Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Case Study: Implementing a Centralized Logging Facility

During the past several years I have found that there is an increase use in the number of Windows based systems appearing in our predominately all UNIX environment. This has been a downfall especially since UNIX and Windows systems are so different with regards to logging facilities, UNIX with its syslog facilities and Windows Eventlog; therefore I needed to find a way so that our Windows and UNIX systems could utilize a more robust logging facility. With budget concerns, being a major contributing factor, I needed to ...

Copyright SANS Institute
Author Retains Full Rights

AD

Case Study:
Implementing a Centralized Logging Facility

08/06/2003

SANS Security Essentials
GSEC Practical Assignment
Version 1.4b Option 2

Richard L. DuClos

© SANS Institute 2003, Author retains full rights

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Case Study: Implementing a Centralized Logging Facility

Abstract

During the past several years I have found that there is an increase use in the number of Windows based systems appearing in our predominately all UNIX environment. This has been a downfall especially since UNIX and Windows systems are so different with regards to logging facilities, UNIX with its syslog facilities and Windows Eventlog; therefore I needed to find a way so that our Windows and UNIX systems could utilize a more robust logging facility.

With budget concerns, being a major contributing factor, I needed to find a solution that was inexpensive. Therefore all the items that I chose to implement at this time are freeware and applications that already exist in our environment. The Windows systems needed to be configured so that they would audit the proper events and then forward that onto a UNIX system for storage and eventually analysis. Next, the UNIX systems needed a bit of tuning to get syslog to log the correct items. Finally, the logs needed to be retained and rotated. After these steps the logs can then under go further forensics and alerting of specific events.

This document will provide all the necessary information needed to configure a centralized logging facility for Windows and UNIX systems including configuration of Windows eventlog, auditing, and UNIX syslog.

Before

After reading the GSEC material especially the UNIX and Windows security sections, the importance of logging becomes more relevant. For this reason I have chosen to improve our current logging configuration. This is supported by a document on the CERT website about logging management. The author states that without a proper logging mechanism running hinders the ability to identify "suspicious behavior and intrusion attempts and to determine whether or not such attempts succeeded (Manage, 1)." Not only is the important item is the act of logging but also what to do with the log files. The CERT author also remarked that the logs should be in a remote, physically secure location but still accessible from the network.

The prior configuration consisted of several Windows 2000, AIX, and Solaris 8 servers in the same environment. These are all independent servers consisting of Web servers, database servers, and application servers all on the same network. Each one of these servers was configured to perform some type of logging, but there was no clear policy in setting them up. One thing that I have learned in the GSEC material is that having a good policy for logging is very important.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

When the Windows 2000 servers were originally installed they were not configured to do any extra auditing and the event logs remained at the default settings. Once the event logs fill up they are overwritten and all history of what is occurring on the server is lost (. At the time the Windows servers were installed security was not the primary focus, and it was rarely discussed. Since there is not a domain controller for these servers management is somewhat difficult and is hindered by not having a common configuration.

The environment not only has many separate Windows 2000 servers, but many AIX and Solaris UNIX servers. As with the Windows servers the UNIX servers performed some local logging, but not what is needed to effectively monitor the system security and performance. The Solaris systems were installed quickly and also before security became better understood. They were not configured to log any messages except for messages from the mail subsystem. Like the Solaris systems the AIX systems were configured prior to any sort of security methodology. I found the logs on the AIX servers were not configured to store the logs on a separate volume. Although, the AIX servers log much more messages from all available facilities which give much more granularity than the Solaris configuration. Since there was not a common configuration at the time these servers were configured, further warrants logging facility modification.

With having described the multiple platforms in the environment, a look at the log retention and rotation is needed. As I have previously mentioned about the Windows servers, they over write there logs and perform no log rotation. The default size of each of the event logs, System, Application, and Security, only store 512KB of data and are set to overwrite when the system needs to. The log would simplify start back at the beginning and erase existing information (Selecting, 4). The UNIX servers were not configured to rotate the logs so the messages would simply get written to the same file until the volume was completely filled.

