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: Using Syslog in a Microsoft & Cisco Environment

This case study details the development of a centralized logging infrastructure using Syslog in a Microsoft and Cisco based environment. The primary technology piece that our company employed was the Kiwi Syslog Daemon1 for Windows. While Kiwi has already been the topic of a wonderfully informative SANS paper by Brian Wilkins2, I have sought to build on his work by discussing ways to extend the product's functionality and by focusing on practical uses of the technology.

Copyright SANS Institute

Author Retains Full Rights

AD

Case Study: Using Syslog in a Microsoft & Cisco Environment

Dan Rathbun

June 27, 2003

GSEC Practical Assignment Version 1.4b

Abstract

This case study details the development of a centralized logging infrastructure using Syslog in a Microsoft and Cisco based environment. The primary technology piece that our company employed was the Kiwi Syslog Daemon[1] for Windows. While Kiwi has already been the topic of a wonderfully informative SANS paper by Brian Wilkins[2], I have sought to build on his work by discussing ways to extend the product's functionality and by focusing on practical uses of the technology.

This study details various aspects of the project including:

- A description of the computing environment both before and after the deployment of Syslog.
- A discussion of the design and product choices that were made.
- Deployment of the Kiwi Syslog Daemon as our central server.
- The configuration of a variety of networked devices as Syslog clients.
- A set of archiving procedures designed to enhance the integrity of the logged data.

1.0 Defining the problem

Many companies pursue information security by acquiring the latest technology or funding trendy projects, often times at a substantial cost. These decisions are frequently driven by industry news reports or other media induced hysteria. While projects of this sort can certainly offer some value they are commonly undertaken before more essential processes are in place. A security program strong on fundamentals can contribute to the success of these more cutting edge efforts.

Our company, like many others, was struggling to get its arms around the wealth of security and performance related data on its network. We managed hundreds of homogenous devices distributed around the globe each with its own log file. These log files were seldom reviewed unless a problem occurred so our security posture was necessarily reactive. Clearly there was a need for improvement at this most rudimentary level. It was with the hope of developing a strong foundation for our Infosec program that I designed and implemented this centralized logging project. The goal was to consolidate our log data and to ensure its confidentiality, integrity and availability (CIA).

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

A project of this nature certainly is not glamorous. Reviewing log files is tedious work and must be done relentlessly in order to extract their full value. Fortunately the right technologies creatively applied can help to automate much of this effort allowing us to focus our attention on the items most worth scrutinizing.

An appealing aspect of this project is that it can be completed with little or no budget. Due to today's weak economy this fact alone put the project right at the top of our priority list. Its affordability stems from several technical factors. Most devices speak syslog natively or can be made to speak syslog using freeware or shareware utility programs. The demands on the syslog server itself are meager so existing hand-me-down hardware may very well suffice. In fact the bulk of the cost to our organization was for the labor and the creative energies expended. These soft costs are more easily approved than capital expenditures in fiscally constrained times.

2.0 Design considerations and product choices

There were several design goals for this project. We wanted to capture the log data from every device on the network at all times. We wanted to have one central place where all of this information would converge and be safely stored. It was important to be able to correlate events across multiple devices. For this reason a common time source and log format were desirable. We also wanted historical access to the log data. It needed to have integrity in case the logs were required for use as evidence. This demanded well designed archiving processes and handling procedures.

From a usability perspective we wanted the log data to be easily reported on and manipulated by common database tools. A final requirement was to have the system immediately alert our administrators whenever interesting items appeared in the logs.

Our company was not experienced with Linux therefore a Microsoft based solution was more sensible. This was discouraging however because it seemed like the most feature-laden products in this arena were UNIX based. Syslog[3] itself is native to UNIX and it seemed like much of the functionality required might need to be sacrificed in order to maintain the system on a Windows platform. That is why discovering the Kiwi Syslog Daemon for Windows was so exciting.

