new reports and the database updated to accept the new data, the analysis may run more smoothly and problems could be pinpointed more accurately.
This and other improvements may happen to this process down the road, but for now the monitoring team is working to get the traffic cleaned up so they can be more equipped to handle an incident. This would not have been possible without the Firewall log analysis project.

11. References

1. Puget Sound Technology, “Glossary of open source, BSD, Linux, Unix, networking, and free software terminology” URL: http://www.pugetsoundtechnology.com/info/definitions/definitions.html (November 20, 2003)

2. Cisco Systems “Messages listed by Severity Level-Cisco PIX Firewall Software” URL: http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_system_message_guide_chapter09186a008008d278.html (December 5, 2003)

3. Cisco Systems “System Log Messages-Cisco PIX Firewall Software” URL: http://www.cisco.com/en/US/products/sw/secursw/ps2120/products_system_message_guide_chapter09186a008008d275.html#24101 (December 5, 2003)

4. Maso, Brian “Track Performance of Distributed Systems” URL: http://archive.devx.com/free/articles/2000/maso01/maso01-1.asp (January 2, 2004)

5. Gailly, Jean-loup “The Gzip home page” URL: http://www.gzip.org/ (January 3, 2004)

6. “Perl Synopsis” URL: http://www.perldoc.com/perl5.6/pod/perl.html (January 20, 2004)

7. “Cron” URL: http://www.ss64.com/bash/cron.html (January 21, 2004)

8. Reavis, Jim “Is your Internet firewall a bottleneck?” 6 September, 1999 URL: http://www.nwfusion.com/newsletters/sec/0906sec1.html (January 25, 2004)

9. McCarty, Bill “The Truth About Text” URL: http://www.linux-mag.com/2001-01/newbies_02.html (February 2, 2004)

10. “PuTTY: A Free Telnet/SSH Client” URL: http://www.chiark.greenend.org.uk/~sgtatham/putty/ (February 3, 2004)