Interested in learning more about security?

SANS Institute InfoSec Reading Room

Cisco Pix Log Analysis In a University Setting

Cisco Pix Log Analysis
In a University Setting

GIAC Gold Certification

Author: Jack Vant

Adviser: Pedro Bueno

Table of Contents

List of Figures

List of Tables

Abstract

This paper describes a study I conducted over a period of two months which attempted to determine whether an IDS system is necessary for one subnet on campus which is currently protected by a Cisco PIX firewall. PIX logs were collected and analyzed using two different reporting tools, Fwanalog and Sawmill. These tools provided summary statistics for the number of exceptions (violations of the PIX rules) generated daily as well as a breakdown of the violation by source host address, port, and destination host address. A list of the top 20 hosts which caused these exceptions was compiled and used to search the logs. These analyses revealed that the greatest threat currently resides on our campus network. The study determined that the university should install an IDS to monitor the subnet protected by the Cisco PIX. The study also tested two logging solutions for university routers, a solution based on a Linux server and a solution that utilized a Windows 2003 server running on VmWare ESX and third-party software called Kiwi which provides syslog services on a Windows platform.

Introduction

This project was inspired by two concepts that resonated with me during the SANS General Security class. First, prevention alone will not protect your assets. Sooner or later a mistake will be made. Someone will forget to shut off a service, or an important patch will be overlooked, and the game will be over. And so it is necessary to know who is trying to gain access to systems ahead of time, which is why IDS is important. The place to start with IDS is with router logs, which is the second concept that resonated. These logs can be used to get a feel for what's out there on the network and where IDS is needed most.

Purpose

The objectives for the project were as follows:

1) To determine whether an IDS system is necessary for the subnet that contains the Unix systems I administer.
2) To capture and analyze the logs for the Cisco PIX firewall that protects university resources.
3) To identify potential threats to key university resources in terms of location and types.
4) To provide a workable logging solution for campus routers that is inexpensive and requires a minimum of maintenance and administrative time.

Scope

This study deals with the network traffic on one subnet of the university network. It is limited to the traffic on our PIX firewall.

Before I proceed I should mention that this paper is not a how to paper. I do describe how I configured programs and installed the tools used for this study, but the focus of this paper is analysis of the information the tools provided. I used a couple of how to papers during this study, and they are mentioned in the text.

I should also mention that in some cases I have not provided ip numbers of machines, and in some of the figures provided in this document I have struck the numbers out to avoid revealing potentially sensitive information.

Methodology

There were 4 phases for the project:

1. Log setup and collection
2. Tool survey and selection
3. Report generation with selected tools
4. Analysis of reports and log analysis

Log Setup and Collection

PIX Configuration

We tested several of the logging levels prior to collecting the data to give us an idea of log size. We had no idea how big the logs might be.

Cisco defines eight log levels. Actually there are only 7 functional levels, since 0 isn't used. Cisco describes them as follows:

Level Number	Level Keyword	Description
0	**emergency**	System unusable.
1	**alert**	Immediate action needed.
2	**critical**	Critical condition.
3	**error**	Error condition.
4	**warning**	Warning condition.
5	**notification**	Normal but significant condition.
6	**informational**	Informational message only.
7	**debugging**	Appears during debugging only.

Table 1 – Log Message Severity Levels
(http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_sw/v_63/63syslog/pixemint.htm#wp1029160)

The Cisco default log level is 3 for error. We experimented with levels 4 through 7 to get a feel for the size of the logs. We determined we had enough space to log at the highest level for at least three weeks.

We did not select this level. First, we looked at the Cisco documentation for the PIX (located at http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_sw/v_63/63syslog/pixemapa.htm#wp1004080) which offered a nice summary of what we could expect to see at the various levels. Our Network Engineer did not think it was necessary to capture the router information contained in level 7 messages, since he debugged the router when he set it up. And so we logged at level 6, the informational level which in effect captures messages for levels 1 through 6. At level 6 the logs averaged 351 megabytes daily. This level captures connections between the PIX and the logging server, tcp and udp connections made and established on the network and all exceptions to the rules for the PIX.

Configuring the PIX to log to a remote host is straightforward. Our network engineer issued two commands to do this. The first was **logging trap 6** which set the log level. Then he pointed the logs to the remote host: **logging host inside $loghostip**.

PIX messages are easy to identify and work with. Each message begins with a % followed by the severity level, the message number and then a message description. So a typical message looks like this:

%PIX-6-302016: Teardown UDP connection 70895209 for outside:XXX.XXX.XXX.XXX/53 to inside:XXX.XXX.XXX.XXX/32 904 duration 0:00:01 bytes 96

In this case a message with a severity level of 6 (informational) has been received. The message number is 302016 and the message text indicates that a udp connection has been torn down.

Log Server Configuration

The logs were collected from February 17th to March 22nd on an old Linux server that had a fresh install of Fedora core 3. The setup for logging was quite straightforward. I added the following line to the syslog.conf file on the server:

local6.* /var/log/pix.log

Linux servers do not accept los from remote hosts by default, so the –r option must be added to the syslog startup. I edited the syslog file in /etc/rc.d/init.d. There's a line in the startup file that defines the startup options. I added a –r so that the line read:

SYSLOGD_OPTIONS="-r -m 0"

I restarted syslogd, and logging began. The logrotate facility was used to roll the logs every night. To do this I created a file in the /etc/logrotate.d directory called pix. The pix file contains the following instructions for logrotate:

```
/var/log/pix.log {
      daily
      missingok
      rotate 28
      compress
      delaycompress
      notifempty
      create 660 root root
      sharedscripts
   postrotate
      /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null || true
   endscript
}
```

Figure 1 – PIX script

The instructions tell logrotate to roll the log daily for 28 days. The logs are compressed with a one day delay for compression. If the log is empty, it isn't rotated. The script also creates the new log with the proper permissions and then restarts syslogd. This setup performed flawlessly, and since it used Fedora core and a retired machine it cost nothing.