Given the description of the configuration of the Windows and UNIX server including the ineffective logging methodology, much needed change is needed. For having read the GSEC material and realizing how important logging is evidence to showing that this current configure provides no “Defense in Depth”. To define this vulnerability, it is not having a good logging practice, which can possibly pose problems in securing the system by not knowing what is going on.

<u>During</u>

In approaching this problem I needed to address the issue of the existence of Windows 2000 and UNIX servers in the same environment. Firstly, I configured the Windows 2000 servers to begin to audit events and forward them onto the UNIX syslog facility. Secondly, I addressed the issue of having multiple UNIX servers, which too will need to be configured to send messages to the centralized

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

UNIX server. Thirdly, a log retention and rotation mechanism was developed for the servers.

Once I identified the systems and their inadequate logging configurations. I began to configure these systems to conform to a better logging methodology. To do this I planed a centralized logging server. The resulting topics in this section will discuss in detail what is needed to do on each respective server. These topics include configuring Windows 2000 event log and auditing, Solaris and AIX syslog configuration, and finally implementing log retention and rotation.

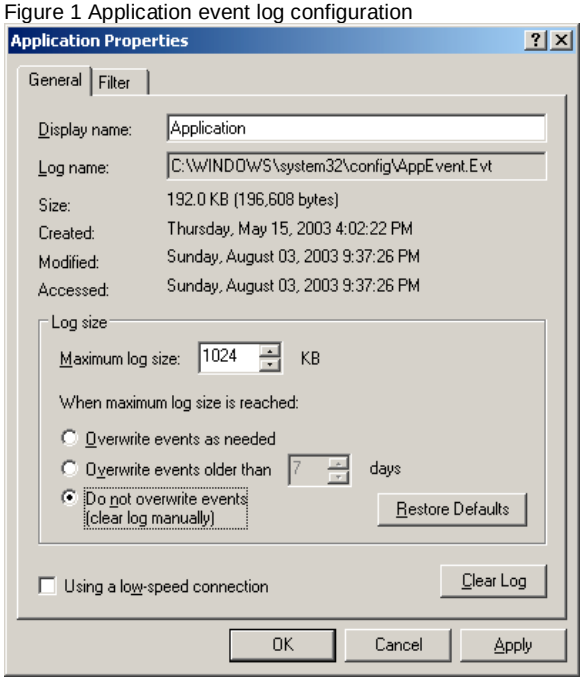
Windows 2000 Server Event Log and Audit Configuration

First, the Windows 2000 servers need to have the event logs and auditing configured. Also, since the UNIX servers in this environment out number the Windows servers, I decided that utilizing one of the UNIX servers for the centralized repository would be most efficient. Therefore, I needed an application that would allow Windows to send messages to the UNIX syslog. But, before utilizing such an application I needed to configure the Event log and auditing.

Although Windows provides a mechanism for logging events, application, and security items it does not do so by default. This proven by the material provided in the Windows security section of the GSEC material. By default Windows does not log much, especially regarding security events (Smith R, 1). The first step, was to configure the size and properties of the System, application, and security logs. I did this by following instructions from the CERT document “Selecting Windows NT 4.0 event log settings”. Although, this document was written with Windows NT in mind, it contained information regarding log file size and properties needed to effectively set the logs up. As depicted in Figure 1 the file size has been changed from 512KB to 1024KB and when it reaches this size it has been set to “Do not Overwrite Events (Clear Log Manually)”.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Figure 1 Application event log configuration



This procedure must be performed on each of the three logs (application, event, and security).