The Kiwi product is standards based, robust and has an intuitive interface. It can save the log data in a number of different formats or if you like you can create a custom format to meet your needs. It includes excellent alerting capabilities searching through any of the fields for interesting text strings and then triggering the action for which you have configured it. The available actions include sending an email, an SNMP trap, an ICQ instant message, playing a sound or running a script or external program. These last two options represent one of the

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

most powerful features of Kiwi, the ability to extend its functionality using external programs. The benefits this can provide are limited only by your imagination.

Because Kiwi runs on the Windows platform our company is able to enjoy the familiarity of that environment. Once the underlying operating system was hardened the goal of a secure and centralized repository for our log data would be met. It seemed that all of our criteria for success would be satisfied by this design and indeed that is how it turned out. Now let's look at some of the details of the actual deployment.

3.0 Syslog server deployment

On our corporate network we had several hundred devices that we desired to monitor via Syslog. We made the choice early on not to monitor desktop computers but rather to focus on all of our perimeter and infrastructure devices such as servers, routers, switches, printers, etc. This reduced the performance burden on our Syslog server. As it turned out Kiwi is capable of handling approximately 500,000 messages per hour so the software could certainly handle the load.

Because this was a grassroots project we were hoping to accomplish it with readily available hardware. We employed an old Pentium II based machine to become the centralized server. Even though this server receives on average 70,000 Syslog messages per hour it is capable of easily handling the load. We have observed that even when the system receives a high volume of messages its processing power appears not to be impacted.

We installed Win2k Professional as the operating system on the Syslog server. The machine had no need of the features of Win2k Server so it made no sense to spend the extra money or to install unnecessary code. After the OS was installed we applied all the service packs and hot fixes. Using the Microsoft Network Security Hotfix Checker[4] we were able to identify the needed security patches and apply them.

Once the server was installed and upgraded it was ready to be hardened. We use The Center For Internet Security's Win2k Professional Gold Standard Security Template[5] for all of our secured machines. We applied this template and effectively locked down the machine. It is recommended that you take the time to read through the template, and do not just apply it blindly. Get familiar with each of the configuration choices that it makes and learn why it does what it does. Failure to do this can result in some surprises. In this case however the default template worked perfectly.

The Kiwi install itself was pretty straightforward. It can be installed as a service, and will minimize to the system tray if you so desire. Using the default

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

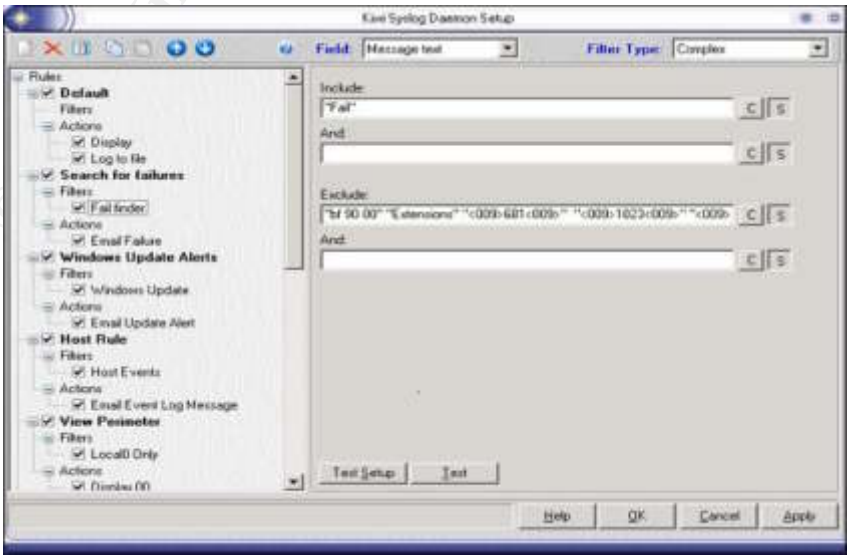
configuration options it can be put to immediate use as a functioning Syslog server.

According to the Kiwi Syslog Daemon help file[6] it processes the incoming messages through a set of configurable rules. Each rule consists of one or more filters and one or more actions. Each message received is processed by the rule engine and compared against each rule in turn. If all the filter conditions are met the program will perform the specified action or actions in order.