The logs were collected on a Windows 2003 server from March 23rd until April 9th. The Windows 2003 server was installed on a VMWare ESX server by one of our Microsoft engineers. A 20 gigabyte volume was added to the system to contain the logs. The Kiwi syslog program was installed on the Windows server. I read about this program in Ben Carlsrud's paper **Cisco Pix: Logging and beyond** .

I installed it to run as a service on the server. It can also be installed as an application. Kiwi has a built in viewer and allows the user to rotate and name logs quite easily. Once it's installed, selecting file and then setup takes you to the Kiwi setup screen where you can tweak settings to your liking.

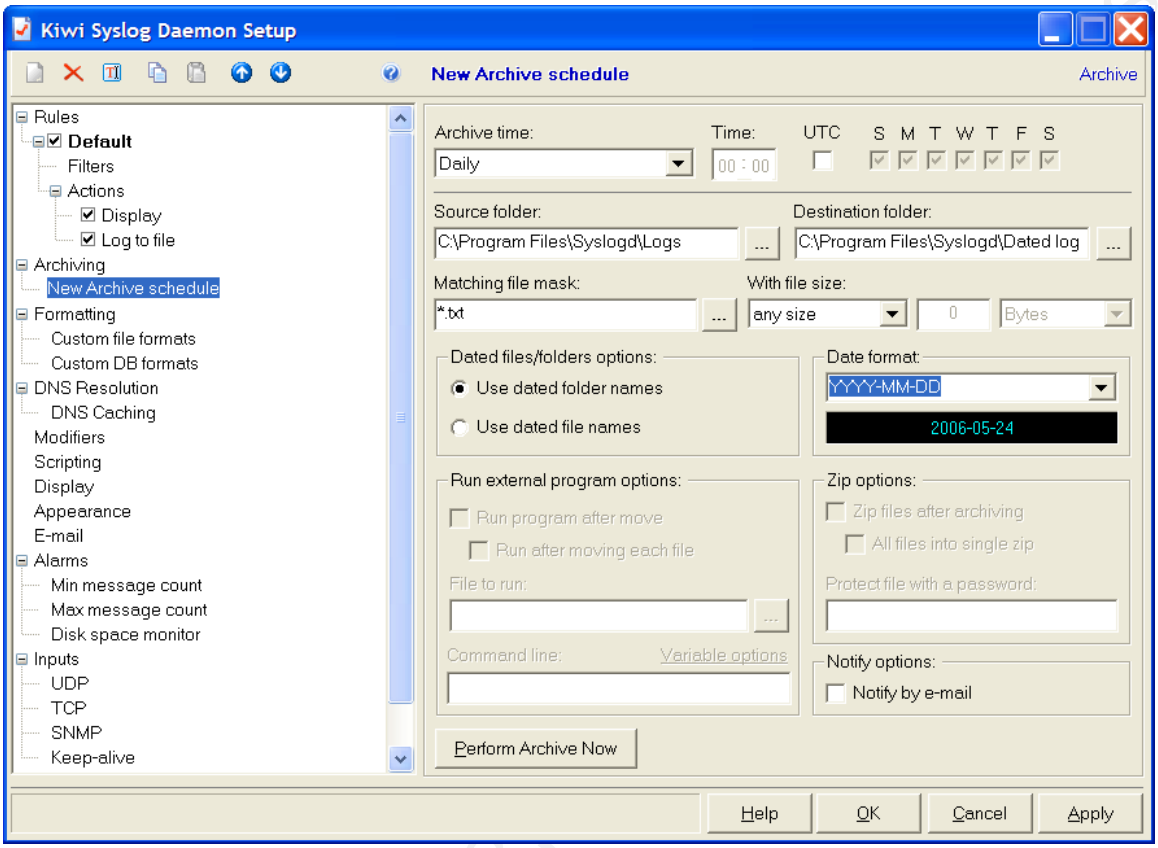

Figure 2 – Kiwi setup screen

The only setting change I made for my installation involved log archiving. I specified a folder on the 20 gig drive for my archived logs.

Kiwi syslog costs $100.00. It can be downloaded from http://www.kiwisyslog.com/index.php. The educational price for the standard version of Server 2003 costs $130.00. The cost of equipment is difficult to calculate. Our ESX team estimates that each VM server costs approximately $1500.00 once SAN storage and memory and CPU usage is factored in. The total cost of this solution is roughly $1730.00. This is still less expensive than the $5,000.00 we typically pay for a new server.

Tool Survey and Selection

While the logs were piling up, I went looking for products to analyze them. I had foolishly thought that I could simply grep through the logs for the PIX message numbers with a few simple scripts. After a day of this, I realized the error of my ways and went looking for some tools that I could use.

My criteria for selection were simple: it had to be free or very cheap, and it had to produce a simple and readable report that would give me summaries for exceptions to the firewall rules, a list of hosts associated with the exceptions and the port or ports these hosts were trying to access.