Once the event logs were properly configured auditing needed to be configured. During my research of the subject, I also found Windows 2000 audits two more categories than NT did. They are "Audit account logon events" and "Audit directory services access" (Smith R, 2). The following, Table 1 contains the nine audit categories and description of each:

Table 1 Windows 2000 Audit Categories

Category	Description
Audit account management	Tracks the creation, changes, and deletion of users and groups
Audit logon events	Records whenever a user attempts to logon
Audit object access	Tracks access to files, directories, registry keys, and printers
Audit policy changes	Track changes to the system's audit policy and rights
Audit privilege use	Tracks use of user rights
Audit process tracking	Monitor each program that executes and how log it ran on the system
Audit system events	Monitor when the system boots or clearing of event log occurs
Audit logon events	System traps authentication attempts on Domain controllers
Audit directory services	Similar to audit object access except it applies to Active Directory object instead

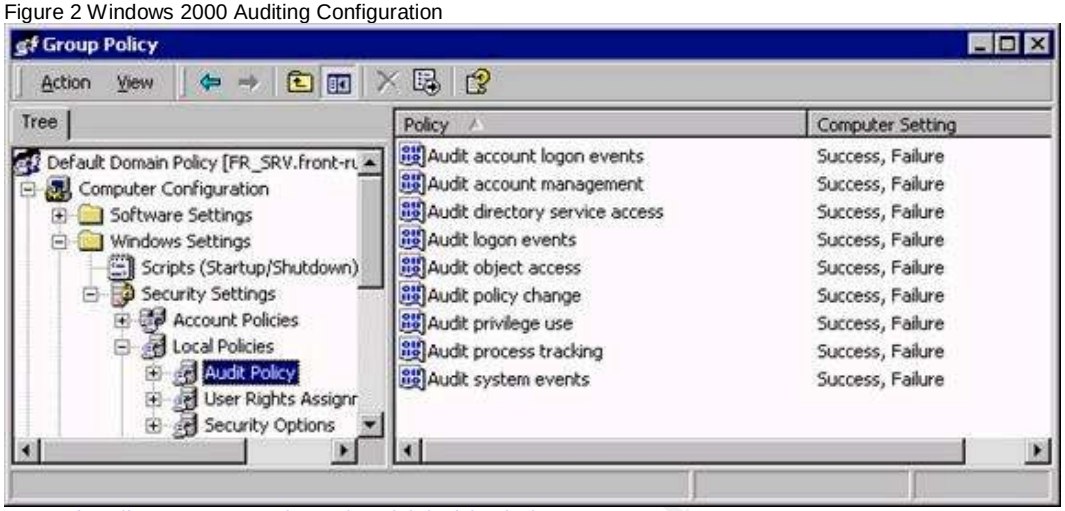
Source Smith, Randy Auditing Windows 2000

The following, Figure 2, is a display taken from a Windows 2000 server during configuration of auditing. After reading the article by Randy Smith, I decided to enable all of Windows auditing for both success and failed attempts, because of

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

the events that Windows can audit on (Smith R, 1). In the future if I seem to be getting too much meaningless messages I can simply modify the audit settings. But, with all these events enabled for Successful and failed attempts, Windows will still be able to better log events that are occurring.

Figure 2 Windows 2000 Auditing Configuration



Source http://www.serverwatch.com/tutorials/article.php/1474211

To configure auditing, I followed the set of steps from Microsoft's Knowledgebase.

Enabling Local Security Auditing

1. Log on to Windows 2000 with an account that has Administrator rights.
2. Click **Start**, point to **settings**, and then click **Control Panel**.
3. Double-click **Administrative Tools**.
4. Double-click **Local Security Policy** to start the Local Security Settings snap-in in Microsoft Management Console (MMC).
5. Double-click **Local Policies** to expand it, and then double-click **Audit Policy**.
6. In the right pane, double-click the policy you want to enable or disable.
7. Click the **Success** and/or **Fail** check box(es) as appropriate.

Source Microsoft Knowledgebase

With having finished the configuration of the event logs and auditing on the Windows 2000 servers the next step for the Windows servers was to configure them to send their messages to a UNIX syslog.

