

A Firewall Log Analysis Primer

As the primary perimeter defense for most networks, firewalls can often be an import intrusion detection and forensic tool. So, for those serious about information security, understanding firewall logs is extremely valuable. This article is a primer on log analysis for a few of today's most popular firewalls – Check Point Firewall 1, Cisco PIX, and NetScreen.

Why Analyze Firewall Logs?

For those with the resources to justify a 24x7 staff of security professionals and associated infrastructure or an outsourced team of pros, logs can be analyzed in real-time. For others, they may be processed in batch. Either way, your firewalls may have more to tell you security-wise than any other type of system. There are countless illustrations of this.

Before virus engine signatures were released for both Code Red and Nimda, firewalls were telling the story of these new worms. Firewalls were backed up with connections from newly infected hosts. Conscientious security administrators listened to their firewalls and, investigating these hosts, were among first to identify the malicious code. The same goes for the OPASERV worm. Firewall logs were filling with alerts of denied connections, or in some cases simply too many allowed connections. Of course, investigating some of the top talkers in the firewall logs revealed the problem. What we are talking about is early warning about major outbreaks. Further, most would be surprised how, in my work (enterprise security monitoring), we routinely find Trojan horses and root kits trying to phone home through firewalls (usually via IRC). These Trojans are more and more common with the increasing numbers of mobile users and more porous networks in general. They are almost never identified by any type of network or host-based IDS because they appear to be normal traffic and are so numerous and changing. Lastly, regarding forensics, most firewalls are single points of entry or nearly so. Presumably, any compromise or attack that comes from the outside world should leave some kind of fingerprint on the firewall logs.

Check Point Firewall/VPN 1

The first issue Check Point self-starters notice is that the logs are not in a human-readable format. They are viewable only through the Check Point GUI, by issuing the 'fw log' command, or via an API called OPSEC LEA. You will usually find the logs in /log under the product installation directory - \$FWDIR, if this environment variable is set. Another thing to note is that the logs are on the management console, which may or may not be the actual firewall (or enforcement point, as Check Point refers to it).

Traffic Logs

The most useful log entries for intrusion detection are the “accepts” and “denies” found in the main log. These entries are especially useful for seeing port scans, host sweeps, and general probing. Check Point gives you deny or drop alerts when traffic is not allowed and accept alerts when it is. This action is configurable. Drop means to drop the packet (read bit bucket), whereas deny means to send a TCP reset or ICMP port/protocol unreachable message. These alerts also contain the rule that applied, which is very useful for troubleshooting.

The format is typically as follows, however there are slight variations from version to version:

Time | Action | Firewall | Interface | Product | Source | Source Port | Destination | Service | Protocol | Translation | Rule

Fields summary

Time	Local time on the management station
Action	accept, deny, or drop. accept=accept or pass the packet. deny=send TCP reset or ICMP port unreachable message. drop=drop packet with no error to sender
Firewall	IP address or hostname of the enforcement point
Interface	Firewall interface on which the packet was seen
Product	Firewall software running on the system that generated the message
Source	Source IP address of packet sender
Destination	Destination IP address of packet
Service	Destination port or service of packet
Protocol	Usually layer 4 protocol of packet – TCP, UDP, etc.
Translation	If address translation is taking place, this field shows the new source or destination address. This only shows if NAT is occurring.
Rule	Rule number from the GUI rule base that caught this packet, and caused the log entry. This should be the last field, regardless of presence or absence of other fields except for resource messages.

Example 1 - Slammer:

This is a log entry triggered by the Slammer Worm hitting the outside of a perimeter firewall. You should see lots of these on most Internet-connected firewalls, as the number of these alerts went from nearly zero on average to hundreds of thousands per day on January 25th, 2003.

```
14:53:16 drop gw.foobar.com >eth0 product VPN-1 &
Firewall-1 src xxx.xxx.146.12 s_port 2523 dst
xxx.xxx.10.2 service ms-sql-m proto udp rule 49
```

Example 2 – Permitted Web Traffic:

This is a log entry for permitted HTTP traffic sourced from inside (eth1) with NAT. If, for instance, you had a CodeRed or Nimda infestation on an internal network, you would see the rate of these accepts increase dramatically. This is an illustration of how a normally benign log entry can signify something bad, in the proper context. This is also a reason you can't rely solely on automated processes for security alerting.

```
14:55:20 accept gw.foobar.com >eth1 product VPN-1 &
Firewall-1 src 10.5.5.1 s_port 4523 dst xxx.xxx.10.2
service http proto tcp xlatesrc xxx.xxx.146.12 rule 15
```