For analysis of the logs on the Linux platform I found perl scripts and more complex products with web front ends and databases. Some were free and others—typically the products with web front ends and database products—cost thousands. I settled on a free product, Fwanalog, which made use of another free product, analog. Fwanalog can be downloaded from http://tud.at/programm/fwanalog/. Its companion program can be downloaded from http://www.analog.cx/.

For the Windows/VM istallation I chose a product that I had some familiarity with. I tried a couple of free products first, and they provided nice daily reports, but when asked to process a week's worth of reports they were unable to handle the volume. Sawmill was my choice for the analysis product on the Windows platform. I had worked with it before when I was asked to analyze some Groupwise logs. I was reminded of it when I read Mark Lachniet's paper, **Inexpensive Cisco Log Analysis**. Sawmill is much more fully featured than Fwanalog, and you do pay for these features. I used a free 30 day trial version for this project. The lite version of Sawmill costs $99.00 for one profile and $249.00 for 5 profiles. (Sets of logs are analyzed by profile.) Sawmill is cheap enough that it is a feasible solution to propose to my managers. I downloaded Sawmill from http://www.sawmill.net/.

Report Generation

Install and report generation with Fwanalog took about an hour. I installed the analog rpm package first with **rpm –ivh analog-6.0-1.i686.rpm**. Fwanalog was easy to install. I untarred it and then edited the configuration file fwanalog.opts.pix and renamed it to fwanalog.opts.

```
#!/bin/sh

##########################################################################
#
#       User-changeable options for fwanalog.sh
#
#       $Id: fwanalog.opts.pix,v 1.4 2003/11/25 17:11:31 bb Exp $
#
##########################################################################
outdir="/archive/logdaily"
# The directory where the output goes to, without / at the end. You need write
# permissions, of course, and should secure this directory with permissions,
# minefields, guard dogs etc. It will be created if you don`t have it yet.

logformat="pix"
# What log format your firewall writes.
# Currently available options:
#       iptables        Linux 2.4 iptables              (probably in /var/log/messages)
#       ipchains        Linux 2.2 ipchains              (probably in /var/log/messages)
#       ipf                     BSD/Solaris ipfilter    (probably in /var/log/ipflog)
#       openbsd         this was the same as ipf until OpenBSD 2.9; this also
#                                       seems to work on NetBSD
#       freebsd         FreeBSD`s output format (probably in /var/log/ipflog)
#       solarisipf      Solaris 8.0 Intel ipf 3.4.20 (using ipmon -sn &)
#       pf_30           OpenBSD 3.0 pf binary log format
#                                       fwanalog *must* run on OpenBSD 3.0 for this to work
#                                       (because of the special tcpdump of OpenBSD)
#       zynos           ZyNOS (ZyXEL, Netgear) logfile
#       pix                     Cisco Pix (tested with version 6.22/IOS)
#       watchguard      Watchguard Firebox
#       fw1                     Checkpoint Firewall-One (not fw-1 NG!)

# Feel free to program a parser for your firewall if it is not supported.
# See the comments in iptables() and ipf()
#
# The officially maintained formats are pf_30 and iptables.

inputfiles_mask="pix.log.*"     # The name of your logfiles, with a wildcard if you want
inputfiles_dir="/archive/logdaily"      # The directory where your logfiles are in,
                                                                #       e.g. /var/log
inputfiles_mtime="31"           # How old the logfiles can be
# You can change this to your log rotate interval + 1 day (so you never miss a logfile entry)
inputfiles=`find $inputfiles_dir -maxdepth 1 -name "$inputfiles_mask" -mtime -$inputfiles_mtime | sort -r`
# This should find the names of the logfiles you want to parse
# It MUST return the names in reverse order (chronologically) or you
# will have LOTS of duplicate lines in your log.

onehost=false
# Available options: false true dynip

# Default: false

# Set to true if this firewall runs on one machine only and you want to see
# the source hosts (not the protected target hosts) in the Blocked Packet
# Report. This is suggested if you protect one server, but loses information
# if you protect a network.
```

Figure 3 – fwanalog.opts

I changed the outdir, inputfiles_mask, and inputfiles_dir variables. I ran fwanalog.sh and it processed 10 gigs of logs in 50 minutes.

The report generated by fwnanalog offers a nice summary of the number of blocked packets, the average per day, the top twenty-five blocked hosts, and a nice breakdown by destination hosts and ports.

Listing blocked packets, sorted by the number of blocked packets.

#blocks	%blocks	bytes	last time	blocked packet
4614	3.05%	0	Mar/22/06 06:38	[redacted]
4327	2.86%	0	Mar/22/06 06:38	[redacted]/icmp
4327	2.86%	0	Mar/22/06 06:38	[redacted]:0/icmp
287	0.19%	0	Mar/22/06 00:55	[redacted]/tcp
176	0.12%	0	Mar/13/06 18:20	[redacted]:ms-sql-s (1433)/tcp
74	0.05%	0	Feb/28/06 10:57	[redacted]:microsoft-ds (445)/tcp
20	0.01%	0	Feb/27/06 23:55	[redacted]:loc-srv (135)/tcp
16	0.01%	0	Feb/26/06 20:01	[redacted]:netbios-ssn (139)/tcp
1		0	Mar/22/06 00:55	[redacted]:13021/tcp

Figure 4 – fwanalog blocked packet report (host ip removed)

I found this part of the report especially interesting, since I run Unix servers that don't provide any Windows services.

The installation of Sawmill was a bit more involved. The Windows installation is obvious enough. You have to accept the license agreement and decline email offers. The interface for the program is your browser and the address is 127.0.0.1:8987. The program guides you through the creation of your first set of reports. The first screen asks for an admin id and password.