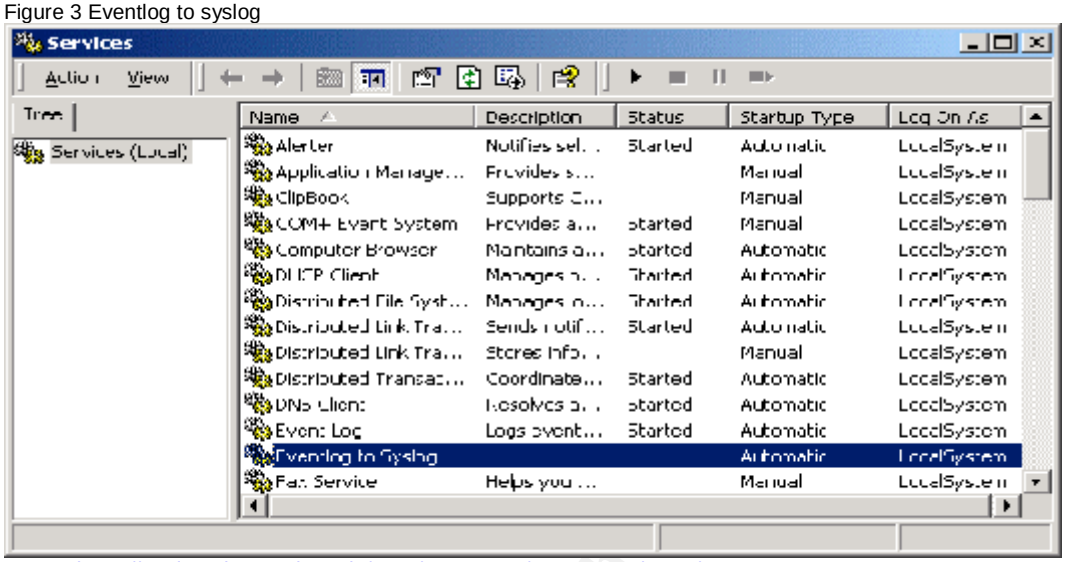
Windows Eventlog to Syslog configuration

Since I chose to configure a UNIX server for the centralized repository Windows 2000 needed to talk to syslog. After exhaustive searching I found an article by Urbana Der Ga'Had about a utility by Purdue's Engineering Computer Network that formats and send Windows eventlog messages to UNIX syslog facility. Curtis Smith described that the messages that appear in the eventlog after

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

installation will be forwarded to the defined server. He also stated that it depends on the message facility and priority as to what will be done with the message. After a careful evaluation of the software I decided to install it to determine if it gave the granularity needed. I followed instructions from Ga'had and Purdue to configure this utility (Figure 3).

Figure 3 Eventlog to syslog



Source https://engineering.purdue.edu/ECN/Resources/Documents/UNIX/evtsys

I followed the instructions from the document by Curtis Smith to install the evtsys application as a service (Smith C, 2). According to the documentation from Purdue this application logs to syslog with the facility of Daemon and priorities of err, warning, or notice. The application was copied into the system32 directory and executed by "evtsys –I –h logger", where the -h field refers to the centralized server name "logger" (Smith C, 3). I also made sure that the logger name was defined in the host file.

With having finished the Windows 2000 server configuration, the next step in the process was to configure the Solaris and AIX servers to communicate with the central log server.

UNIX syslog configuration

Since UNIX syslog will be the facility that will store the entire logs for both platforms some information must be given on UNIX syslog. For this implementation, AIX 4.3.3 will be running on the logging server and most of the UNIX servers in our environment. According to IBM's Redbook, the syslogd daemon "reads a datagram socket and sends each message line to a destination described by the /etc/syslog.conf file (Pruett, 62)." Since syslogd will read messages from a socket is one reason it can be used to receive messages from other systems. The syslogd running on Solaris is defined as " a collecting mechanism for various logging messages generated by the kernel and applications running on UNIX operating systems (Configuring, 1)."