This ends up being a pretty powerful feature as you can have up to 100 rules and each rule can have up to 100 filters and 100 actions. As far as alerting functions go the need for anything more flexible or robust is unimaginable. We have just begun to scratch the surface of what can be done with it. I will detail some of the more useful rules that we came up with later in this study.

By default Kiwi is configured with one rule that contains zero filters and two actions. When no filter is defined the actions get applied to each and every message. The default actions are *Display* and *Log To File*. What this rule accomplishes is pretty much what you would expect. Each and every message gets displayed on the screen for viewing and gets saved to a file for archiving. It is recommend that you leave this rule unchanged and make sure it is always the first rule listed.

As an interesting test of the alert functionality we created a simple rule that scans each message for the word “Fail” and emails our administrators when that condition is met. This turned out to be a pretty useful rule both for security and performance management. As we got used to some of the more common alerts that this generated we were able to suppress some of the less useful messages by entering a list of keywords to exclude. This really helped to fine tune the alerts which kept the attention level of our administrators from waning. Here is an image of the rule definition that may help you to envision all of this.



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

Even though the exclude statements suppress some of the alerts each and every syslog message is still being displayed to the screen and saved to a text file because of the default rule. This is critically important because you will want to save every message for the purpose of historical analysis.

Now that we had established the Syslog server and had gotten some basic functions working it was time to configure our networked devices to use it. This was not a difficult task once we determined the steps to take for each device. The hardest part was finding that information. As a result I want to spend some time detailing the more common devices that we dealt with.

4.0 Configuring the network devices

Before we began configuring all the devices a little planning needed to be done. We needed to understand a little about how the Syslog protocol itself was structured in order to be able to design an efficient plan for using it at our company. Here are the pertinent details of how we deployed the Syslog protocol.

Each Syslog message includes a priority value which is made up of two parts. The value is expressed as facility.severity where facility is a type of category for the message and severity is its relative importance. Some of the facilities are assigned for static purposes and others are user definable. It was the user definable facilities that we needed to consider. They are named local0 through local7.

When configuring our devices to use Syslog we often had the option to determine which facility and sometimes which severity the device would use. This provided us with an opportunity to organize the incoming log data in a way that would make it easier to manipulate both in an immediate and an historical context. Here is how we defined the facility values for our network.

Facility Name	Purpose
Local0	Perimeter devices (various devices)
Local1	unassigned
Local2	Interior networked servers
Local3	Peripheral devices (printers, scanners)
Local4	Workstations (for future use)
Local5	unassigned
Local6	unassigned
Local7	Interior routers & switches

Configuring all of our devices according to this scheme lent itself to a number of applications. Applying different Kiwi rules to the different facilities was really helpful. This approach allowed us to easily focus our attention on high-risk devices. Those messages coming from perimeter hosts could be isolated from the rest of the traffic flow for increased scrutiny. I will detail some examples of

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

this approach later in the study but now we will examine how some specific devices were configured.

4.1 Microsoft servers as Syslog clients

Many programs exist for the purpose of forwarding Windows NT and Win2k Event Log messages to a Syslog server. Some are more feature-laden and robust than others. We required no particular features except the ability to specify the Syslog facility that each Microsoft server would use. For this purpose we found BackLog[7] from Intersect Alliance to be very effective.

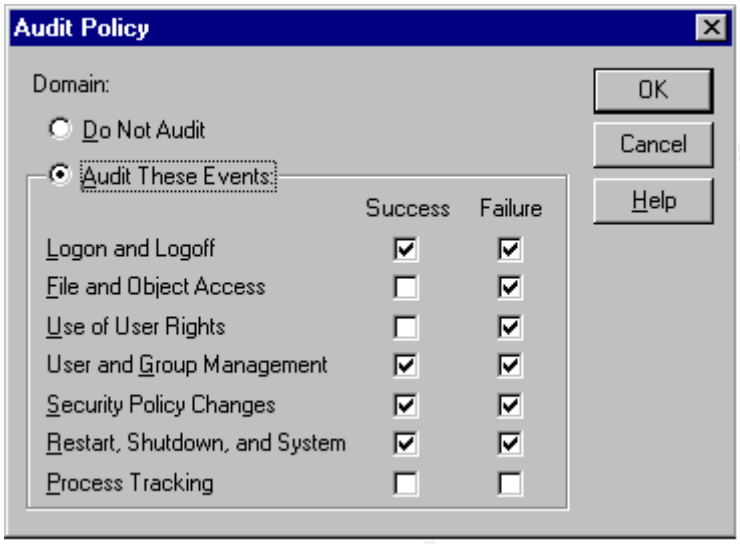
The installation was simple and the only information required to configure it was the IP address of the Syslog server. If desired you could also specify the facility.severity for it to use when logging. In this case we set it to use Local2.notice because it was an interior server. If the server was a perimeter device we would simply have changed the facility to Local0 but all the other settings would have remained the same.