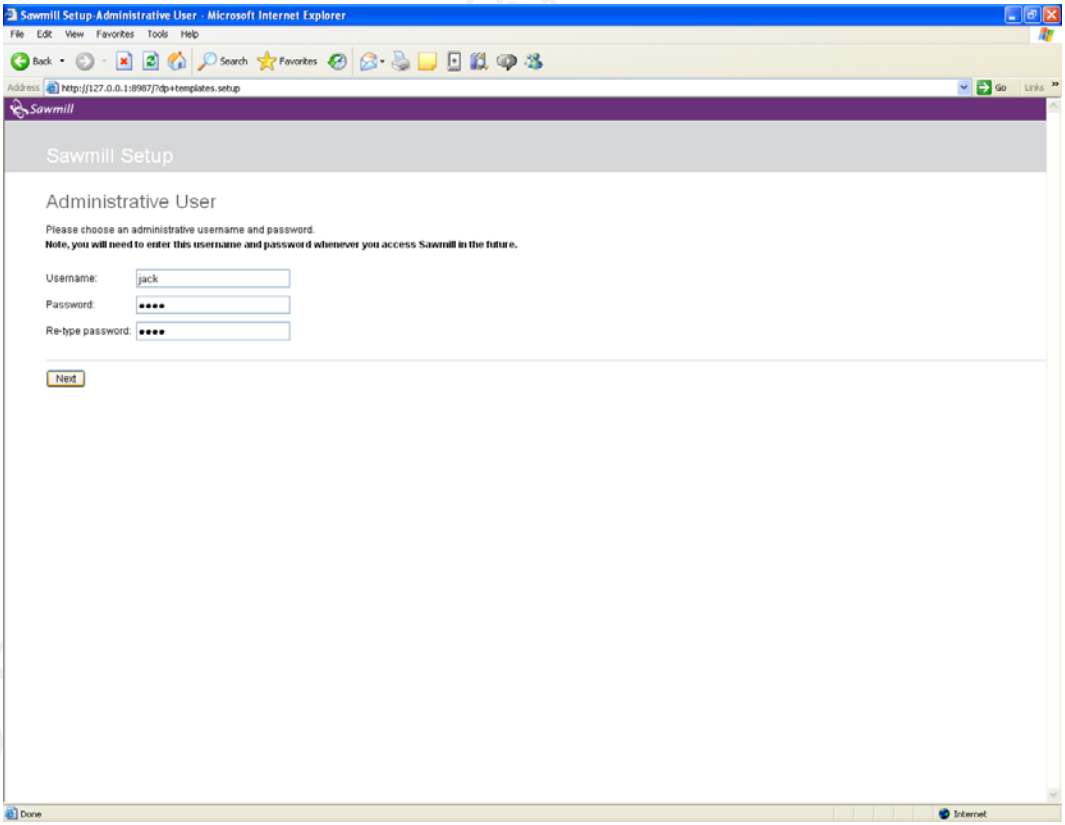

Figure 5 – Sawmill Admin user screen

When you click next, you are asked to specify the source for log files.

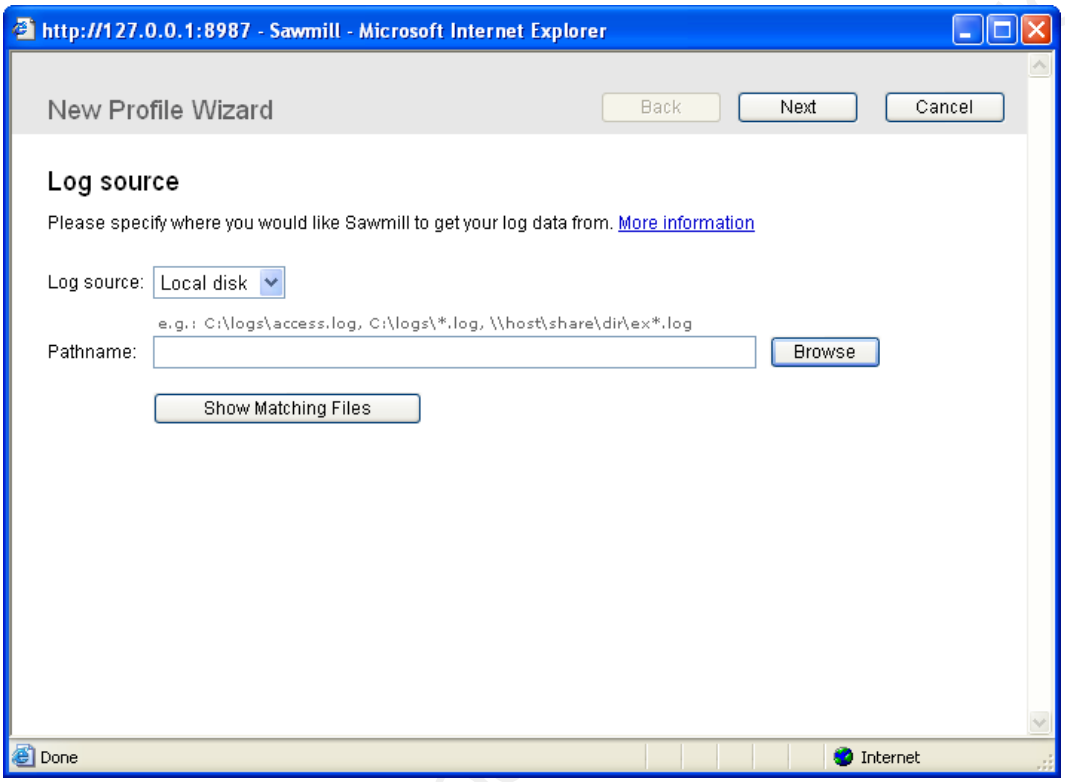

Figure 6 – Sawmill log source screen

Once you click next, the program identifies the log format.

