Global Information Assurance Certification Paper

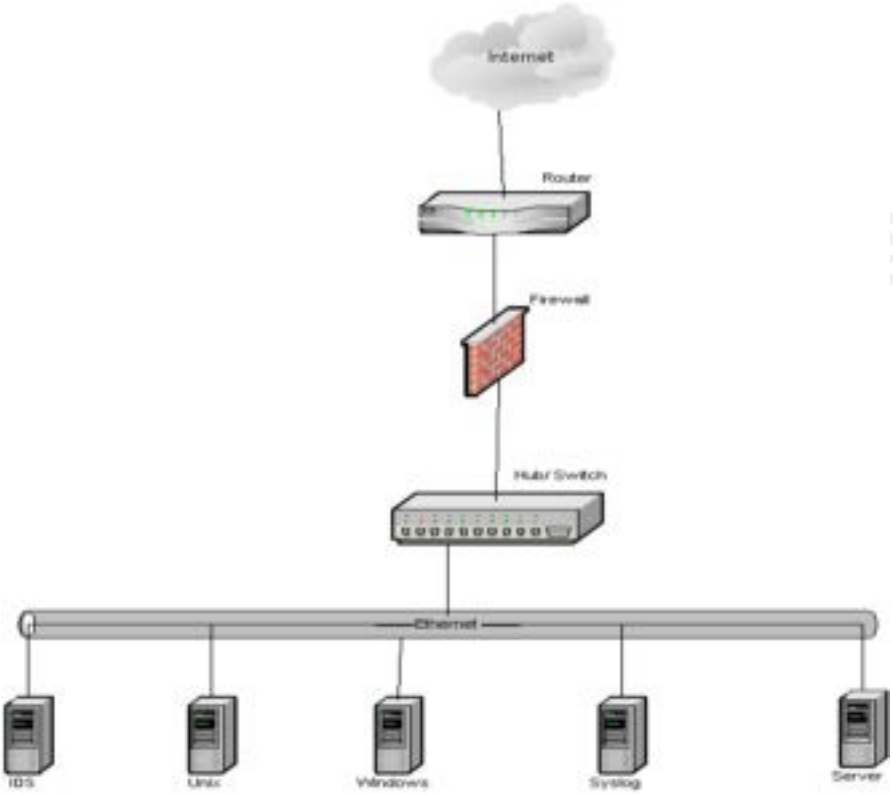
Shawn J. Mastria GSEC V1.3

Managing Log Files

Logging host and network activity is one of the most important security functions. Often managing log files on a large network can be a time consuming task. Proper logging and alerting techniques can greatly reduce the amount of time it takes to manage this data. Within this document you will learn effective logging and alerting techniques that will ease the management process of log data.

Security breaches are inevitable no matter how secure your network may be. Often the only evidence you are going to have after a compromise, or attempted compromise, is your logs. Logging the traffic entering and exiting your network is a good start, but this is not sufficient. Logging should be performed on every host, not only on the firewall and NIDS (Network Intrusion Detection System), which unfortunately seems to be a common mistake. "According to the Computer Security Institute/FBI and Ernst & Young, nearly 50% of all network attacks come from the inside. Often, from unhappy workers." [1]

Proper logging can often help identify malicious activity. For example, many companies allow users full control (local machine administrator) of their workstations. With full control a user can run l0phtcrack (or the password cracker of choice) to crack the .sam file and have the password of the administrator account on the local machine. In some instances this is the same as the administrators network login. Whether or not that is the best method of compromising the network, you want to know when this type of activity is occurring. Unfortunately you wouldn't without proper logging. To help improve effective logging, I am going to explain how to configure some common devices found on corporate networks to forward log data to a central syslog aggregator. Furthermore, within this document I am going to assume these "common" devices have already been installed and configured on your network and you are attempting to implement a more robust logging solution.