Upon completing the installation we immediately began to see messages arriving on the Syslog server. Now we just needed to complete this installation on a couple hundred more machines.

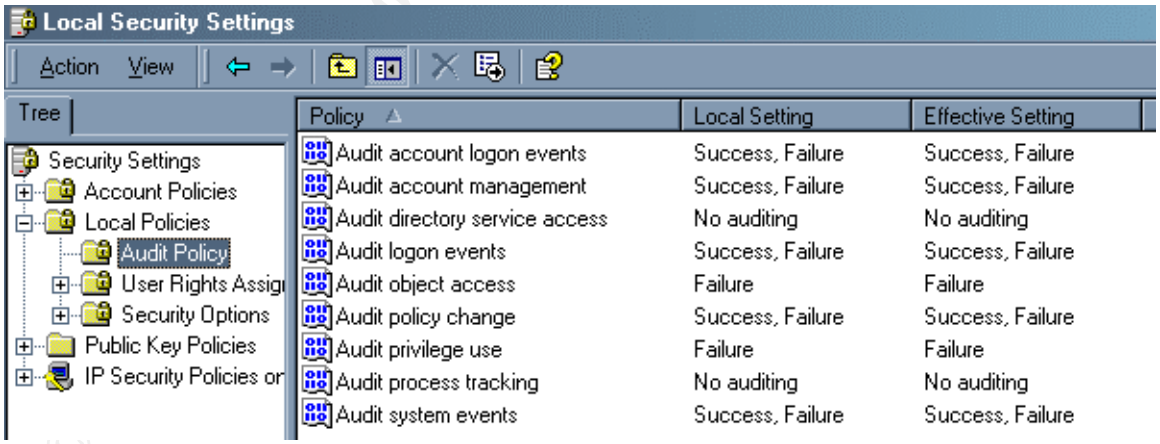
This is probably a good time to discuss auditing policies. BackLog will only forward messages if they get written to the local Event Log in the first place. That means it is important to configure the server to log all the information that you wish to capture. This configuration is done within the operating system itself.

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

On a Windows NT 4.0 machine the Audit Policy settings are located within User Manager. Simply pull down the Policies menu from the toolbar and select Audit. Choose the settings that comply with your company's security policy. Here are the settings that we used:



On a Win2k machine these settings are located in Control Panel / Administrative Tools / Local Security Policy. The Center for Internet Security's Win2k Gold Standard Template sets these parameters for you. Here are the settings that we used:



Once these settings were configured on all the servers and Backlog was installed on each of them we had successfully deployed Syslog to our Microsoft network.

It is recommended that you set your server Event Logs to a large size and to "overwrite as needed". This is because each event gets written to the local Event Log as well as to the Syslog server. Usually it is more

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

convenient for administrators to examine messages in the Event Log instead of having to grant them access to the Syslog data.

4.2 Cisco routers as Syslog clients

Our Cisco routers were easy to configure because they speak Syslog natively. A few simple commands were all that it took to get started. Here are the configuration steps that we followed.

```
Router# config t
Router(config)# logging on
Router(config)# logging 172.16.1.1  (address of syslog host)
Router(config)# logging trap informational
Router(config)# logging facility local7
```

There are additional commands that can be used to fine tune the router but this is a good starting point. It served to get the logging process underway. Most of the commands are intuitive. However, I will take a moment to explain the "logging trap informational" statement.

As we know from looking at the Syslog RFC each message has a priority value comprised of a facility and a severity. In Cisco's implementation we can designate which severity level interests us using the "logging trap x" statement. When we enter a keyword in that statement we will receive all logged message of that severity or lower. So by using the "logging trap informational" statement we are saying that we want to see all messages of the info, notice, warning, err, crit, alert or emerg severities. We are also saying that we do not wish to log debug messages.

There is a certain amount of overhead consumed by the logging process on a router. It is generally minimal but logging debug messages can consume a lot of processing cycles. Use that keyword cautiously.

One further topic of interest on Cisco routers is logged access lists. We need to tell the router what types of items we are interested in logging. One way to do this is with a logged access list.

By adding the keyword "log" to the end of an access list statement we tell the router that any time a packet meets the criteria of that statement that it should log a message. One useful place to employ this function is in a "deny" statement at the end of an access list. For example:

access-list 101 deny ip any any log

would log any IP traffic that was trying to traverse the interface. This would obviously be excessive but it demonstrates proper syntax. How this feature is best applied will be determined by the needs of each network.

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

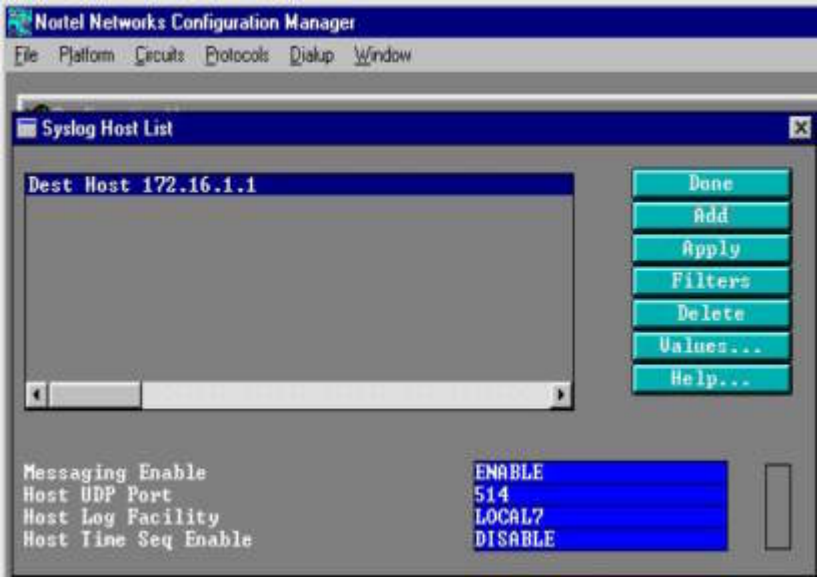
4.3 Bay Networks or Nortel routers as Syslog clients

In our environment we had a homogenous mix of networking gear. Consequently we needed to learn how to enable Syslog on a variety of devices. One of the devices we were confronted with was an old Bay Networks router. Here is a link to some information that we found about configuring Syslog on Bay routers[8].

Sometimes a picture helps to illustrate, and so in an effort to explain how we got this working I will include a few screen captures.

To enable Syslog using Site Manager we opened the router's configuration file dynamically. Then we pulled down the *Platform* menu from the toolbar. On that menu we selected *Syslog* and a submenu emerged with a *Create Syslog* menu option. We chose that option which created a new instance of Syslog on the router. The default parameters did not need to be changed for our application.

Once the Syslog instance was created we needed to pull down the *Platform* menu once again. This time on the *Syslog* submenu several new choices were available. *Syslog Host Table* was where most of the configuring took place. Here is a screen shot:



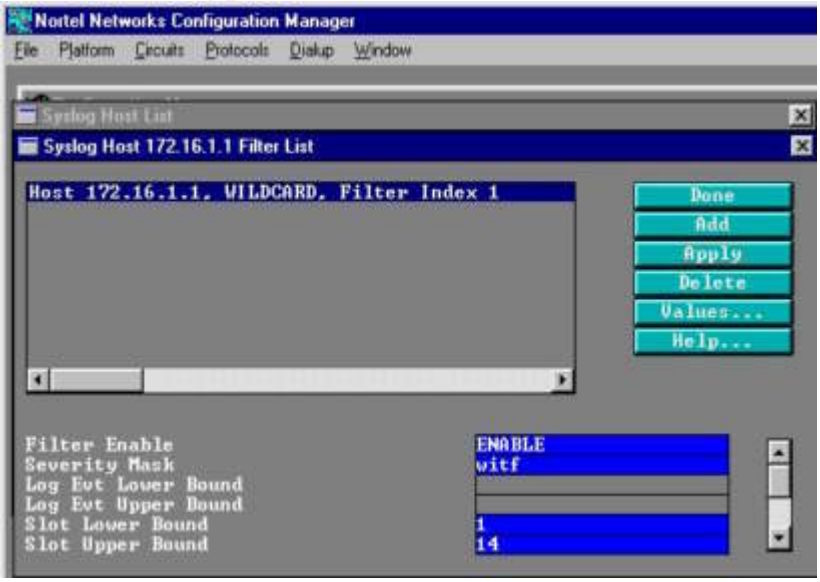
The first step required to configure Syslog was to press the *Add* button and enter the IP Address of our Syslog server. Once we had completed that entry a *Filter List* box popped up. The list was empty to start with so we needed to configure a filter. On the *Filter List* box we hit the *Add* button and were prompted to enter a *Filter Entity Name*. The required

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

value was selected from a predetermined list which was displayed by pressing the *Values* button. Here is a screen shot of that list:

It appeared to be a feature where we could specify which protocols were to use Syslog. In our case we did not want that type of granular control. We desired that all protocols would use Syslog. That requirement was fulfilled by choosing the *WILDCARD* Filter Entity value. Unfortunately that choice was at the very bottom of a long list of values. We scrolled all the way down and selected it.

Once we had saved our filter configuration we could see the host and filter definition listed in the *Filter List* box. Here is a screen shot to demonstrate our progress:



Finally we saw a few remaining parameters that needed to be configured. They were located below the *Filter List* itself. We needed to give them careful consideration because they were not very intuitive.

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

Severity Mask was where we specified which messages we were interested in logging. The options were witfd which stand for the severity values w-arning, i-nformation, t-race, f-ault, d-ebug. We found witf to be sufficient for our needs without placing an undue burden on the router.

The other parameters that needed to be set were the Slot Lower Bound and Slot Upper Bound values. These settings represented the slot numbers in the router. This made more sense when we envisioned a modular router with replaceable cards in it. These values defined which modules we wanted using Syslog. Because we wanted the entire router to use it we entered the lowest slot number and the highest slot number in the router. Since we were unsure how many slots we had (we were working on a remote router) we entered an excessively high number for the Slot Upper Bound value. We actually entered a value of 14 even though there were probably only 4 slots in our router. It accepted the higher number and automatically rounded it down to the actual Upper Slot number in the router. Syslog would not function on the Bay router until the severity mask and the slot values were specified, so it was important to configure those parameters before the settings were saved.

Once we had saved all the Filter settings we found ourselves back at the *Syslog Host List*. The host we had just configured now appeared in the list. In addition an option was given to customize the UDP port for Syslog to use. This would have been helpful if we intended it to use a custom port and not the standard of UDP 514. *Host Log Facility* was where we determined which facility value we wanted the messages stamped with. In this instance it was an interior router so we assigned it to Local7.

After all the parameters were set to our satisfaction we hit the *Done* button to save the entire Syslog configuration. Once we had also saved the router's config file we had successfully enabled Syslog on that device.

4.4 Additional devices

Other devices that can use Syslog include printers and print servers, switches, firewalls, workstations, etc. Rather than detailing them all we will now move on to discuss some concerns regarding management of the collected log data.

5.0 Maintaining log data integrity

Once collected, the log data can be parsed and manipulated in many ways by using a database such as MS Access or SQL. I will avoid a discussion of how this is best accomplished because methods can vary greatly based on the peculiarities of each computing environment.