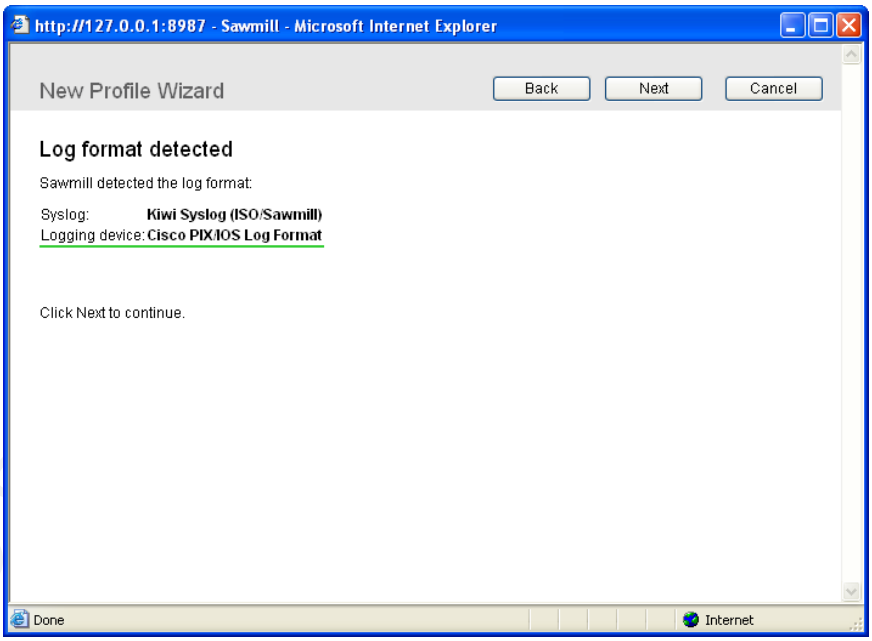

Figure 7 – Sawmill log format detected

When you click next, the program processes the logs. This took 2 hours and 55 minutes in my case, which may seem like terrible performance, but I was running the program on my laptop while I was doing 5 other things. The last step asks you to name the profile and then you are presented with your reports.

Unlike the fwanalog reports, the Sawmill reports allow you to drill down a bit and examine events. There are more reports than I can possibly go into here. The report called Single-page summary provided all the information that I needed.

The reports generated from the two products provided me with summary statistics and lists of destination hosts, source hosts and ports. The Sawmill summary report also provided me with a list of Cisco message codes by frequency. I looked up the top 20 hosts in the logs and examined the incidents as they appeared in the logs. I simply used grep and the ip number of the host and redirected the output to a file, **grep 10.1.1.10 * > 10.1.1.10.txt**. I looked up the messages in the Cisco documentation for an explanation of the message. I then attempted to group the exceptions and generate some categories. Most of the message codes appeared during these lookups; however, some of the codes on the list did not appear in the host lookups so I was forced to look these up as well.

The log reports then provided a way into the logs. Admittedly, this method was a bit primitive, but it served my purposes.

Results

The top ten Cisco PIX messages are presented in Table 2 below. This list was generated by the Sawmill report.

Rank	Message Code	Event #	Percentage
1	PIX-4-106023	20,942	55.2
2	PIX-3-305005	13,296	35.0
3	PIX-6-302016	1,564	4.1
4	PIX-6-302015	1563	4.1
5	PIX-6-302014	256	.7
6	PIX-6-302013	247	.7
7	PIX-6-106015	50	.1
8	PIX-5-111008	4	0
9	PIX-6-303002	4	0
10	PIX-5-111004	2	0

Table 2 – Top Ten PIX messages March 23 through April 9

Table 3 provides an explanation of these messages.

Message	Cisco Explanation
PIX-4-106023	An IP packet was denied by the **access-list**.
PIX-3-305005	A packet does not match any of the outbound **nat** rules.
PIX-6-302016	A UDP connection slot between two hosts was deleted.
PIX-6-302015	A UDP connection slot between two hosts is created.
PIX-6-302014	A TCP connection between two hosts was deleted.
PIX-6-302013	A TCP connection slot between two hosts was created.
PIX-6-106015	This message is logged when the PIX Firewall discards a TCP packet that has no associated connection in the PIX Firewall unit's connection table. PIX Firewall looks for a SYN flag in the packet, which indicates a request to establish a new connection. If the SYN flag is not set, and there is not an existing connection, the PIX Firewall discards the packet.
PIX-5-111008	This syslog message is for accounting purposes. The user entered a command that modified the configuration.
PIX-6-303002	This is an FTP/URL message. This message is logged when the specified host successfully stores or retrieves data from the specified FTP site.
PIX-5-111004	This message is logged when you enter the **config floppy/memory/ network** command, or the **write floppy/memory/network/standby** command. The *IP_addr* indicates whether the login was made at the console port or via a Telnet connection.

Table 3 – PIX message explanations

The top ten services that caused the router to log events are presented in table 4 below.