Example 3 – Nimda with Security Server Logging:

This is a log entry that illustrates how using Check Point's security server (read proxy) for HTTP traffic allows for more in-depth IDS and forensic analysis. Check Point security servers, also referred to as resources, are much like application proxies, and generally log more application-specific information. For HTTP, this includes the requested URL. Notice the difference below as compared to Example 2. Also notice the classic Nimda URL pattern.

```
14:55:20 accept gw.foobar.com >eth1 product VPN-1 &
Firewall-1 src 10.5.5.1 s_port 4523 dst xxx.xxx.10.2
service http proto tcp xlatesrc xxx.xxx.146.12 rule 15
resource=http://xxx.xxx.10.2/scripts/..%35c../winnt/sys
tem32/cmd.exe?/c+dir
```

Audit Logs

There are a couple of logs that are exceptions to the Check Point-specific logging format. Probably the most important is the cpmi_audit.txt (cpmgmt.aud in pre-NG versions). This log tracks all changes made via the GUI. Each entry shows the user who logged in, the machine they came from, the component they used (log viewer, policy editor, etc), the authentication method, and the change made. These are very useful for general auditing and for forensics regarding a compromised firewall host, especially since in distributed environments the logs are not actually on the firewall itself.

Example 4 – Change an Object:

Changing an object's IP address results in the following type of log entry. You will get similar entries for all object additions, deletions, or modifications.

```
OperationTime=Thu Dec 13 15:00:48 2002,  
ObjectName=Sanitized-Router, ObjectType=host_plain,  
ObjectTable=network_objects,Operation=Update,  
Administrator=fwadmin, Machine=cp-mgmt-station,  
ClientType=Policy Editor SessionId=Modification Info:  
ipaddr: changed from '10.10.5.3' to '10.10.5.7' ;
```

the following type of log entry.

```
OperationTime=Thu Jun 13 13:29:05 2002,  
ObjectName=Standard, ObjectType=firewall_policy,  
ObjectTable=fw_policies,Operation=Update,  
Administrator=fwadmin, Machine=cp-mgmt-station,  
ClientType=Policy Editor SessionId=Modification Info:  
rule 1 - track: added 'Log' ;rule 1 - track: removed  
'None' ;rule 3 - track: added 'Log' ;rule 3 - track:  
removed 'None' ;rule 4 - track: added 'Log' ;rule 4 -
```

You can see above that the actions were essentially to change the logging for rules 1,3, and 4 from “none” to “log.” Each action is preceded by the word ‘Track:’.

Example 6 – Log In/Log Out:

Logging in and out of the GUI, and of the Log viewer, look like the following.

```
OperationTime=Thu Jun 13 09:09:00 2002, Operation=Logged  
in, Administrator=fwadmin, Machine=cp-mgmt-station,  
ClientType=Policy Editor, Info=connected with user  
password
```

```
OperationTime=Thu Jun 13 09:09:11 2002, Operation=Logged  
in, Administrator=fwadmin, Machine=cp-mgmt-station,
```

Check Point-Specific Logging Issues and Challenges

As stated earlier, the normal logs are not clear text, and the GUI log viewer is not especially useful for real-time remote log analysis in that you can look at Check Point devices only. It is not useful for batch analysis either, as you cannot manipulate the information in familiar ways, as you would with text files or logs stored in a database. You are limited to the filters and sorting programmed into the client.

There are a few ways to get Check Point logs into familiar formats and transmit them to an analyst's workstation or into some central log aggregation facility.

OPSEC LEA:

Programming to this API is beyond the scope of this article, but is desirable for real time analysis because it enables authentication and encryption of Check Point log data traversing a network. Interestingly, instead of pushing logs from the host to a remote syslog server, LEA is a pull mechanism by which a client retrieves the logs from the Check Point management station.

The 'fw log' command and syslog:

As mentioned earlier, you may view Check Point logs from the command line with 'fw log'. More specifically, to view logs in real-time you issue the command 'fw log -ftn'. Many analysts use this command in conjunction with other tools to more securely send these logs in real-time over the network. Most involve piping this command to the UNIX logger utility so that the Check Point logs are transferred into UNIX syslog. For instance:

```
fw log -ftn | logger &
```

This command is often included in a script that also stops this process, rolls the logs, and runs this command again. This in itself is not as secure as necessary in many cases, as syslog will transmit this sensitive log data as clear text. There are many solutions to this. A VPN between the syslog server and the Check Point management station is probably the most common and is very secure. I have also seen Stunnel, SSH, and SSL toolkits used. Regardless of the approach, this data should be treated as sensitive and proprietary since configuration details about perimeter security, partial authentication credentials, and other private information are contained within it.