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

UNIX client syslog

The UNIX servers that would be acting as clients, with regards to logging, they needed to be configured to send their respective messages to the centralized repository. The Solaris servers needed to have their configuration changed from logging only the mail subsystem to logging all available facilities. The AIX servers needed not have there facilities changed but only modified and having the logs stored on a separate volume. The following section will cover configuring Solaris clients, AIX clients, and the centralized logging server configuration.

First, the Solaris servers, clients, needed to have their syslog.conf files modified to correctly log the proper events the to the central log server. According to a CERT document on configuring syslogd on Solaris contained information on understanding the facilities and priorities of syslog. As in Table 2 Solaris syslog receives messages from the same facilities as AIX except, Solaris has a definition for local0.7 which are "Reserved for local service"(Configuring, 2). Solaris also has the same message priorities as AIX (Table 3).

Table 3 Syslog Facilities

Facility	Description
kern	Kernel
user	User level
mail	Mail subsystem
daemon	System daemons
auth	Security or authorization
syslog	Syslogd daemon
lpr	Line-printer subsystem
news	News subsystem
uucp	Uucp subsystem
*	All facilities

Source IBM RedBook

Located in Table 4 are the different priority levels that can be given to each of the previously defined facilities. The priority levels contained in Table 4 are sent as well as the level above the selected. For example, if alert was chosen then messages with alert and emerg would be sent to the syslog facility (Pruett, 66).

Table 4 Syslog Priority levels

Priority Level	Description
emerg	Specifies emergency messages (LOG_EMERG). These messages are not distributed to all users. LOG_EMERG priority messages can be logged into a separate file for reviewing.
alert	Specifies important messages (LOG_ALERT), such as a serious hardware error. These messages are distributed to all users.
crit	Specifies critical messages not classified as errors (LOG_CRIT), such as improper login attempts. LOG_CRIT and higher-priority messages are sent to the system console.
err	Specifies messages that represent error conditions

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

	(LOG_ERR), such as an unsuccessful disk write.
warning	Specifies messages for abnormal, but recoverable, conditions (LOG_WARNING).
notice	Specifies important informational messages (LOG_NOTICE). Messages without a priority designation are mapped into this priority. These are more important than informational messages, but not warnings.
info	Specifies informational messages (LOG_INFO). These messages can be discarded but are useful in analyzing the system.
debug	Specifies debugging messages (LOG_DEBUG). These messages may be discarded.
none	Excludes the selected facility. This priority level is useful only if preceded by an entry with an * (asterisk) in the same selector field.

Source IBM RedBook

The client machines, Solaris and AIX, were configured to log all messages to the central logging server and also locally. This was decided since the Windows servers will still retain their copies of eventlog and for redundancy. In a document by Colin Bitterfield, he stated in his configuration that the client machines will restrict remote messages. This is done to keep clients from accepting messages from remote sources (Bitterfield, 3). So, I took this into account when I configured the client machines. According to the IBM RedBook on AIX System Administration, start the syslogd daemon with the –r flag to suppress these messages (Pruett, 63). Figure 5 depicts the /etc/syslog.conf file for the AIX and Solaris client machines. The local logs are stored in the /var/log directory on AIX and /var/adm on Solaris.

Figure 5 Log AIX 4.3.3 and Solaris 8 syslog.conf

```
#AIX syslog.conf                #Solaris 8 syslog.conf
#Local logging                  #Local logging
*.info     /var/log/syslog.log  *.info      /var/adm/syslog.log
*.debug  /dev/console           *.debug   /dev/console
*.crit          *               *.crit              *
#Remote logging                 #Remote logging
*.info    @logger               *.info    @logger
```

As in Figure 5 the messages and facilities are the same except for the last line that will forward all facilities with info or above priority to the server “logger”. After the syslogd file is changed on the Solaris servers the syslogd must be started and stopped. The following commands are used to start and stop:

- # /etc/init.d/syslog stop
- # /etc/init.d/syslog start

Source: http://www.cert.org/security-improvement/implementations/i041.08.html

Once syslog is restarted it can be tested using the following command:

- /usr/ucb/logger –p mail.warning –t sendmail “this is the test for mail.warning”

Source: http://www.cert.org/security-improvement/implementations/i041.08.html

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

The command for restarting the AIX client machines are the same as located below in the server configuration. The final step for the UNIX configuration was setting up the centralized log server named “logger”.