Rank	Service Name	Event #	Event %
1	Ms-sql-s	26,575	41.8
2	epmap	12,021	18.9
3	Microsoft-ds	11,761	18.5
4	http	11,224	17.7
5	Netbios-ssn	1,973	3.1
6	13010_TCP	3	0
7	22_TCP	2	0
8	61100_TCP	2	0
9	48100_TCP	2	0
10	59100_TCP	2	0

Table 4 – Top ten services

I used the Sawmill and Fwanalog reports to generate a top twenty list of machines that produced events on the PIX. Both reports contain number of blocks per host, so this was a simple task. All of the events originated on campus. I won't present the top 20 here, simply because I cannot give away details about machines on our network, but I can

make some general observations. Several machines from labs on campus showed up in the list. Enough so that it's probably time to go talk to the people that run the Physics lab on campus. One machine from our wireless network showed up in the list, and a number of machines from offices around campus and the campus dormitories showed up on this list.

According to the Fwanalog report, we averaged 4,582 blocked packets per day for the time period between February 17th and March 22nd. Since Sawmill views all traffic as events, it was a little more difficult to come up with an average using this report for blocks. I estimated this by using the two most common operations, deny and no translation group, which were listed on a separate operations report. Both of these operations result in blocks. The average using these two numbers was 6,394 for 18 days.

Discussion

The first thing that jumps off the statistics pages are the sheer number of events or blocks that are attributed to Microsoft services, ports 1433, 445, 139 and 135. There is no reason for this traffic on this subnet. Granted the PIX is configured so that it rejects these packets without fail, but the presence of this traffic on this part of the network and the volume should serve as a reminder of what transpired when the worms propagated on campus last summer, producing what was essentially a denial of service on many parts of the campus network. We do not appear to be any better off than we were last summer when we saw widespread network congestion.

I also found the http traffic quite curious. When I used Sawmill to drill down a little further to investigate this, it blamed this behavior on a network appliance, which is very curious. Yet another question for the Network Engineer.

We also discovered two of the top hosts generating blocks or events were two of our own network devices. This is a misconfiguration that will have to be cleared up in the near future.

After examining the logs for the top twenty, I managed to generate some event or block types. The categories I generated are as follows: 1) misconfigured network devices; 2) simple blocks which are indicated with PIX message PIX-4-106023; 3) blocks with the PIX message PIX-3-305005; and 4) activity that indicates some packet fragmentation, which is suggested by PIX message PIX-6-106015.

The vast majority of blocks were 106023 message codes where the router simply says you're not allowed. Many of these came from the same machines over and over again usually attempting to connect to hosts real or imagined on the subnet on one of the Microsoft service ports. I talked to one of our Microsoft engineers about these machines and he attributed most of this activity to unpatched machines that were infected with a worm, virus or even a bot of some kind. I've already talked about traffic for these services on this subnet.

The last two types of messages are cause for concern. In looking through the logs for the top twenty, I came across a number of instances of message code PIX-3-305005 and I stumbled across this sequence.

```
pix.log.15:Mar  8 08:17:42 XXX.XXX.XXX.XXX Mar 08 2006 09:59:02: %PIX-6-302013: Built
inbound TCP connection 41073293 for outside:XXX.XXX.XXX.XXX/1656 (XXX.XXX.XXX.XXX
/1656) to inside: XXX.XXX.XXX.XXX /445 (XXX.XXX.XXX.XXX /445)
pix.log.15:Mar  8 08:17:48 XXX.XXX.XXX.XXX Mar 08 2006 09:59:07: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX /2681 dst inside:
XXX.XXX.XXX.XXX /445
pix.log.15:Mar  8 08:17:51 XXX.XXX.XXX.XXX  Mar 08 2006 09:59:11: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX /1409 dst inside:
XXX.XXX.XXX.XXX /445
pix.log.15:Mar  8 08:17:52 XXX.XXX.XXX.XXX Mar 08 2006 09:59:12: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX /4544 dst inside:
XXX.XXX.XXX.XXX /445
pix.log.15:Mar  8 08:17:53 XXX.XXX.XXX.XXX Mar 08 2006 09:59:13: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX /4768 dst inside:
XXX.XXX.XXX.XXX /445
pix.log.15:Mar  8 08:38:56 XXX.XXX.XXX.XXX 1 Mar 08 2006 10:20:16: %PIX-6-302014:
Teardown TCP connection 41106392 for outside: XXX.XXX.XXX.XXX /1523 to inside:
XXX.XXX.XXX.XXX /445 duration 0:00:00 bytes 0 TCP Reset-I
pix.log.15:Mar  8 08:38:58 XXX.XXX.XXX.XXX Mar 08 2006 10:20:17: %PIX-6-302013: Built
inbound TCP connection 41106434 for outside:1 XXX.XXX.XXX.XXX /2224 (XXX.XXX.XXX.XXX
/2224) to inside: XXX.XXX.XXX.XXX /445 (XXX.XXX.XXX.XXX /445)
pix.log.15:Mar  8 08:39:11 XXX.XXX.XXX.XXX Mar 08 2006 10:20:30: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX /3398 dst inside:
XXX.XXX.XXX.XXX /445
pix.log.15:Mar  8 08:39:12 XXX.XXX.XXX.XXX Mar 08 2006 10:20:32: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX /4564 dst inside:
XXX.XXX.XXX.XXX /445
pix.log.15:Mar  8 08:39:18 XXX.XXX.XXX.XXX Mar 08 2006 10:20:37: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX /2680 dst inside:
XXX.XXX.XXX.XXX /445
pix.log.15:Mar  8 08:39:34 XXX.XXX.XXX.XXX Mar 08 2006 10:20:54: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX /3698 dst inside:
XXX.XXX.XXX.XXX /445
pix.log.15:Mar  8 08:39:44 XXX.XXX.XXX.XXX Mar 08 2006 10:21:03: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX /3606 dst inside:
XXX.XXX.XXX.XXX /445
pix.log.15:Mar  8 08:39:44 XXX.XXX.XXX.XXX  Mar 08 2006 10:21:04: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX 113/1860 dst inside:
XXX.XXX.XXX.XXX /445
pix.log.15:Mar  8 08:39:45 XXX.XXX.XXX.XXX 1 Mar 08 2006 10:21:04: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX /2541 dst inside:
XXX.XXX.XXX.XXX /445
pix.log.15:Mar  8 09:40:11 XXX.XXX.XXX.XXX Mar 08 2006 11:21:29: %PIX-6-302014:
Teardown TCP connection 41200530 for outside: XXX.XXX.XXX.XXX /2856 to inside:
XXX.XXX.XXX.XXX /1433 duration 0:02:01 bytes 0 SYN Timeout
pix.log.15:Mar  8 15:06:52 XXX.XXX.XXX.XXX Mar 08 2006 16:48:06: %PIX-3-305005: No
translation group found for tcp src outside: XXX.XXX.XXX.XXX /4062 dst inside:
XXX.XXX.XXX.XXX /445
```

