Global Information Assurance Certification Paper

Logging Windows 2000 Events

With

Unix Syslog

SANS GIAC Securing Windows GCNT Practical Assignment version 2.1b
July 2001
Eric Yurick

1. Introduction.

In the UNIX world, as in Windows, system generated log files are important for diagnostics as well as security. The UNIX community has developed many tools and methods of logging events, manipulating messages, and alerting administrators or application systems of failures or break-ins. The Windows world has only begun to develop these tools and alerting capabilities. This paper discusses one approach to dealing with Windows NT and Windows 2000 Event logs. The challenge of sifting through copious and disjunctive system message log stores from multiple hardware and operating system platforms is common to many computer installations. Windows NT/2000 and UNIX are not the only concerns here. Any institution connected to the Internet has a router of some sort and a switch or hub. These devices generate log messages, too.

Many system administrators have been running Unix / Linux based monitoring and alerting for a long time**.** The basic idea behind a successful monitoring and alerting system is to centralize all system events at a single monitoring station. Once the information is centralized, it can be used to build an alerting system or even carry out corrective actions. [1]

The tools and techniques used in this paper specify a UNIX-centric approach to the problem of log message consolidation. If you are new to the UNIX syslog facility, then this paper is probably not for you. The reverse, Windows-centric approach could also be contrived from some of the tools used here on a Windows NT/2000 server at a computing site, but that implementation is not the focus of this paper.

2. Defining System Logging.

In the Windows NT/2000 environment, system logging refers not only to the Event Viewer provided by the operating system, but also an array of third party application logging sources. In the UNIX environment, system logging applies to the syslog service and the host of other third party application logging processes. Third party logging can be an Apache web server, Oracle database or any other non-standard logging service that exists on either Windows NT/2000 or UNIX platforms. The reason these are non-standard log stores is usually because these applications exist on a variety of hardware and operating system platforms that have no common system messaging facility (so they had to invent their own). Some of the larger applications have even integrated their logging into a common network management platform, such as Hewlett Packard's Openview, but many other applications have not. Most UNIX applications can be built to use the syslog facility and can even control which facility of syslog to use. Most commercial Windows 2000 applications use the Event Log facility or can be written to make use of it.

3. Importance of System Logs.

[1] Gerhards, Rainer. *EventReporter Home Page*. 11, Feb. 2001. Adiscon GmbH. July 13, 2001 <http://www.eventreporter.com/Common/en/Articles/EventReporter-Monitor-Windows-NT-From-Unix.asp>.

Reviewing system log messages is an often forgotten part of system administration tasks. Projects, daily administrative chores and meetings often end up consuming most of an administrator's time. System logs are important for many reasons.

3.1. Early Warning System.

Many times a hardware or software failure will show up as a warning message in a log file. With today's hardware, many components and systems will notify an operating system of trouble brewing, before it becomes a really critical problem. This can help in scheduling downtime at less expensive time periods, rather than have a mid-day crash affecting many users. Some examples of the components that can keep working in the presence of a potential problem are RAID devices, Error Correcting memory, redundant power supplies or high available systems.

3.2. Intrusion Detection System (IDS).

Most third party host-based intrusion detection systems rely on the presence of Windows Event Log, syslog or other logging facilities. Without them the IDS is incapable of finding certain signatures of an attack. A good logging system with a notification event can serve as a simple IDS or feed a commercial grade IDS.

Though many host-based IDS systems exist for Windows, most do not interoperate with UNIX syslog notifications. Those that do, do not provide good cohesion between the two platforms.

3.3. Debugging System Problems.

Reviewing the Event logs is usually one step in troubleshooting a software failure, especially after an upgrade or new installation. Often times it is the only time we review the logs of a particular system.

3.4. Legal Reasons.

Most administrators are unaware of the legal implications of disregarding system messages or not maintaining a copy of these important messages. In the event of a security breach, system logs are needed by forensic experts to reassemble the crime to determine the origin, nature and scale of the intrusion. An administrator may be asked at any time to turn over their logs to proper authorities to show either criminal intent on the part of an inside or outside intrusion, or to verify one's own innocence and help further an investigation. Centralizing the storage of these messages is an important step in furthering the security of a computing installation. Understanding the process of logging is important to effective system administration.

4. Central Storage Design.

As discussed earlier, system logs are important for many administrative reasons. The problem becomes one of volume desired and time available. We want these system message stores to be

very detailed on the occasions described above; however, most of the time we never need to read them. Let's face it, configuring the Event Log and auditing settings on a Windows 2000 server isn't something that we place high priority. Multiply this by the number of servers in the installation and add in different operating systems, like UNIX, and the problem exacerbates. It can become too much to deal with.

You need to centralize logging in order to gain perspective on all of this information. This is not a quote from Dr. Richard Carlson author of the "Don't Sweat the Small Stuff" books. My preference is to log everything to a UNIX machine and combine, compare, correlate and compress it there. Alternately, you could also log everything to a Windows machine, but have to be a much better programmer and spend more money to tie everything together. In Section 8, I will discuss in detail how to build the server, size it and where to put it, but first you need to have an understanding of the two event notification systems.

5. The Syslog Protocol.

System messages in a UNIX system are handled locally by syslog.[2] The syslog protocol also defines the transport of these notification messages. Syslog is a simple protocol without any acknowledgement of reception of a message from the "collector" or server back to the transmitter (or client). There is not any complex authentication or coordination present between the syslog sender and receiver. Hence, there are probably many devices and systems sending syslog messages to a server that is no longer present.

An informational RFC defines the BSD Syslog Protocol its uses, configuration and programmatic interfaces[3]. This RFC states that syslog is not without its faults. It does not authenticate the sender on the network, secure the data, or ensure delivery. Possibilities of attack against the syslog channel are plenty. Syslog is however, a well-defined, customizable and simple protocol. A secure syslog system has been proposed, but for now, good network and system security practices can overcome these limitations and insecurities.

6. The Event Log System.

The Event Log is the Windows NT/2000 corollary to the UNIX syslog. It is one of the most important problem troubleshooting tools available for Windows 2000. It can also be an important tool for improving performance, for preventing problems before they occur, and for keeping your Windows servers secure.

The Event Log is actually made up of several logs, each with its own individual settings. Windows 2000 Professional version still has the same three that were standard in Windows NT:

[2] Bing, Matthew and Erickson, Carl. "Extending UNIX System Logging with SHARP." *Proceedings of the 14th Systems Administration Conference (LISA 2000).* 3 Dec 2000. Grand Valley State University.
[3] C. Lonvick. *The BSD Syslog Protocol.* Cisco Systems. May 2001. http://www.ietf.org/internet-drafts/draft-ietf-syslog-syslog-12.txt. July 22, 2001.

the Application Log, the Security Log, and the System Log. Windows 2000 Server has added several additional logs, present only if the service is installed. The new Event Logs are named for the services they represent: Directory Service Log, DNS Server Log and File Replication Service Log. The breakout of these new Event Logs in Windows 2000 was necessary since many more messages are written to Event logs in general. The Event Log viewer has changed somewhat in Windows 2000 from Windows NT. It is now an MMC (Microsoft Management Console) utility, though still available off of the Start button and Administrative Tools.

Microsoft has provided interfaces into the Event Log in a variety of methods for programmers. In Visual Basic, the functionality is built in with the App.LogEvent object. [4]

```
app.LogEvent(LogBuffer as String, [EventType])
```

Microsoft's Technical Article number Q154576 discusses this and other programmer interfaces into Event Log. The log files themselves are not stored in a standard text or database format, which has poised a problem for many administrators and has provided an opportunity for many third party application writers.

The Windows NT/2000 Event Log system does not have any way to push messages onto the network. Each Windows system must maintain its own store of logs. In order to tie together multiple systems you need to purchase a large, third party Windows NT/2000 management platform.

7. Central Server Configuration.

From a topology point of view the central system message storage design would be as shown in Figure 7-1. Creating this central system message design can be accomplished in a series steps with very little impact on system availability. In the next sections, I will go into detail about the components of the event message system. The overall plan involves minimal reconfiguration of any existing equipment.

[4] Toby Patke. *Writing Messages to the WinNT Event Log with a COM Component.* July 1997. http://www.15seconds.com/issue/990930.htm. July 22, 2001.

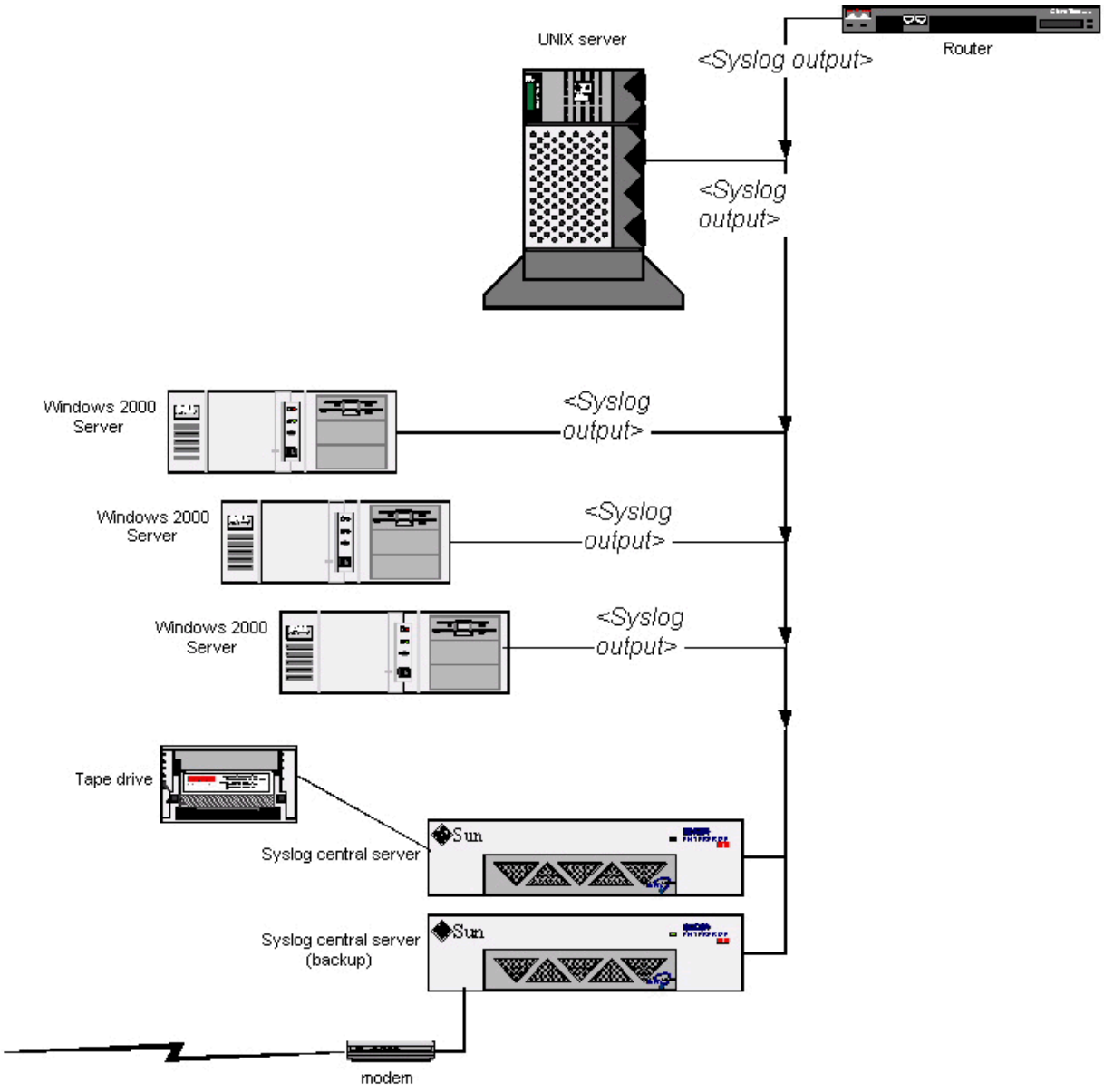

Figure 7-1. Central Server Topology.

7.1. Hardware Requirements.

Storage and reliability are the key components of a central system message store. The machine need not be too powerful, because it should only be performing one function. For a site of 3000 users on a network containing 40-60 server devices, an available Sun Sparcstation 4 or 5 with upgraded internal hard drive, an additional external disk drive and available CD-ROM disk is adequate for the task. Install just the recommended amount of memory for the version of the operating system selected. There is not any need to have fast CPU clock speed or large amounts of memory, since the syslog daemon and the log scanning applications use very little system resources. Preferably the machine needs its own backup software and device as well as a

secondary communications channel (i.e., modem). For my average site, I found that one gigabyte of message store was needed on average. Make sure you have at least double your average log utilization if not quadruple. Remember, disks are relatively inexpensive.

7.2. Software Requirements.

The software configuration of the central system message server includes the following components: a well secured operating system, a customized syslog configuration file, log message filters and rotational scripts. Backup software is a trivial decision since syslog files are ordinary UNIX text files, not databases. Either in-house scripts utilizing standard operating system commands (i.e., ufsdump) or freeware (for example, amanda) will suffice for backup.

7.3. Security Considerations.

A central system log machine should be a specially hardened, physically separate and obscure machine. It is important that the machine be physically secure and kept away from other hardware outside of central computing facilities. A physical break-in shouldn't involve the machine tracking events. Ideally, the hardware and operating system should be different than other equipment on a site. Good computer hackers know how to cover their tracks (erasing or manipulating logs) on the machines that they can hack, but few are disciplined to detect the existence of remote logging and fewer can hack multiple operating systems in order to cover themselves. Having a multiple vendor approach can be beneficial.

On both the Windows NT/2000 server as well as the UNIX servers, the log files must be protected with Access Control lists. The default permissions for the Event log files (*.evt) only allow the Administrators access to these files. The Solaris default permissions are 600 for the log files that come with Solaris. When adding additional log file capabilities to any UNIX variant (see the syslog.conf examples below), take care to make root the owner with only read and write bits set (600).

On the syslog server, follow the guidelines of hardening the operating system allowing only the Network Time Protocol (NTP) port (tcp 123), and the syslog port (udp 514). Depending on the filtering software selected, the registered (1024-49151) and dynamic ports (49151-65535) may need to be available for the reverse connection for NTP and SMTP.

Choose a secure network in which to place this server, not in a Demilitarized Zone (DMZ), and not where users have direct access on the same LAN or VLAN. To avoid packet sniffing, the servers should be on switch ports without packet capture or port mirror capabilities. Too many network devices have advanced Remote Monitoring (RMON) I and II features with little or no security to accessing those features. Remember that the syslog protocol does not secure the data in transit; you must secure your network.

For my project, a commercial firewall product (in its smallest form) was purchased. However, for the Solaris platform, the "ipfilter" package or the Sun Solstice product, Sunscreen Lite, are freely available alternatives.

8. Steps in Building the Server.

The first step in the project is to acquire the pieces of hardware for the central system message servers. I recommend two nearly identical workstation devices. This makes the build of the second machine easier since you will be able to completely clone the first. Choose an obscure machine hostname for each server. Don't call the machines "syslogserver" or "logcentral" as is done just as an example in this paper. Configure the network domain name service (DNS) with static IP address of this server. This is the only change necessary to your network DNS.

Freshly install the latest UNIX operating system and patches from CD while the system is completely off of the network. This is the only way to ensure a tamper-proof server. Secure the operating system. Configure the syslog server software (see Section 8.1 below) to accept all messages to just one file for now (/var/adm/messages). Test syslog messages from the server to the server itself (See Section 12.3 for the UNIX test and Section 12.1 for verifying the test). Install the TCP/IP filtering software (see Section 7.3) and configure the central server to accept only the syslog channel from your internal network. Configure and test syslog from another UNIX server to the central server. Install and configure the Adiscon Event Reporter software on a Windows NT or Windows 2000 server to direct all Windows Event Logs to the central message server. Verify that the Windows server can send to the syslog server by logging into the server and looking at the central message server /var/adm/messages file (Section 12.1 example). You should see your login attempt.

Once both a UNIX and Windows NT/2000 server have successfully sent to the syslog server the real work begins. Install and configure the log rotation (Section 9.1 or 9.2) and scanning applications (Section 9.3 or 9.4). Configure the notification system (usually E-mail) and backup notification system (modem). Configure the tape backup software to avoid the log rotation schedule, but be able to backup every version level of the log files.

8.1. Syslog.conf.

The configuration file for syslog is a file called syslog.conf. It usually resides in the /etc directory. This file can be quite daunting at first glance. I recommend you save a copy of the original operating system version and start anew. Here are my recommendations. Every UNIX operating system has man pages, read them. Then apply the following lines to your /etc/syslog.conf file:

```
# This is a basic example of the syslog.conf on the central machine
*.info                  /var/adm/messages
kern.debug                 /var/log/kern
local0.notice              /var/log/local0
local1.notice              /var/log/local1
```

The first line is a comment. The second line begins with the format: **facility.level**. This line

indicates that all informational (info) level messages or greater from any message facility (kern, auth, daemon, etc) are to be sent to a file /var/adm/messages. The informational priority is the second lowest, only debug contains more verbosity. The syslog priority levels are in increasing order of verbosity: emergency, alert, critical, error, warning, notice, info and debug. The syslog facilities are in order of importance: kern, user, mail, system, auth, lpr, news, uucp, cron, local0-7 and mark. There are more facilities available in some versions of UNIX, those listed here are for Sun Solaris. Tabs, not spaces, separate the message type from the action. The remaining lines send more specific message levels and facilities to separate files that *must exist before starting the syslog daemon*, in order for messages to be written. More elaborate facility.level pairs can be constructed to filter for particular groups of messages, but that is outside of the scope of this paper. In my production syslog.conf file, I expanded the sorting of individual facilities into separate files (i.e., the last 3 lines).

The next example demonstrates a simple client syslog.conf configuration. Certainly, more should be added and some client level "filtering" could be added. It is still desirable to maintain some system messages on the machine for quick diagnostics when visiting a particular server. This one configuration line sends all messages to the machine called "logcentral" and keeps nothing on the local machine. The log central machine would be responsible for combining all client machines messages into a single file (i.e., /var/adm/messages in the example above) as well as filing different facility messages into there own log file or application. This setup will enable a log analysis program to scan through the logs. This configuration is only a starting point for the clients. After later analysis at the central server, this could be later modified to reduce overall logging traffic by selecting only certain facilities and levels of syslog.

```
# This is an example of the syslog.conf on a client UNIX server
*.info                          @logcentral.org
```

9. UNIX Tools.

In addition to the syslog daemon, two critical pieces of software are required to manage the ever-growing logs. Note that any script must signal (HUP) the syslog daemon process to finish writing and initialize its file pointer. Otherwise, the renamed file will continue to grow.

9.1. Newsyslog.

As important as collecting and analyzing all of the system messages is the role of cleaning up and expiring the older ones. The log messages can grow very rapidly. One tool that is provided by some UNIX operating system vendors is a script called newsyslog. It is generally run from the UNIX job scheduler, cron. The newsyslog script is a simple UNIX Borne-shell script that rotates the log files from syslog. It must be customized to reflect any changes to the /etc/syslog.conf file described earlier. Below is an example of some of the customizations necessary for the "logcentral" server.

```
#!/bin/sh
```

```
# example customized newsyslog
LOG=messages
cd /var/adm
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
#
LOGDIR=/var/log
LOG=local0
if test -d $LOGDIR
then
      cd $LOGDIR
      if test -s $LOG
      then
            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
      fi
fi
#
LOGDIR=/var/log
LOG=kern
if test -d $LOGDIR
then
      cd $LOGDIR
      if test -s $LOG
      then
            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
      fi
fi
#
#
LOGDIR=/var/log
LOG=local1
```

```
if test -d $LOGDIR
then
      cd $LOGDIR
      if test -s $LOG
      then
            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
      fi
fi
#
sleep 4
#
kill -HUP `cat /etc/syslog.pid`
```

9.2. Logrotate.

Logrotate is a better script than newsyslog in handling the rotation of logs. Several of the useful features include automatic compression and rotation based on size. Originally a Linux RPM installable package, logrotate, has since been "ported" to other UNIX platform packages (i.e., Sun Solaris package install) and is also available in source form. Below is an example of its use in our simple logcentral configuration:

```
# sample logrotate configuration file
   errors root@logcentral.org
   compress

   /var/adm/messages {
      rotate 5          # keep 5 copies
      size=10M          # rotate if the file grew to 10 Meg
      postrotate        # must signal the syslogd process to initialize file pointers
            /sbin/killall -HUP syslogd
      endscript
   }
   /var/log/kern {
        rotate 5
        weekly
        missingok
   }
  /var/log/local0 {
        rotate 5
        weekly
```

```
}
/var/log/local1 {
        rotate 5
        weekly
}
```

9.3. Swatch.

Available from the website: http://www.cerias.purdue.edu/coast/archive, or ftp site: ftp://coast.cs.purdue.edu/pub/tools/unix/logutils/swatch, swatch, is a perl script based matching tool. This log message filtering program was rewritten several years ago to use perl. Many perl modules are necessary to be added to a standard perl installation in order to build the swatch utility. This makes for a more difficult filter to setup. Though I had worked with swatch in the past, I found logcheck easier to install and it also contained a nice array of templates to help with notification setup.

9.4. Logcheck.

Logcheck is one of many BSD or GNU freeware licensed script tools that scan logs for patterns and can take action. The default action of this tool is to E-mail interesting message logs to an administrator. Because it is a script (logcheck.sh), it is completely customizable as to the alert facility (E-mail). This tool is downloadable from: <http://www.psionic.com/download>. Logcheck was originally designed for the Gauntlet Firewall and the TIS toolkit.

The configuration of this script is contained in several files, each appropriately named:

logcheck.hacking – defines keywords from syslog that describe hacking attempts
logcheck.ignore – defines keywords to completely ignore
logcheck.violations – defines keywords of system logon attempt failures
logcheck.violations.ignore – defines keywords in message violations to not report

Logcheck is run from the UNIX crontab and works well with the logrotate script described earlier. A sample output from the logcentral.org below shows a test syslog message sent from the console of logcentral itself (using the UNIX logger utility described below) as well as an actual system message.

```
Message  1:
From logcentral Sat Jul 14 11:38:30 2001
Date: Sat, 14 Jul 2001 11:38:29 -0400 (EDT)
From: EricY <root>

Unusual System Events
=-=-=-=-=-=-=-=-=-=-=-=
Jul 14 12:38:12 logcentral logcentral : [ID 702911 daemon.error] help help test
```

Jul 14 11:38:24 [192.168.123.193.4.160] EvntSLog:2508: [WRN] Sat Jul 14 11:31:07 2001: ERIC-INSPIR-7K/w32time (54) - "The Windows Time Service was not able to find a Domain Controller. A time and date update was not possible."

10. Windows Tools.

10.1. Adiscon Event Reporter.

The existence of this type of product makes a multi-platform, central system message logging possible. The Adiscon Event Reporter converts the Windows NT/2000 workstation and server system event logs into syslog messages. This product has a shareware license that must be registered after 30 days and has a nominal cost per installed machine associated with it. It can be obtained from www.eventreporter.com. Read the article posted by Gerhards Rainer on the product website. Its features include the capability to send to a backup syslog server and/or E-mail messages to an administrator (based on Windows NT/2000 message type). Figure 10.1-1 shows the General Configuration tab with the syslog server selection.

Each Windows NT/2000 Event log service type is separately configurable. The Windows NT/2000 System Events are configured as shown in Figure 10.1-2. Figure 10.1-3 shows that any of the standard UNIX syslog facilities can be specified in each of the setup tabs.

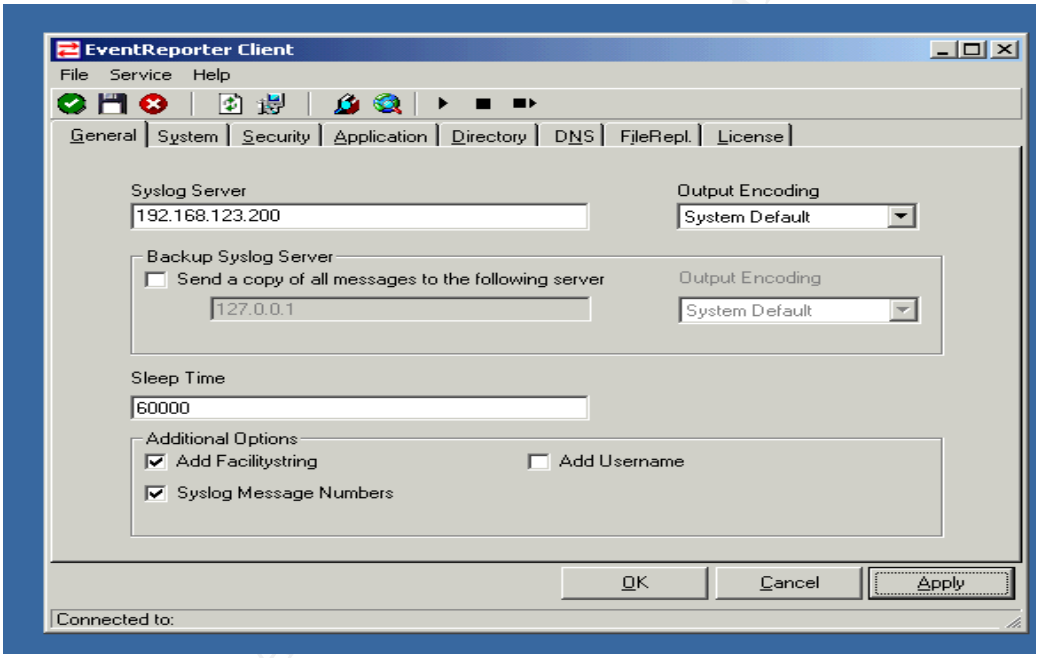

Figure 10.1-1. General Tab.

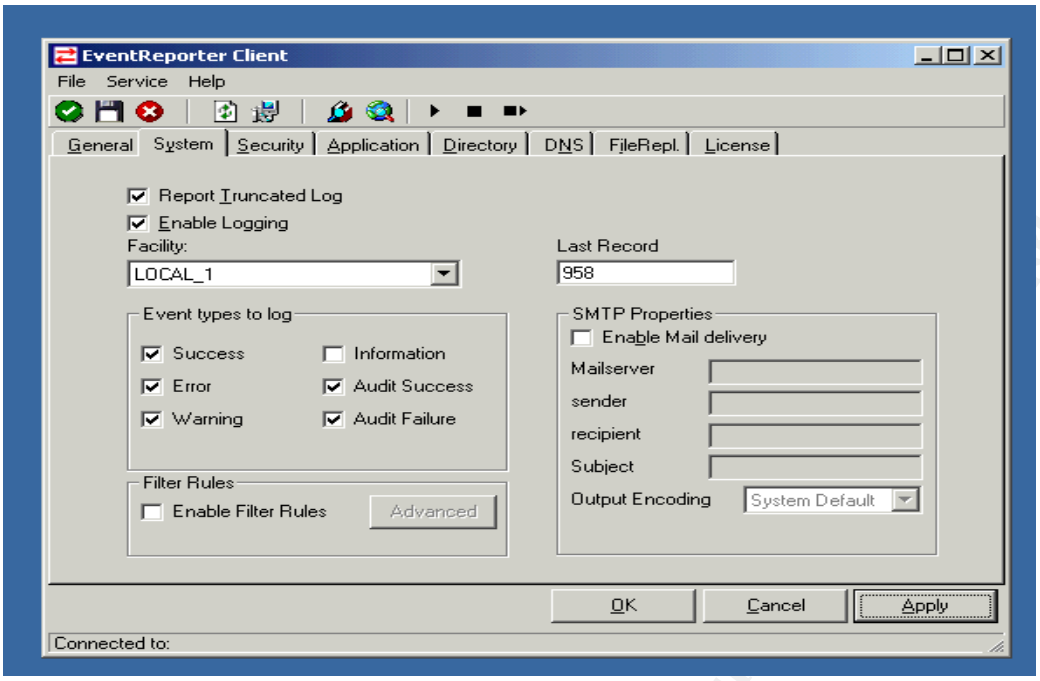

Figure 10.1-2. System Tab

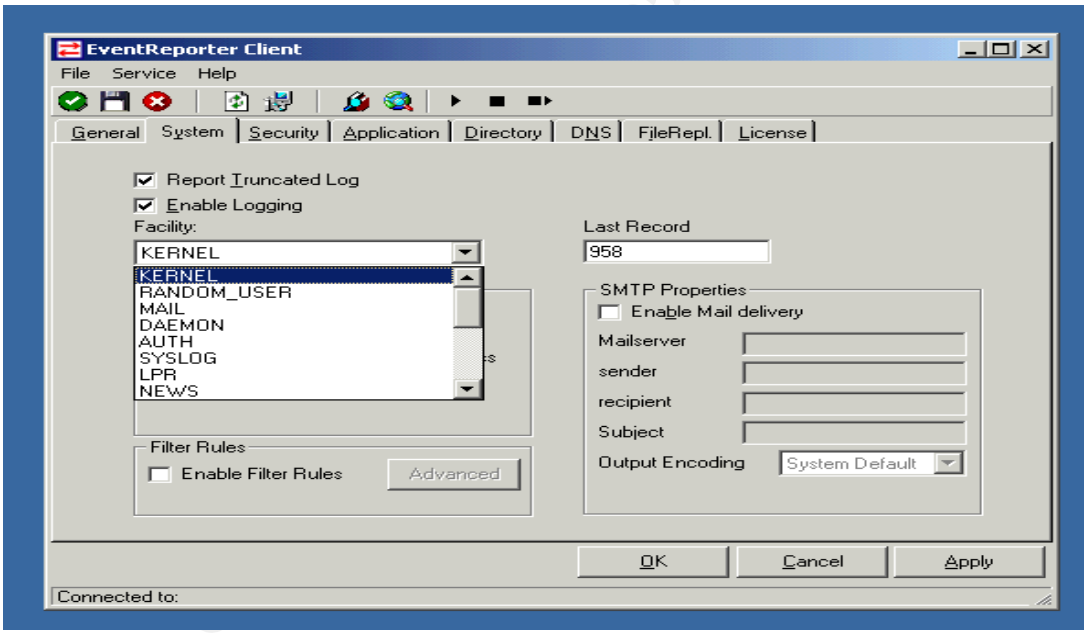

Figure 10.1-3. System Tab with Facilities view

In the configuration of this product, a good technique is to send each type of Windows Event Log message (System through FileRepl. Tabs) to a separate Local facility. Remember that the central server is going to combine each of the message facilities together into one file in order for the logcheck or swatch program to operate, but the checking program can glean information about the type of message from the facility used. Standardizing on Local0 through Local5 Facilities for corresponding Windows Event Log means better handling of the messages at the

central server later.

10.2. Netal Clsyslog.

The Windows Event Log message facility is generally used by most of today's commercial applications; however, custom applications and scripts can benefit from this service if they have an interface into the Event Log message system. In the case of the context of this paper, it may be useful to record events directly to syslog itself from within a Windows application or script. The Clsyslog toolkit is available from <http://www.netal.com>.

10.3. Syslogx 1.0.

Syslogx is another product to syslog-enable Windows applications. This is a tiny freeware product available from <download.cnet.com>. Intended for a software developer to write application using calls into a simple Application Programming Interface (API).

10.4. Windows Syslog Servers.

10.4.1. 3Com Wsyslogd.

A Windows NT/2000 only site might be able to implement a syslog server on a Windows server itself using one of many syslog server products that are available for free from 3Com and downloadable from the site: <http://infodeli.3com.com/software/utilities_for_windows_32_bit.htm>. This server implements a Windows service to capture syslog messages. A lightweight Windows syslog viewer is included with this download (see Figure 10.4.1-1).

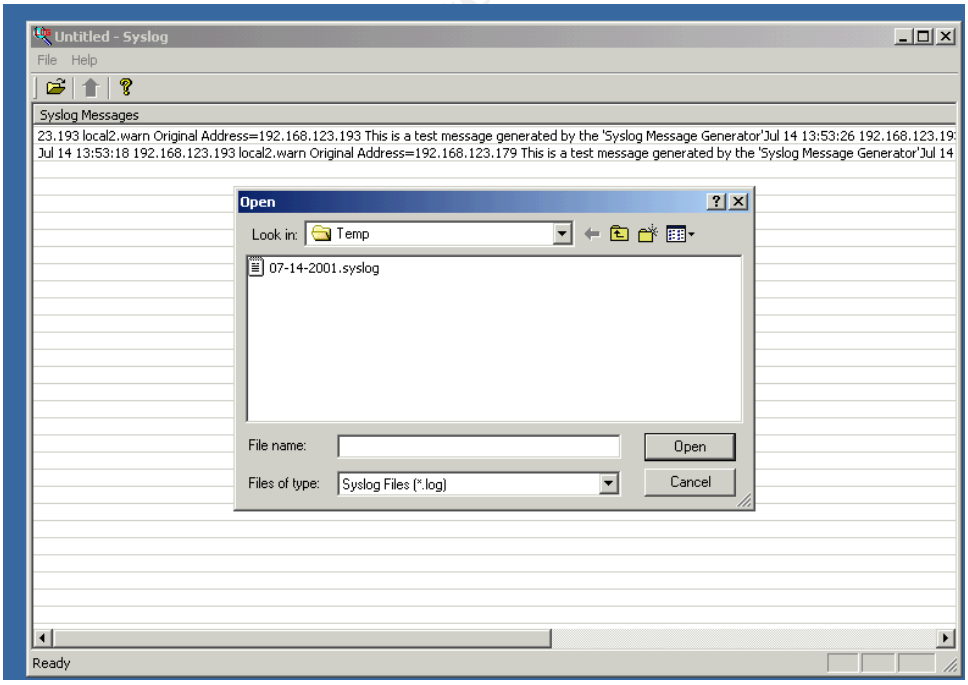

Figure 10.4.1-1. 3Com Wsyslogd.

10.4.2. SL4NT.

SL4NT is another free Windows NT/2000 syslog server. It is downloadable from <www.netal.com> and installs as a control panel applet. The configuration is simple (see Figure 10.4.2-1). This company also has released a more robust server for a small charge. This free product and its larger version allow a NT administrator to capture UNIX syslog messages not only into a log file, but put them into Event Log format. This is handy for the Windows NT/2000 installation with only one or two UNIX machines who wish to have syslog messages report to a Windows 2000 server Event log.

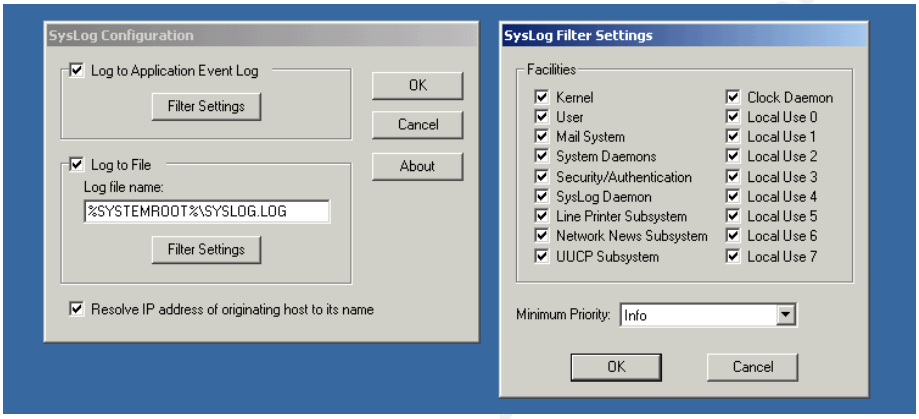

Figure 10.4.2-1. SL4NT Windows Syslog to Event Viewer Control Panel Applet.

11. Switch and Router Configurations.

In some computer installations, the networking personnel are a separate department with independent system monitoring tools from their system administrator colleagues. If the turf battle allows, router and new layer three and four switch devices have the capability of sending messages to a syslog server. The author has investigated the following two vendor products. I again recommend standardizing on a unique facility for all switches and routers.

11.1. Cisco Router Configuration.

The current Cisco Internetwork Operating System (IOS) support several methods of message logs. The system configuration keyword, logging, is used to enable syslog messaging, choose the interface to source the messages, select the syslog server, determine the syslog facility to use and what level of message to send to that facility. The following four lines select the informational level messages, employ the local6 facility, source the messages from Ethernet0/0 and send to the

logcentral server discussed in this paper.

```
logging history informational
logging facility local6
logging source-interface Ethernet0/0
logging 192.168.123.200
```

Additionally, on a Cisco IOS router, any access list, IDS or firewall inspection can contain the keyword, log, on the end of the line to include "hits" on this rule to fire off a message to the selected server.

11.2. Cabletron/Enterasys Router Configuration.

Since this product line from Cabletron (now renamed, Enterasys) SmartSwitch Router (SSR) emerged with its own IOS it is important to note the different syntax of this device. In the native mode of the IOS (there is now a Cisco IOS compatible mode), the syslog parameters are contained in one line:

```
system set syslog level info facility local6 source 192.168.124.100 server 192.168.123.200
buffer-size 10
```

12. Testing your Syslog server.

A robust syslog message server should be tested not only for functionality, but for strength as well. A security incident may involve a Denial of Service attack and cause many machines to generate hundreds of messages per second to a syslog server. Some tools are referenced below that can test the syslog server.

12.1. Syslog Server tester for Windows 2000.

One tool, which tests functionality and strength, is a shareware tool from Kiwi software. It is downloadable from: <http://www.kiwi-enterprises.com>. One of the features of this tool is its ability to send corrupt syslog data and send bursts of traffic to the server from randomized hosts. The application (shown in Figure 12.1-1) can also be used to test the syslog.conf file to determine whether messages of particular Facility and Level that are supposed to be captured are indeed saved in the log file, and those that are not are omitted. Using the screen setup in Figure 12.1-1 to send to my target logcentral syslog server at 192.168.123.200 (an RFC 1597 address[5]), several packets were sent to my syslog server.

[5] *rfc1597 - Address Allocation for Private Internets*. March 1994. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. de Groot. July 21, 2001. http://www.faqs.org/rfcs/rfc1597.html.

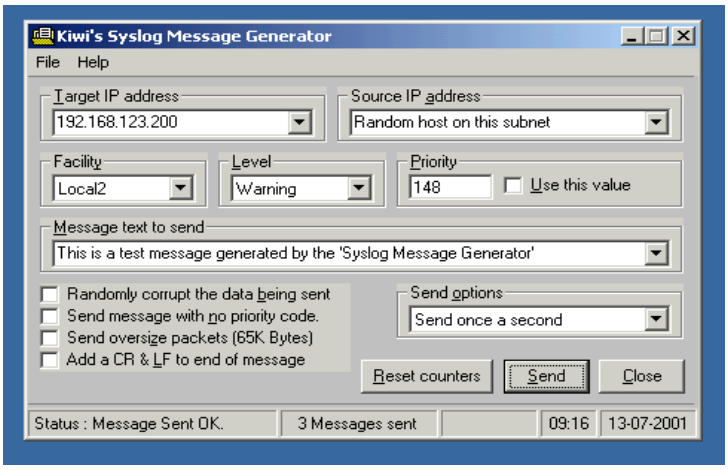

Figure 12.1-1. Kiwi Syslog Message Generator.

After sending the messages using this application, look for the log messages on your syslog server as follows:

```
# tail -3 messages
Jul 13 10:23:38 [192.168.123.193.4.150] Original Address=192.168.123.179 This is a test
message generated by the 'Syslog Message Generator'
Jul 13 10:23:39 [192.168.123.193.4.150] Original Address=192.168.123.135 This is a test
message generated by the 'Syslog Message Generator'
Jul 13 10:23:40 [192.168.123.193.4.150] Original Address=192.168.123.147 This is a test
message generated by the 'Syslog Message Generator'
```

Repeat this test to a client UNIX machine and verify the results in both the client log files and the central log server.

12.2. Event Log testing for Windows 2000.

Windows NT/2000 clients can be tested using the Windows 2000 Resource Kit program, logevent. From the command line an example follows:

```
logevent -s F -c 13 -r "User Event" -e 44 "Test message"
  Logevent command completed successfully!
```

Giving this input, the output will first show up in the Event Viewer Application Log as type Failure Audit (the –s F option). Further trace this message to the central syslog server.

12.3. Syslog Server tester for UNIX.

Most UNIX operating systems include a tool called, logger, which can be used to test the syslog

daemon. It accepts input from programs and keyboard via the UNIX stdin. An example of generating a syslog message to the daemon facility with a level of error is as follows:

>echo “help help test” | logger –p daemon.err

13. Time Synchronization.

With any system message logging, the importance of time synchronization is very important. Windows 2000 will by default attempt to synchronize time with a w32time server. The PDC FSMO server at the root of a Windows 2000 forest should be configured to gather time from an external source.[6] The capability of synchronizing a Windows 2000 machine using a SNTP is now built into the operating system. However, a Windows NT or other Windows clients do not have a standard time synchronization service. A time solution for Windows NT and lesser Windows clients can be found from shareware. The previously referenced, <www.netal.com> website contains the Unitime 2.1 suite.

14. Conclusion.

Nobody has time to sieve through the voluminous, disparate and mostly unimportant logs on every machine in there computing installation. The installation of a central logging machine for security and ease of maintenance in a mixed UNIX and Windows NT/2000 is possible with the use of available tools. Not only is it possible, it makes practical sense to expend the effort to make one central system logging store rather than one per operating system. The idea here is to quickly deploy generic agents on all of the servers in an installation, capture as much as the information as possible to a single machine (or redundant machine pair), then customize just one tool to sift through the data and rapidly discard useless information and maintain only “interesting” events for a defined period of time. The syslog protocol has been providing the capability of central logging to UNIX systems for years. The Event Reporter tool allows for the extension of this centralized effort to include the Windows NT/2000 server platform.

[6] *Windows 2000 includes the W32time.* S.I.E. Systems. 14 July 2001. <http://www.siesystems.com/windows_2000_w32tim.htm>.

References

Gerhards, Rainer. *EventReporter Home Page*. 11, Feb. 2001. Adiscon GmbH. <http://www.eventreporter.com/Common/en/Articles/EventReporter-Monitor-Windows-NT-From-Unix.asp>. July 13, 2001.

Windows 2000 includes the W32time. S.I.E. Systems. <http://www.siesystems.com/windows_2000_w32tim.htm>. July 14, 2001.

Y. Rekhter, B. Moskowitz, D. Karrenberg, G. de Groot. *rfc1597 - Address Allocation for Private Internets*. March 1994. http://www.faqs.org/rfcs/rfc1597.html. July 21, 2001.

Bing, Matthew and Erickson, Carl. "Extending UNIX System Logging with SHARP." *Proceedings of the 14th Systems Administration Conference (LISA 2000)*. 3 Dec 2000. Grand Valley State University.

C. Lonvick. *The BSD Syslog Protocol*. Cisco Systems. May 2001. http://www.ietf.org/internet-drafts/draft-ietf-syslog-syslog-12.txt. July 22, 2001.

Toby Patke. *Writing Messages to the WinNT Event Log with a COM Component*. July 1997. http://www.15seconds.com/issue/990930.htm. July 22, 2001.