UNIX server configuration

A server running IBM’s AIX 4.3.3 was chosen to be the centralized repository. Therefore the following procedures we reflect this version of UNIX. All the machines used in this configuration are located on the same network except the log server.

The first step I performed was deciding what facilities were going to be logged. For this installation I also chose to log all subsystems available to syslog. Also all facilities with debug priority will be sent to the console. The final item will be all critical messages will be sent to all users. Figure 6 depicts what the /etc/syslog.conf file contains. The logs will be stored in the /var/log directory and is located separate file system.

Figure 6 Log Server AIX 4.3.3 syslog.conf

```
#AIX Logger syslog.conf

*.info          /var/log/syslog.log
*.debug         /dev/console
*.crit                  *
```

On both server and client, I performed the following steps, from the IBM RedBook, to restart the syslog facility:

1. Stopping the syslogd daemon
   # stopsrc –s syslogd

2. Make sure syslogd is stopped
   # ps –ef | grep syslogd

3. Start the syslogd daemon
   # startsrc –s syslogd

I then ran “tail –f /var/log/syslog.log” to make sure information was getting written into the syslog file. Finally, with having configured the Windows 2000, and UNIX servers, I needed to configure the central repository to perform rotation and retention of the logs.

Log Retention and Rotation

With the Windows 2000 and UNIX servers are now configured it is time to discuss the log retention and rotation. I decided to keep seven days worth of logs online at any time on both the central log server and client machines and the

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

they are rotated daily. This section deals with configuring Solaris and AIX for log rotation on the centralized server and local machines.

The Solaris servers first needed to be configured to rotate their logs daily. I performed the log rotation with the Sun script /usr/lib/newslog. This script has been modified to rotate the seven logs located in the /var/adm directory. This data should be written to a “write-once/read-many device” or a “write only” device to assure the logs will not be tampered with (Manage, 3). This prompted me to make sure the system configured to be the central log server is also backed up every day and a system image is produced weekly using AIX’s mksysb utility. I have decided to keep several days of data online but the logs are rotated daily. To rotate the logs I also used information from the CERT logging management document that performs the following:

- Make a copy of active log daily
- Rename the old file so that the old file is not appended to.
- Reset file contents
- Testing the logging facility (Currently not implemented).

Utilizing information from the configuration script from Colin Bitterfield’s document on configuration of syslog, the following script was used:

```
#! /bin/sh
#
# Copyright(c) 1997, by Sun Microsystems, Inc.
# All rights reserved.
#
#ident @Z%newsyslog     1.3     97/03/31 SMI
#
LOGDIR=/var/adm
LOG=syslog.log
if test -d $LOGDIR
then
        cd $LOGDIR
        if test -s $LOG
        then
                test -f $LOG.6 && mv $LOG.6  $LOG.7
                test -f $LOG.5 && mv $LOG.5  $LOG.6
                test -f $LOG.4 && mv $LOG.4  $LOG.5
                test -f $LOG.3 && mv $LOG.3  $LOG.4
                test -f $LOG.2 && mv $LOG.2  $LOG.3
                test -f $LOG.1 && mv $LOG.1  $LOG.2
                test -f $LOG.0 && mv $LOG.0  $LOG.1
                mv $LOG    $LOG.0
                cp /dev/null $LOG
                chmod 644    $LOG
                sleep 40
        fi
fi
#
kill -HUP `cat /etc/syslog.pid`
```

To configure the AIX clients and servers for log rotations the following script was already in place but not used.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
#!/bin/ksh
#rotate_logs
cd /var/log
mv syslog.6 syslog.7
mv syslog.5 syslog.6
mv syslog.4 syslog.5
mv syslog.3 syslog.4
mv syslog.2 syslog.3
mv syslog.1 syslog.2
mv syslog.log syslog.1
touch syslog.log
kill -HUP `cat /etc/syslog.pid`
```

Both of these scripts are configured to run nightly as cron processes. When both of these scripts run the simply move syslog information from file to file for a period of seven days. After seven days the data begins to be overwritten for the previous week. Since backups are preformed on these servers nightly the logs are retained for an extended period of time.

One final and important note on the log files is security. In the log management document by CERT they say the log server should be on a separate subnet and the log files should be read only at the console. Also, the log files should be located on a separate partition to prevent a denial-of-service attack (Manage, 2). I have configured /var/log to exist on a separate partition for that very reason.

After

After providing the information for configuring the Windows 2000 and UNIX systems I have some results to discuss. Going from having to view the logs on each server and not having a routine methodology on each machine, to that of having a centralized location for managing these logs has provide a useful tool in diagnosing vulnerabilities and also problems with the servers. I did not experience any problems in configuring the Windows 2000 auditing, “event to syslog”, UNIX server or clients, and in fact it went very smoothly.

Overall results

After seeing the abundance of information that has been produced by the servers syslog and eventlogs some tuning is needed. Overall, I am satisfied with the Windows 2000 configuration and functionality, but I am currently reviewing what items are logged and a way to analyze these logged events on both platforms. The following is a list of the major outcomes of this implementation:

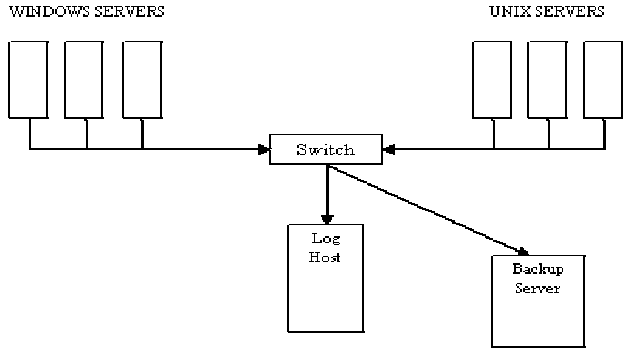
- Windows performing auditing on events.
- Windows event log properly tuned and utilized.
- Improved security of logs due to centralized location.
- Different operating systems logging to the same log facility.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

- Better granularity of logs due to centralized location.
- Ability to perform further forensics on logs.

The following image is a generalized diagram of the data flow of the logs from the servers to the log host and connection of the tape backup server.

Figure 7 Log flow from servers



Windows 2000 results

With the Purdue "Eventlog to Syslog" application running on the Windows 2000 servers, I have found the information is useful except for the amount of information that I am faced with since I configured auditing on success and failure events on every audit item. One excellent fact is that if the central server goes down for any reason, each one of the servers will still log their information locally some time. The draw back is that the information will be distributed but at least each machine logs in a uniform manner.

UNIX results

The use of an existing UNIX server to be the centralized log server has proved to be an excellent choice. Overall performance and availability has not been a problem. As discussed basically, since most of our UNIX machines are of the same vendor and operating system, if the central server were to fail configuring another machine will not be difficult. Since I have completed basic installation and configuration of the centralized facility, one of our programmers can write applications to perform real-time analysis and alarming.

Next Steps

- Change facilities that are logged to maintain efficiency of the logs.
- Configure log rotation on Windows 2000 servers.
- Create scripts, which will monitor the logs on a real-time basis.
- Perform log forensics on all logs for a better view of what is occurring on the system.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

- Configure an alerting mechanism, which will alert on specific events as they occur.
- Testing the logging facility daily.

Conclusion

In conclusion, after reading the GSEC material and becoming more alert to the importance of logging, I feel that going from an environment without a centralized logging server to having one is a large move toward good security practice. Since financial concerns played a major part in the decision of using already existing application and freeware in my configuration but as I have depicted in this case study it is a step in the right direction and still a work in progress.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

References

Bitterfield, Colin. "Configuring Syslog for Data Center Use." 3 August 2001
URL: http://colin.bitterfield.com/Syslog_for_the_datacenter.html
(9 July 2003).

CERT. "Configuring and using syslogd to collect logging messages on systems running Solaris 2.x." 29 January 2001
URL: http://www.cert.org/security-improvement/implementations/i041.08.html
(30 July 2003).

CERT. "Manage logging and other data collection mechanisms." 1 May 2001
URL: http://www.cert.org/security-improvement/practices/p092.html
(9 July 2003).

CERT. "Selecting Windows NT 4.0 event log settings." 17 Mar 1999
URL: : http://www.cert.org/security-improvement/implementations/i041.03.html
(1 Aug 2003).

Ga'had, Urbana D. "Make Windows Talk to Syslog." 10 July 2003.
URL: http://www.netadmintools.com/art284.html
(16 July 2003).

Microsoft Corp. "How to: Enable Local Security Auditing in Windows 2000." 3 June 2003.
URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;248260
(16 July 2003).

Pruett, Christian. IBM Certification Study Guide – pSeries AIX System Administration. IBM Redbooks, 2001, 62-68.

Smith, Curtis. "Eventlog to Syslog Utility." 16 June 2003.
URL: https://engineering.purdue.edu/ECN/Resources/Documents/UNIX/evtsys
(16 July 2003).

Smith, Randy Franklin. "Auditing Windows 2000."
URL: http://www.ntsecurity.net/Articles/Index.cfm?ArticleID=9633
(27 July 2003).

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Last Updated: May 22nd, 2013

Upcoming SANS Training

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

SANS Malaysia @ MCMC 2013	**Cyberjaya, MY**	**Jun 03, 2013 - Jun 08, 2013**	**Live Event**
SANS Pen Test Berlin 2013	**Berlin, DE**	**Jun 03, 2013 - Jun 08, 2013**	**Live Event**
Industial Control Systems Security Training - Houston	**Houston, TXUS**	**Jun 10, 2013 - Jun 15, 2013**	**Live Event**
Security Impact of IPv6 Summit 2013	**Washington, DCUS**	**Jun 14, 2013 - Jun 16, 2013**	**Live Event**
SANSFIRE 2013	**Washington, DCUS**	**Jun 14, 2013 - Jun 22, 2013**	**Live Event**
SANS Canberra 2013	**Canberra, AU**	**Jul 01, 2013 - Jul 13, 2013**	**Live Event**
Digital Forensics & Incident Response Summit 2013	**Austin, TXUS**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
SANS London Summer 2013	**London, GB**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
SANS Rocky Mountain 2013	**Denver, COUS**	**Jul 14, 2013 - Jul 20, 2013**	**Live Event**
SANS Mumbai 2013	**Mumbai, IN**	**Jul 22, 2013 - Jul 27, 2013**	**Live Event**
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	**Washington, DCUS**	**Jul 22, 2013 - Jul 26, 2013**	**Live Event**
SANS San Francisco 2013	**San Francisco, CAUS**	**Jul 29, 2013 - Aug 03, 2013**	**Live Event**
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	**Bangalore, IN**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
SANS Boston 2013	**Boston, MAUS**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
Critical Security Controls Summit	**Washington, DCUS**	**Aug 12, 2013 - Aug 18, 2013**	**Live Event**
Industrial Control Systems Security Training - DC	**Washington, DCUS**	**Aug 12, 2013 - Aug 16, 2013**	**Live Event**
SANS Thailand 2013	**Bangkok, TH**	**Aug 19, 2013 - Aug 31, 2013**	**Live Event**
SANS Virginia Beach 2013	**Virginia Beach, VAUS**	**Aug 19, 2013 - Aug 30, 2013**	**Live Event**
Mobile Device Security Summit 2013	**OnlineCAUS**	**May 30, 2013 - Jun 06, 2013**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**