Figure 8 - PIX-3-305005

It appears that a host did manage to penetrate the PIX firewall. See the text in bold above. The host in question resides in a computer lab on campus. This will be reported to our Security Officer in our Central Audit Department.

As for PIX message PIX-6-106015, the activity indicates some Microsoft servers and one workstation in our financial area are attempting connections on various ports with very mixed results. Here's a sample of this traffic.

```
pix.log.1:Mar 22 07:18:42 XXX.XXX.XXX.XXX Mar 22 2006 08:55:08: %PIX-6-106015: Deny
TCP (no connection) from XXX.XXX.XXX.XXX /636 to XXX.XXX.XXX.XXX/44829 flags FIN PSH
ACK  on interface outside
pix.log.1:Mar 22 07:32:47 XXX.XXX.XXX.XXX Mar 22 2006 09:09:12: %PIX-6-106015: Deny
TCP (no connection) from XXX.XXX.XXX.XXX /45093 to XXX.XXX.XXX.XXX /636 flags RST  on
interface inside
pix.log.1:Mar 22 07:56:19 XXX.XXX.XXX.XXX Mar 22 2006 09:32:44: %PIX-6-106015: Deny
TCP (no connection) from XXX.XXX.XXX.XXX /47284 to XXX.XXX.XXX.XXX /636 flags RST  on
interface inside
pix.log.1:Mar 22 08:03:59 XXX.XXX.XXX.XXX Mar 22 2006 09:40:24: %PIX-6-106015: Deny
TCP (no connection) from XXX.XXX.XXX.XXX /45744 to XXX.XXX.XXX.XXX /636 flags RST  on
interface inside
pix.log.1:Mar 22 08:30:35 XXX.XXX.XXX.XXX Mar 22 2006 10:07:00: %PIX-6-106015: Deny
TCP (no connection) from XXX.XXX.XXX.XXX /46527 to XXX.XXX.XXX.XXX /636 flags RST  on
interface inside
pix.log.1:Mar 22 08:45:52 XXX.XXX.XXX.XXX Mar 22 2006 10:22:16: %PIX-6-106015: Deny
TCP (no connection) from XXX.XXX.XXX.XXX /2501 to XXX.XXX.XXX.XXX /53109 flags PSH
ACK  on interface outside
pix.log.1:Mar 22 08:47:32 XXX.XXX.XXX.XXX Mar 22 2006 10:23:56: %PIX-6-106015: Deny
TCP (no connection) from XXX.XXX.XXX.XXX /2612 to XXX.XXX.XXX.XXX /53109 flags PSH
ACK  on interface outside
pix.log.1:Mar 22 08:48:34 XXX.XXX.XXX.XXX Mar 22 2006 10:24:58: %PIX-6-106015: Deny
TCP (no connection) from XXX.XXX.XXX.XXX /2735 to XXX.XXX.XXX.XXX /53103 flags PSH
ACK  on interface outside
```

Figure 9 - PIX-6-106015

I think it will take a while to sort out the hosts and ports indicated in these messages. I see LDAP frequently which is yet another unnecessary service for this subnet. The Cisco documentation for this message warns that if "the firewall receives a large volume of these invalid TCP packets" then it's necessary to "trace the packets to the source and determine the reason these packets were sent." In other words, keep an eye on this traffic. I'll probably start my investigation with lsof on some of my hosts.

Conclusions

The best logging solution for our purposes is a Linux machine running on a VMware ESX server and running syslogd and Secure Shell. The best reporting software for our purposes is Sawmill. So for $99 we can at least log and generate daily reports that might help us intervene in a more timely fashion.

I'm confident that we can sort out the problems with the two network devices mentioned in the report. The Network Engineer is aware of this problem, and we have to sit down and look at the logs. Probably the biggest problem currently is unnecessary network traffic: Microsoft-sql, Microsoft-ds, epmap, netbios-ssn, http, and ldap. This traffic should never reach the subnet.