Of course without proper logging considerations in the firewall configuration, the logs will be of limited value. It is important to configure for logging every rule that may give good information. Generally, in fact, it is best to have logging on in every rule except for one that drops common trash traffic - Netbios for instance - directed to the firewall, as this is usually excessively noisy and provides limited visibility. Also, it is very important to have a stealth rule and a cleanup rule. The stealth rule blocks all but authorized management traffic directed at the firewall, and the cleanup rule (normally in the last position of the rule set) drops all traffic not expressly permitted in the above rules. These rules should always have

logging turned on. In pre-NG versions of Check Point the logging can be set to short or long. When disk-space permits, go long.

PIX

Cisco PIX has some of the most exhaustively documented logs in the firewall arena. The public Cisco site has good explanations of the different log categories, where applicable, and of individual log messages. See:

http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_62/syslog/pixmsgs.htm

The individual messages are each tagged with "message codes." These message codes are loosely organized into categories. For instance, message codes 400000 through 400051 are for Cisco Secure Intrusion Detection System signature messages. Not all messages are grouped so well, but you will generally find clusters of similar message types.

PIX logs are viewable in two ways - the PIX Device Manager (PDM) and UNIX style syslog. Most PIX administrators probably use syslog, so we'll use it for our examples here. The format is generally as follows, but the message portion varies widely.

Date | Time | IP/Hostname | Message Code | Message

Access List and URL Messages

The out-of-the-box configuration for PIX firewalls is built around security zones. The zone model essentially allows all traffic from higher security zones to lower security zones and not the other way around. This is very easy to set up, but is not the most secure approach. Among the many dangers is the possibility of traffic from backdoor programs and other malicious code to pass outbound unchecked. I encourage the use of classic access lists for both inbound and outbound connections. These, of course, should be configured for logging. That is, even though there is an implied 'deny any any' rule at the end of every access list, you should add an equivalent explicit rule, but ending with the 'log' tag. All explicit drops should have the 'log' tag, as well, except for trash traffic directed at the firewall - Netbios for instance. Verbose logging and strict traffic control are the keys to providing good logs for IDS and forensics. Most denied connections have a message code in the 106001 to 106023 range. The access list violations you would normally see are message code 106023, and the format is:

```
%PIX-4-106023: Deny protocol src [inbound-  
interface]:[src_address/src_port] dst outbound-  
interface:dst_address/dst_port [type {type}], code {code}]  
by access_group access-list-name
```

Example 7:

Just like in the Check Point Example 1, here we see denied Slammer traffic, this time hitting the access list "outside_acl," which is applied to the outside interface.

```
Feb  4 23:57:54 gw.foobar.com %PIX-4-106023: Deny udp src
outside:xxx.xxx.146.12/2523 dst inside:xxx.xxx.10.2/1434
by access-group "outside_acl"
```

Example 8:

Just as with the Check Point Example 2, permitted traffic can often be important information. Permitted HTTP traffic logs the requested URL, which can be useful in detecting in-protocol attacks like the Nimda alert seen below. Different URL-related messages are found in the message codes between 304001 and 304009 - the most common being 304001.

```
Feb  5 07:38:50 10.87.62.40 %PIX-5-304001: 10.5.5.1
Accessed URL
xxx.xxx.10.2:/aharrison@awod.com?on_url=http://xxx.xxx.10
.2/scripts/..%35c../winnt/system32/cmd.exe?/c+
```

Configuration Changes and Login Messages

Just as with Check Point's audit logging, PIX has many messages regarding configuration changes. Most are found between message codes 111001 and 112001.

Example 9:

The following message indicates that the current configuration is being written to memory. The console could be an IP address if the user was logged in remotely, and memory could be terminal, flash, standby, or floppy. This is bad, of course, if no authorized individuals are logged in the firewall.

```
%PIX-5-111001: Begin configuration: console writing to
memory
```

Normally you will see the user logging in before this type of activity, so it is useful to look for authentication messages, which are between message codes 611101 and 611103.

IDS Messages

PIX comes with some basic attack detection functionality inherited from its cousin Cisco Secure IDS (formerly NetRanger). PIX message codes 400000 through 400051 are Intrusion Detection System signature messages. There are two sets of IDS signatures – attack and info. To enable these you have to add them to the configuration with the 'ip audit' commands. Globally this would be:

```
ip audit info action alarm
ip audit attack action alarm
```

Per interface it would be something like:

```
; Create audit list for each set and
; call them "attack-ids" and "info-ids" respectively
ip audit name attack-ids attack action alarm
ip audit name info-ids info action alarm
; apply to outside interface
ip audit interface outside attack-ids
ip audit interface outside info-ids
```

Example 10:

This IDS log entry alerts that a TCP packet with incorrect flag settings. Of course illegal flag settings can be used for IDS evasion and OS detection, among other attack modes.

```
Dec 19 04:40:54 gw.foobar.com %PIX-4-400027: IDS:3041
TCP SYN+FIN flags from xxx.xxx.146.23 to xxx.xxx.10.2 on
interface outside
```

Check out the link at the beginning of this section for the full listing of PIX IDS signature.

NetScreen

Like PIX, NetScreen logs are very well documented. They have a PDF available at the following URL, which explains the message format, types, and conventions used.

<http://www.netscreen.com/support/downloads/Msg.pdf>

ScreenOS has message types similar in function to the PIX message codes - essentially categories. They also have severities, which are equivalent to standard syslog severities.

8 Standard Severity Levels

0 Emergency	System is unusable
1 Alert	Action must be taken immediately
2 Critical	Critical conditions
3 Error	Error conditions
4 Warning	Warning conditions
5 notification	Normal but significant conditions
6 Informational	Informational conditions
7 Debugging	Debugging-level messages

The basic format of ScreenOS messages is as follows, however traffic logs vary from this quite a bit – more about them later. Also, the fields are well labeled, so they should be mostly self-explanatory.

Date | Time | Module | Severity | Type | Message Text

You can view the logs through the NetScreen Global Pro manager or the web-based administration interface. You can configure certain messages to send SNMP traps, as well. The most common way mechanism is to configure the NetScreen to log to a remote syslog server. This is very straightforward to do from the GUI. You can individually configure each severity to syslog or not. You may only configure only one syslog server, but you can configure two predefined groups of messages to go to different syslog facilities. The two groups are “security” and “not security” and are not configurable. Although it is documented, it is not obvious what events are security and which are not, so you may find it easier to send all messages to the same syslog facility.

Traffic Logging

As mentioned earlier, the most useful log entries for intrusion detection are the “accepts” and “permits.” These entries are especially useful for seeing port scans, host sweeps, and general probing. In the NetScreen, these types of messages generally show up with a severity of notification, and with a type of 0025x.

Example 11:

The following is a NetScreen log entry for denied Slammer worm traffic. Note the type is 0025. The 7 means this as a subset of 0025. So, this is not just a type 00257 message, it is a type 0025. This is an important distinction when looking up type codes, as most will be listed specifically but not all.

```
Feb 5 19:39:42 10.1.1.1 ns25: Netscreen
device_id=00351653456 system-notification-00257(traffic):
start_time="2003-02-05 19:39:04" duration=0
policy_id=320001 service=1434 proto=17 src zone=Untrust
dst zone=Trust action=Deny sent=0 rcvd=40
```

Example 12:

The following is a NetScreen log entry for permitted HTTP traffic. Note that this message also contains the NAT address for this inbound service.

```
Feb 5 19:39:42 10.1.1.1 ns25: Netscreen
device_id=00351653456 system-notification-00257(traffic):
start_time="2003-02-05 19:34:44" duration=1 policy_id=0
service=http proto=6 src zone=Trust dst zone=Untrust
action=Permit sent=11903 rcvd=31454 src=10.5.5.1
dst=xxx.xxx.10.2 src_port=1254 dst_port=80 translated
```

NetScreen doesn't log URLs, so the Nimda example we've used for the other firewalls doesn't apply here. NetScreen integrates with WebSense (www.websense.com) for web content control and URL logging, so we'd have to monitor that product for in-protocol HTTP attacks.

Administrative and Change Logs

Most of the configuration changes and administrative user logging is tagged as warning or informational severity messages.

Example 13:

Logging in or out via the web-based administrative interface, telnet, or SSH (SCS in NetScreen terminology) are all warning severity and are type 00515. The format is as follows. The message text changes to signify SSH, telnet, or web.

```
Feb 7 14:37:30 10.1.1.1 ns25: NetScreen
device_id=00351653456 system-warning-00515: duration=0
start_time="2003-02-07 14:37:04" netscreen: Admin User
"netscreen" logged in for Web(https) management (port
443) from 12.146.232.2:3473. (2003-02-07 14:34:32)
```

Example 14:

Configuration changes are informational severity and type 00767. They generally look like this, though the messages text changes to represent the action.

```
Feb  7 14:41:33 10.1.1.1 ns25: NetScreen
device_id=00351653456 system-information-00767:
duration=1 start_time="2003-02-07 14:40:04" netscreen:
The system configuration was saved by admin "netscreen".
(0000 00 07 14 00 00)
```

Syslog Configuration

The examples above all used UNIX syslog. It is important to have the syslog server configured properly for logging to work. Though advanced syslog configurations and syslog alternatives are outside the scope of this article, I want to outline basic syslog configuration steps, as this is so easy, yet important. The steps are basic:

- Choose a syslog facility.
- Configure the firewall to send to your syslog server on that facility.
- Configure your syslog server to accept these messages
- Configure your syslog server to send these messages to the correct file.
- Remember to roll this file periodically.

Let's use NetScreen as an example. I have chosen local4 for my facility (no special reason). I mentioned earlier that the NetScreen was simple to configure from the web-based administration tool, so consider that done. Then we configure the syslog server. The following would be a likely syslog.conf.

```
*.debug;local4.none          /var/log/messages
local4.*                     /var/log/netscreen.log
```

This configuration sends debug level messages (all severities, that is) of all facilities *except local4* to /var/log/messages. It sends all severities of local4 to /var/log/netscreen.log. Next, if we are running Linux, we add /var/log/netscreen.log to the list of files in /etc/logrotate.d/syslog that should be periodically rotated. Note, there are many other ways to accomplish this, and different operating systems have their preferred ways. Finally, in most cases we'll have to change an rc file or init configuration (again, depending on OS) to start syslogd so that it will listen to remote hosts. This is usually a -r argument, something like this is common:

```
syslogd -r -m 0
```

The PIX is equally as easy. The syslog configuration is essentially the same, except for file names. To configure the PIX itself for logging issue the following commands from configuration mode:

```
logging on
logging trap debugging
logging host 10.1.1.1
logging facility 20
```

The one tricky item is the facility. For reasons directly tied to Cisco's penchant for bit masking, PIX logging facility 20 equals syslog facility 4. Their equivalents are as follows.

PIX Facility to Syslog Facility Mappings

16	00010000	Local0
17	00010001	Local1
18	00010010	Local2
19	00010011	Local3
20	00010100	Local4
21	00010101	Local5
22	00010110	Local6
23	00010111	Local7

We mentioned different ways to get Check Point logs, one being syslog via the logger command. When using this method, you may specify the facility and severity with the `-p` option. For instance:

```
fw log -ftn | logger -p local4.info
```

The syslogd configuration is essentially the same as the other firewalls covered.

Conclusion

We've covered the basics of firewall logging, and the good news is that most firewalls not covered here are just variations on a theme. Of course, this is a deceptively simple subject. There are projects underway to create new, more reliable, more standardized logging mechanisms; and there are currently a variety of syslog alternatives that boast these types of features. The issues surrounding log management and interpretation have vast tentacles - log normalization, log interpretation, log aggregation, secure transport of logs, real-time analysis, legal viability of logs, etc. For more information check out the following resources:

[Syslog Replacements](#)
Syslog-NG

<http://www.balabit.hu/en/products/syslog-ng>

Socklog

<http://smarden.org/socklog/>

Nsyslogd

<http://coombs.anu.edu.au/~avalon/nsyslog.html>

Mailing Lists

LogAnalysis - Moderated mailing list

<http://lists.shmoo.com/mailman/listinfo/loganalysis>

Archives of the LogAnalysis mailing list

<http://lists.shmoo.com/pipermail/loganalysis/>

Archive of the syslog security mailing list

<http://www.mail-archive.com/syslog-sec@employees.org/>

Tools

SHARP

<http://www.csis.gvsu.edu/sharp/>

Swatch

<ftp://ftp.stanford.edu/general/security-tools/swatch>

Logsurfer

<http://www.cert.dfn.de/eng/logsurf/>

Standards/Formats

IETF syslog Working Group Home Page

<http://www.employees.org/~lonvick/index.shtml>

CIDF (Common Intrusion Detection Framework) web site

<http://www.isi.edu/~brian/cidf/>

Legal

Computer Records and the Federal Rules of Evidence

http://www.usdoj.gov/criminal/cybercrime/usamarch2001_4.htm