Syslog

Aggregator

"You can't go wrong if you decide to log all possible activities on your system. To be on the safe side, apart from logging it's a good thing to keep those logs secure and store them somewhere safe to minimize the possibility of someone tampering with them." [4] There are many ways to monitor and alert on log files. One way to implement a good logging procedure is to have all devices on the network report to a central syslog aggregator via syslog (UDP port 514). This implementation of logging provides for easier management and alert of potential unauthorized activity occurring on the network. The first step in effective logging via a syslog aggregator is configuring a host to collect syslog data from all devices on the network. Almost any flavor of Unix is capable of this, however FreeBSD is preferable. The syslog aggregator should be configured to receive syslog from specific IP's only. A hacker trying to cover his tracks could easily pollute the syslog data and potentially crash the syslog aggregator if the syslog aggregator accepts syslog data from any host.

To configure your syslog aggregator to accept syslog data from specified hosts you should edit the /etc/defaults/rc.conf and have the "syslog_flags" option configured with the "–a" flag. This flag allows you to specify the IP or IP's the syslog daemon will allow incoming syslog messages from., Tthis should appear as follows in the /etc/defaults/rc.conf (the IP should be the address of the syslog aggregator on your network): syslogd_flags="-a 10.0.0.1". After the syslog flags have been appropriately configured the next step is to install and configure a

program to monitor and alert on the log activity the syslog aggregator will be receiving.

SWATCH
http://www.stanford.edu/~atkins/swatch/

Often a program is used to monitor syslog data and report on events "near real time". One of these programs is called Swatch. Swatch can be installed from the ports collection within FreeBSD, and then run from the /usr/local/bin/ directory. The nice feature about Swatch is the ".swatchrc" configuration file you create in the /root directory. This configuration file allows you to specify patterns for swatch to look for and actions to perform (such as email) when each pattern is found.[2] If no configuration file is defined swatch will automatically echo everything in syslog, so be sure to configure "/root/.swatchrc", if the file doesn't exist then create it.

After you have created a .swatchrc configuration file, you can run Swatch using the following command: "/usr/local/bin/swatch –t /var/log/messages". The "–t" flag indicates the file for swatch to tail, in this case syslog is logged to /var/log/messages. To make sure Swatch is started every time the system is started a swatch.sh file should be created in the /usr/local/etc/rc.d directory. This file simply contains the "/usr/local/bin/swatch –t /var/log/messages" command and is executed during system boot. After configuring the various utilities on the syslog aggregator, ensure all services not in use are disabled. Appendix A shows a sample ".swatchrc" configuration file[3].

FreeBSD Host
http://www.freebsd.org

In addition to making good syslog aggregators, FreeBSD hosts often provide various other functions on corporate networks and have the ability to log effectively using the syslog daemon. FreeBSD is going to be referenced; however other flavors of Unix can be configured similarly. After the initial installation and configuration of a FreeBSD host verify syslog is functioning properly by typing "ps aux |grep syslogd". The host should return with an output similar to the following, displaying that syslog is running:

```
root       68  0.0  0.3   952  660  ??  Ss   Wed11AM   0:02.60 syslogd –s
```

When the above output is displayed ensure the "–s" is appended. This causes syslog to run in secure mode. [6] Secure mode does not allow remote network devices to log syslog data (or any other data) to your host. This is important because a large amount of data could be directed to you, fill up your disk space and possibly crash the system. If syslog is running without the

security flag it can be modified within the “/etc/defaults/rc.conf” file. You may find this option under “/etc/rc.conf” in older versions of FreeBSD, however now the “/etc/rc.conf” only contains overrides for the “/etc/defaults/rc.conf”.

The syslog daemon runs from “/etc/rc” upon system boot and flags are manipulated within the “/etc/defaults/rc.conf”. However what the syslog daemon logs and where it is stored is configured from the file “/etc/syslog.conf” file. If enough disk space is available consider configuring syslog to log everything locally as well as log everything to the syslog aggregator. To enable syslog to start logging all activity on the host simply uncomment the following entry within the sysolog.conf file, this can be done by simply removing the “#”

```
#*.*                                    /var/log/all.log
```

Some prefer to direct the log messages to event specific log files, such as sending failed logins to the /var/log/auth.log file. Send everything to the /var/log/all.log file since all messages are going to be forwarded to the syslog aggregator anyhow.

In addition to storing the logs locally we also want to forward them to the syslog aggregator. To have Unix hosts send syslog data to a log aggregator you need to make another change in your “/etc/syslog.conf” file. Within the syslog.conf file you should modify the entry for logging to a remote host. To begin sending the log data to the syslog aggregator uncomment the following entry by removing the “#” and replacing the word “loghost” with the IP or hostname of the syslog aggregator. The ending result of the file should be similar to the following:

```
# uncomment this to enable logging to a remote loghost named loghost
*.*                                     @10.0.0.1
```

Remember, every time a change is made in the syslog.conf file the syslog daemon (syslogd) has to be restarted. The syslog daemon can be restarted using the following command:

```
killall -HUP syslogd
```

Once a FreeBSD host is configured appropriately additional programs and services can be installed to further utilize the host. In addition to the system most software that is installed on the host can be configured to log to syslog as well. Intrusion Detection Systems are one type of software that is often installed on FreeBSD hosts and can be configured to log event data to syslog data as well.

SNORT

http://www.snort.org

Network Intrusion Detection Systems (NIDS) are often deployed in the corporate environment and when configured properly are very effective. There are a wide variety of NIDS, however I am going to inform you how to configure Snort logs to be redirected to the syslog aggregator. Once again, lets assume Snort has been installed on a FreeBSD host and the host is forwarding syslog to the syslog aggregator. In particular the objective is to configure Snort to log to syslog, then your host forwards these and your host logs to the syslog aggregator.

A standard installation of Snort logs to the "/var/log/snort" directory. Within this directory there is an alert, portscan and an IP specific file for every computer Snort has detected potentially malicious traffic from. Logging to the "/var/log/snort" directory can be effective, however someone would have to be monitoring these files to know when attacks are in progress. Of course Snort will be tuned according to your network activity. Without proper tuning of Snort you will likely be alerted with false positives more often then wanted.

Snort can be configured to log alerts to syslog, which can make managing logs and intrusion events more effective. Snort can be started with a large variety of options, however the "-s" flag will cause the alerts to be written to syslog ("/var/log/messages"). Even if the "-s" flag is specified port scans and IP specific events are still recorded within the "/var/log/snort" directory. "One thing to note about the last command line is that if Snort is going to be used in a long term way as an IDS, the "-v" switch should be left off the command line for the sake of speed."[8] One way to start Snort is as follows:

/usr/ports/security/snort/work/snort-1.8.3/snort -s –D

The above command starts Snort from the "/usr/ports/security/snort/work/snort-1.8.3" directory, logs alerts to syslog (/var/log/messages) and runs snort in daemon mode. Daemon mode "-D" causes snort to run as a process in the background. To ensure snort is started at system boot create a snort.sh file in the "/usr/local/etc/rc.d" directory. Within the snort.sh file simply add the "/usr/ports/security/snort/work/snort-1.8.3/snort -s –D" command and it will be executed during system boot.

Windows

www.microsoft.com

As every one knows, one of the most widely used operating systems is Microsoft's Windows Operating System. Within the corporate environment the most often deployed versions of Windows are Windows NT and Windows 2000. One major problem with many versions of windows is event logging. Both Windows NT and 2000 have logging options available, however doesn't seem to

be robust enough. Of course Microsoft isn't going to create an operating system that interacts easily with other operating systems for obvious reasons, so frequently "third party" software needs to be utilized.

Windows NT and 2000 logs can be viewed by using event viewer, which is located within the administrative tools. However, there is no easy way in either Windows NT or Windows 2000 to have the logs forwarded via syslog to a central location without third party software. There are a few vendors who make software to convert Windows event logs into syslog, one of these programs is called Event Reporter.

Event Reporter takes the standard Windows event logs and generates syslog messages that can then be forwarded to a syslog aggregator or stored on the local host. The Event Reporter application just takes the event logs and sends them to the syslog aggregator in standard syslog format, and if you have a large number of Windows hosts configured to log successful events this could cause a problem because of the amount of events that could occur. As a service running in the background on a Windows host, Event Reporter executes its function almost seamlessly. This piece of software runs as a service in the background and is very easy to configure.

Configure your windows auditing policy to best fit your needs. When configuring your audit policy remember that Windows stops logging when the event logs are full unless you choose "overwrite as needed" within the properties for each event log. [4] Therefore, if the logs were to fill up when "overwrite as needed" is selected you would still be able to record these events because Event Reporter would forward them off to the syslog aggregator before they are overwritten. For example, suppose you're auditing user logins. A success audit is a situation in which a user logs in successfully, and when you have several thousand employees in a large corporation the successful logins will generate a large portion of the log data alone. [7]

After initial installation open the Event Reporter client from programs within the start menu. Initially the only change that needs to be made is the destination IP of the syslog server within the client. Of course the software is more configurable, such as the interval you would like Event Reporter to check your log files for new entries. Many of the options are going to be configured based on the amount and type of activity occurring on each host. After you enter the IP of the syslog server simply press apply and your logs will start being forwarded. Once an effective logging baseline on the hosts has been established a continued effort should be made for all devices.

Cisco IOS

Logs from routing and switching devices on your network are very important because these devices have vulnerabilities just like most other devices on your network. There are a variety of routing and switching vendors, however the widely deployed Cisco products are going to be covered in this document. Cisco's IOS is the "operating system" that runs on their routers and switches. You can think of the IOS as an operating system just like Windows or Unix, except it is for Cisco devices and uses proprietary commands.

Having Cisco devices deployed on a network and configured correctly to send syslog data to a syslog aggregator is the main objective. Logging can vary depending upon your architecture, however in our case we want to standardize our logging as much as possible. Along with the other devices we also will forward the Cisco logs to the syslog aggregator.

Configuring the logging parameters on the Cisco device is the first step. Remember you must be in "enable" mode to change configurations on the device. When using telnet to connect to the device a password prompt is the first thing seen. After the password is entered you are taken into an "unprivileged" mode. In the "unprivileged" mode you are able to perform simple commands such as "ping", "telnet" and "traceroute". However you cannot change configurations on the device.

To make the necessary changes enable mode would have to be entered. Enable mode is a "privileged" interface to the router, which allows you to have full control of the device. The easiest way to know you are in enable mode is the "#". When you receive the "hostname#" then you can begin to make changes to the logging parameters. Once in enable mode type "configure terminal".

From this point you can begin to configure the logging parameters on the router. For starters direct the logging data to the syslog aggregator using the "logging" command with the syslog aggregator IP 10.0.0.1. The "logging" command simply specifies the destination for the log data to be sent to. The final step is ensuring logging is turned on, by issuing the "logging on" command. Appendix B shows the entire sequence to enter enable mode and configure the router.

With Cisco devices be sure to write the configuration to memory using the "write memory" command. Until the configuration is written into memory it is stored in a buffer and will be erased if the router is restarted. At this point the Cisco device will be configured to send syslog data to the log aggregator.

Cisco Pix

One of the most important devices to receive detailed log information from is a firewall. Often the firewall is the first line of defense when attacks originate

from outside the local network, therefore would also be the first device to alert you of suspicious traffic. Typically an attacker will perform some sort of reconnaissance before executing an actual compromise and frequently the firewall can give a preliminary idea of what's to come.

There is a wide variety of firewalls that perform their functions effectively. One of these devices is the PIX manufactured by Cisco. "Cisco PIX Firewalls are purpose-built firewall appliances that utilize a proprietary, hardened operating system which eliminates security risks associated with general purpose operating systems. PIX firewalls also provide the latest in security technology ranging from stateful inspection firewalling, IPsec and L2TP/PPTP-based VPNs, content filtering capabilities, and integrated intrusion detection to help secure your network environment from next-generation attacks."[5]

In addition to a large quantity of security features the PIX also includes a robust logging functionality. Events that are generally reported to the PIX console can be redirected to a syslog aggregator. By having the PIX send logging information to a syslog server it will make managing performance, possible malicious activity and trouble shooting problems easier.

There are a few versions of the PIX proprietary operating system, however I am going to cover 4.3.X and later. Not having specific events logged and the ability to have log messages time stamped are some of the added logging features within 4.3.X and later versions. In addition to the new functionality it also has the logging host, logging facility and logging trap commands. Within these commands you would specify the IP of the syslog aggregator, the facility and the level. "The *logging facility* can be thought of as *where* and the level can be thought of as *what".* Depending upon your network configuration the facility and level could change, however specifying level 7 is often the best choice because it logs all activity that occurs.

After a basic configuration is complete a few commands need to be entered to direct the logging data from the PIX to the syslog aggregator. First identify the IP of the syslog aggregator with the "logging host" command. If necessary, use the "logging facility" command to further identify where the log data originated. Finally the "logging trap" command is specified to have the PIX forward all messages to the syslog aggregator. Appendix C shows the sequence to configure the PIX to forward syslog data to the syslog aggregator.

As a last point syslog data is not encrypted and if not properly implemented could be "sniffed" off your network or the Internet. Data not encrypted could result in sensitive information being compromised. For that reason syslog data should never be sent from one device to another in an unsecured environment without first being encrypted (of course that would open up a whole new can of worms).

Appendix A

```
### Snort alerts from firewall
watchfor /IDS/
        echo bold
        mail addressess=admin,subject=--- Snort IDS Alert ---
        exec echo $0 >> /var/log/IDS-scans
        throttle 01:00 use=IDS27

watchfor /PORTSCAN DETECTED/
        echo bold
        mail addresses=admin,subject=--- Snort Port Scan Alert ---
        exec echo $0 >> /var/log/IDS-scans

### DNS zone transfers
watchfor /approved AXFR/
        echo bold
        mail addresses=admin,subject=--- Zone transfer Alert ---
        exec echo $0 >> /var/log/IDS-scans
### Bad login attempts
# watchfor   /failed/
#       echo bold
#       mail addressess=root,subject=Failed Authentication
### Some is sniffing!
# watchfor   /promiscuous/
#       echo bold
#       mail addressess=root,subject=Someone is sniffing the network!
### Ignore this stuff
# ignore   /sendmail/,/nntp/,/xntp|ntpd/,/faxspooler/
### Kernel problems or system reboots
# watchfor   /(panic|halt|SunOS Release)/
#       echo bold
#       mail addresses=root,subject=System Panic,Halt, or Reboot!

# watchfor   /file system full/
#       echo bold
#       mail addresses=root,subject=File system Full
#       throttle 01:00

# watchfor   /su:/
#       echo bold
#       mail addresses=root,subject=Someone sued to root access
```

Appendix B

password:”enter password here”

test-router>en

password:”enter password here”

test-router#

test-router#configure terminal

test-router (config)#

test-router (config)#logging 192.168.1.2

test-router (config)#logging on

test-router (config)#exit

test-router#write memory

Appendix C

```
logging host 10.0.0.1

logging facility local0

logging trap debugging
```

Works Cited

[1] Berst, Jesse "The Biggest Threat to Your Network's (It Isn't What You Think"
http://www.zdnet.com/anchordesk/story/story_1959.html

[2] Atkins, Todd "swatch manual pages"
http://www.stanford.edu/~atkins/swatch

[3] Spitzner, Lance "Swatch configuration file for Linux box"
http://www.guides.sk/lspitz/swatchrc.txt

[4] Stancin, Aleksandar "Introduction to logging"
http://www.net-security.org/text/articles/logging.shtml

[5] Cisco Systems, Inc "Cisco PIX Firewall Series"
http://www.cisco.com/univercd/cc/td/doc/pcat/fw.htm

[6] "System logging via syslog"
http://people.freebsd.org/~jkb/howto.html#log

[7] Posey, Brien "Creating an Audit Policy"
http://networking.earthweb.com/netos/article/0,,12282_624801,00.html

[8] Roesch, Martin "Snort Overview"
http://www.snort.org/docs/writing_rules/chap1.html#tth_sEc1.5