I think an IDS is in order for the subnet I work on. There are a number of reasons that I say this. First, one computer in a physics lab did defeat the PIX. I was lucky that I found this, and I found it 2 weeks after it happened. It appears that no harm was done. The connection was torn down by the PIX, but I find little comfort in that. Second, the machines on this subnet are the most important on the university network. They represent a sizable investment and they contain valuable information. The PIX affords some protection, but it is apparently vulnerable. We have also believed and incorrectly that we could do security through obscurity by "hiding" important machines on a specific network. It is evident that the subnet in question is no mystery to hosts generating the exceptions. These machines know exactly what hosts are on the network. And finally, our environment is simply too open and our organization does not have control of desktop and lab computers in other departments. The top 20 hosts were all computers on campus. Four of the top twenty were located in a physics lab where we have no control. There are several labs on campus where this is the case. We have thought—and incorrectly—that the greatest danger is out there on the internet when it is right here on campus. Another layer of security is necessary to protect valuable university resources, and that layer is an IDS system.

References

Carlsrud, Ben. **Cisco Pix: Logging and beyond.**

http://www.sans.org/rr/whitepapers/logging/199.php

Cisco PIX Firewall System Log Messages, Version 6.3.

http://www.cicso.com/univercd/cc/td/doc/product/iaabu/pix_sw/v_63/63syslog/index.htm

Lachniet, Mark. **Inexpensive Cisco Network Log Analysis**.

http://lachniet.com/cheaplogging/

http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_sw/v_63/63syslog/pixemint.htm#wp1029160 Cisco Pix documentation message severity levels

http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_sw/v_63/63syslog/pixemapa.htm#wp1004080 Cisco Pix documentation PIX messages by severity

http://www.kiwisyslog.com/index.php Kiwi home page

http://tud.at/programm/fwanalog/ fwanalog home page

http://www.analog.cx/ Analog home page

http://www.sawmill.net/ Sawmill home page

Last Updated: October 2nd, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Seattle 2013	Seattle, WAUS	Oct 07, 2013 - Oct 14, 2013	Live Event
SANS Baltimore 2013	Baltimore, MDUS	Oct 14, 2013 - Oct 19, 2013	Live Event
SEC760 Advanced Exploit Development for Penetration Testers	Baltimore, MDUS	Oct 14, 2013 - Oct 19, 2013	Live Event
SANS Bangalore 2013	Bangalore, IN	Oct 14, 2013 - Oct 26, 2013	Live Event
GridSecCon 2013	Jacksonville, FLUS	Oct 15, 2013 - Oct 17, 2013	Live Event
Healthcare Cyber Security Summit	San Francisco, CAUS	Oct 17, 2013 - Oct 24, 2013	Live Event
Securing the Internet of Things Summit	San Francisco, CAUS	Oct 17, 2013 - Oct 22, 2013	Live Event
SANS Tokyo Autumn 2013	Tokyo, JP	Oct 21, 2013 - Oct 26, 2013	Live Event
ICS410	Sterling, VAUS	Oct 21, 2013 - Oct 25, 2013	Live Event
October Singapore 2013	Singapore, SG	Oct 21, 2013 - Nov 02, 2013	Live Event
SANS Dubai 2013	Dubai, AE	Oct 26, 2013 - Nov 07, 2013	Live Event
FOR572 Advanced Network Forensics and Analysis	Washington, DCUS	Oct 28, 2013 - Nov 02, 2013	Live Event
SANS Chicago 2013	Chicago, ILUS	Oct 28, 2013 - Nov 02, 2013	Live Event
MGT415 at (ISC)2 SecureSoCal 2013	Manhattan Beach, CAUS	Oct 31, 2013 - Oct 31, 2013	Live Event
SANS South Florida 2013	Fort Lauderdale, FLUS	Nov 04, 2013 - Nov 09, 2013	Live Event
SANS DHS Continuous Diagnostics & Mitigation Award (CDM) Workshop	Washington, DCUS	Nov 06, 2013 - Nov 06, 2013	Live Event
MGT415 at (ISC)2 SecureDallas 2013	Dallas, TXUS	Nov 06, 2013 - Nov 06, 2013	Live Event
SANS Pen Test Hackfest Training Event and Summit	Washington, DCUS	Nov 07, 2013 - Nov 14, 2013	Live Event
SANS Sydney 2013	Sydney, AU	Nov 11, 2013 - Nov 23, 2013	Live Event
SANS Korea 2013	Seoul, KR	Nov 11, 2013 - Nov 23, 2013	Live Event
Cloud Security @ CLOUD Expo Asia	Singapore, SG	Nov 13, 2013 - Nov 15, 2013	Live Event
SANS London 2013	London, GB	Nov 16, 2013 - Nov 25, 2013	Live Event
SANS San Diego 2013	San Diego, CAUS	Nov 18, 2013 - Nov 23, 2013	Live Event
FOR585 Adv Mobile Device Forensics	Vienna, VAUS	Nov 18, 2013 - Nov 23, 2013	Live Event
Asia Pacific ICS Security Summit & Training	Singapore, SG	Dec 02, 2013 - Dec 08, 2013	Live Event
SANS San Antonio 2013	San Antonio, TXUS	Dec 03, 2013 - Dec 08, 2013	Live Event
SANS Cyber Defense Initiative 2013	Washington, DCUS	Dec 12, 2013 - Dec 19, 2013	Live Event
SANS Oman 2013	Muscat, OM	Dec 14, 2013 - Dec 19, 2013	Live Event
SANS Golden Gate 2013	San Francisco, CAUS	Dec 16, 2013 - Dec 21, 2013	Live Event
SANS Forensics Prague 2013	OnlineCZ	Oct 06, 2013 - Oct 13, 2013	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced