Global Information Assurance Certification Paper

UNIX Logging and Security

(Systems Under Siege)

This paper will discuss system the use of the UNIX logs to detect possible intrusions or intrusion attempts. I will discuss the UNIX syslog facility, freeware applications, and the use internally developed scripts for the detection of successful and attempted system intrusions. By default UNIX installations do not log all critical information. The information that is logged resides on the same system. Without proper monitoring of log files including automatic email notifications and the forwarding of messages to remote servers an administrator cannot effectively monitor his systems.

System Logging and Security

Effective logging and log processing is the key system security. System logs can report everything from hardware errors, bad su attempts, failed logins, rejected TCP-Wrappers connections and scans. Processing the log entries via scripts run out of cron on the loghost provides near immediate email notification upon any event of interest.

What to log

Different UNIX systems utilize log files in different locations and with different names. In this document I will discuss the log file structure on Solaris 7.

All successful and unsuccessful su attempts are logged to **/var/adm/sulog**. Keeping track of who is using the su command specifically who is attempting to su to root is critical for successful security monitoring.

To log failed login in attempts Solaris uses **/var/adm/loginlog**. This file must be created manually be owned by root and group sys, and must have the permissions of 600. Log entries will be created after five failed login attempts by default.

The last command monitors who is logged into your systems, and when, and from where. It's information is logged to the **/var/adm/wtmpx** file. This file is stored in binary format and the **last** command must be used to read the file.

System events including software and hardware. Events are logged into the **/var/adm/messages** file. These include hardware errors, Operating System errors, and security related messages. These messages could be generated from successful and failed logins, connections from TCP-Wrappers, su attempts, and from sshd.

Syslog Facility

The syslog facility consists of a program that runs as a daemon (**syslogd**) and it's corresponding configuration file **(/etc/syslog.conf**). Syslogd takes the messages it receives and forwards them to the proper location based on the instructions in the syslog.conf file. Syslog encodes messages by severity (emerg, alert, crit, err, info, debug, none) and facility or source (kern, auth, daemon, mail, lp, user). These messages are stored in ASCII files on the local system or forwarded to a remote system's syslog daemon or mailed.

Monitoring su root attempts:

A simple script can be written to monitor the **/var/log/remote/sulog** for attempted and successful **su root** commands. If the **sulog** is not zero in size then search for the su string. Compare the user attempting to su to a list of authorized administrators. If the user is not part of the list send an email. Move the **sulog** file to **sulog.0** (for example).

Remote Logging

Syslog by default logs information locally. It can be configured to forward the log events to a remote system running syslogd or to be emailed . In the event of a successful intrusion the hacker will attempt to remove any traces of himself by editing the local log files. If syslogd is configured to forward the log information to a remote host the hacker will not be able remove his log entries. It is critical to configure the system to send the log information remotely to prevent the hacker from covering his tracks.

When logging to a remote host syslog will use send the log information to the system defined as the "loghost" in the **/etc/hosts** file:

```
121.232.343.454     myhost          loghost
```

The **/etc/syslog.conf** on the **local host** will contain the typical entries below. These entries tell syslogd to send the log information to the **loghost** as well as the local **/var/adm/messages**. Note that all fields in the **/etc/syslog.conf** file must be separated by tabs.

```
auth.info       @loghost        # For logging authentication info
kern.notice     @loghost        # For logging kernel messages
uucp.info       @loghost        # For logging TCP Wrappers info
local6.info     @loghost        # For logging sshd log info
```

On the **loghost system** that receives the log information from the monitored systems the **/etc/syslog.conf** file should contain similar entries to below:

```
auto.info       /var/log/remote/authlog     # For logging authentication info
kern.notice     /var/log/remote/kernlog     # For logging kernel messages
uucp.info       /var/log/remote/wraplog     # Used for receiving TCP-Wrappers
local6.info     /var/log/remote/sshlog      # For logging sshd log info
```

Processing log information

There are different ways to monitor and process the log files. These depend on what you want to watch for and you level of programming or scripting. Much can be done with simple perl or shell scripts. There are also free tools available on the internet that can be used to process log files such as **Swatch** and **Logcheck.**

Swatch "The Simple WATCHer and filter" was written to actively monitor messages as they are written to a log file. **Swatch** monitors log files for specific triggers. When these triggers are matched the **Swatch** notifies you in the predefined manner. For more information on **Swatch** go to http://www.stanford.edu/~atkins/swatch/.

Logcheck is a free public domain program that is used to monitor UNIX log files. **Logcheck** monitors for keywords then notifies the administrator when a keyword match is found in a log file. **Logcheck** can work in a report everything mode. In this mode you must explicitly ignore messages. This is a good feature because it is impossible to know every possible type of message that can be logged. For more information on **Logcheck** to go: http://www.psionic.com/abacus/logcheck/ .

Watching for unauthorized connections and scans

Write a script that watches the **/var/log/remote/wraplog**. To get messages on all connections accepted and denied. Check the file, if the file is not zero in size email it's contents then send the kill –HUP `cat /etc/syslogd.pid` command. If you are only interested in denied connection to the same as above but search on the string denied. Email all lines that contain the word denied.

Additionally the **/var/log/remote/wraplog** can be processed and emailed only on apparent scans. Process the log file looking for connections to a minimum of three different hosts from the same IP address within two minutes. If this condition is matched send an email.

The secure shell daemon can be configured to log connection information via syslog. The **/etc/sshd_config** file must contain the entry **SyslogFacility local6**. This entry will cause sshd to log to the local6 source. Processing of the **/var/log/remote/sshlog** would be done the same as for the wraplog.

Conclusion

Closely monitoring all of your UNIX logs via automated tools is essential to effective security. By default UNIX does not log all activity. The default logging that UNIX does is stored on the same system where it can be easily modified by an intruder.

The UNIX syslog facility can be configured to forward it's information to a log host. This is the first and most important step. There are free tools on the internet that can be used to automate the monitoring of log files. Some log file on UNIX do not use the syslog facility. Utilizing a log monitoring tool like "SWATCH" or "Logcheck" that can automatically email upon matched conditions is an easy way to provide effective log file monitoring. Custom scripts and programs can also be easily developed to automate the monitoring of log files.

References

Gregory, Peter H. " Solaris Security" – Sun Microsystem Press 2000

Frisch, AEleen. "Essential System Administration" – O'Reilly and Associates, Inc. 1993

Dik, Casper "Solaris FAQ" - July 6 2000
http://www.science.uva.nl/pub/solaris/solaris2/

Akins, Todd "SWATCH, The Simple WATCHer" July 26,2000
http://www.stanford.edu/~atkins/swatch/

Rowloand, Craig "Psionic Software Logcheck" May 5, 2000
http://www.psionic.com/abacus/logcheck/

Last Updated: May 28th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event