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

5.1 Time Synchronization

I do want to mention an important consideration that affects all environments equally. That is the problem of synchronizing the time and date on the network. An excellent discussion of this issue can be found in a paper called Forensic impact of Network Time Synchronization by Belinda Brockman[9] in the SANS reading room. I want to consider the benefits that synchronized time can offer when analyzing log data.

If multiple time sources are employed or, worse yet, if each device is allowed to keep time for itself the results can be quite confusing. It will be difficult to correlate log data from multiple devices. When independent time sources exist an accurate chronology of what occurred during an incident can never really be developed. Synchronizing the network time can provide improved analyses of your log files.

We used Network Time Protocol (NTP) to synchronize all of the devices on our network to one time source. Many of the devices we configured included native support for NTP. For those that did not have native support there was a wide variety of utility programs to help. This was a simple part of the project and was really crucial. It is recommended that you consider this issue when designing your deployment.

5.2 Archiving

Another issue that needed to be addressed was how to maintain the integrity of the log data. Procedures for archiving and handling the log files were needed if the information within was to be found trustworthy. Fortunately my SANS training equipped me with the knowledge needed to satisfy this requirement.

We setup the Kiwi Syslog Daemon to create a unique log file for each day. It was also configured to archive its log file every night. One of the archiving features that Kiwi offers is the ability to run an external program after the archiving process has completed. This extensibility was extremely valuable to our design.

We wrote a batch file to be executed by Kiwi that triggers several utility programs. These programs generate an MD5 hash value[10] for the log data, and then write the log data and its MD5 hash to write-once media[11] (CDROM). This batch file gets executed within seconds of the log data being archived. Consequently the integrity of the log data is thoroughly established. As long as its MD5 hash value compares precisely with the value stored on the CDROM we can be confident that the log data has not been tampered with.

Here is some of the pertinent code from the batch file. By presenting this example I hope to show the syntax for using these utilities. As a result I

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

am showing only those code snippets that pertain to the utilities themselves. Remember that directory structures and SCSI ID's can differ from server to server.

```
@echo off
md5.exe %1.txt c:\MD5\%1.MD5
mkisofs -J -V SYSLOG -o Temp.iso c:\Temp
cdrecord -v speed=2 dev=0,6,0 -eject Temp.iso
```

In addition to the nightly process of archiving to write-once media (CDROM) we also established a weekly backup job using Digital Linear Tape. The backup job copies an entire weeks worth of log data and MD5 hashes to tape. The tapes are then stored offsite in a data vault for safe-keeping. This provides redundant storage in case something happens to one of the CDROM discs.

6.0 Interesting Applications

We have found the alerting capabilities of the Kiwi Syslog Daemon very useful. Although we have just begun scratching the surface of potential uses it might be interesting to look at a couple of the successes we have achieved.

As described in section 3.0 the default rule on the Kiwi server applies two actions to every incoming message. One action writes each message to a file and the other displays each message to the screen. It does not take long before the high volume of displayed messages becomes overwhelming. This can cause the display to move too fast to be of any use.

We decided that a better way to display the incoming messages was to sort them based on Syslog facility. Kiwi has ten different user definable displays numbered "Display 00" thru "Display 09". We used this feature to send all the Local0 messages to "Display 00" and all the Local2 messages to "Display 02", etc. Consequently we could focus our attention on high priority devices instead of being overwhelmed by less important traffic. In addition, if we were troubleshooting routers we could choose to watch "Display 07" for a while to assist us in that process. By sorting the messages we slowed down the display rate to a readable pace and improved the pertinence of the displayed data.

Another interesting rule we created was an alert to inform our administrators of Windows Update files on our servers. By configuring a filter that searched for the text string "Automatic Updates" we could locate any message which announced the arrival of a software patch. Then the rule would fire off an email alert telling our administrators of the need to apply that patch. This has really helped to increase the timeliness of our updates.

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

As a performance management tool we occasionally will create a temporary rule to assists us in working on a problematic host. We simply configure a rule that emails all messages coming from the host in question. Then we proceed with our troubleshooting process. The immediate feedback that this generates can be of tremendous assistance in isolating the cause of a particular symptom.

7.0 Conclusion - Post deployment benefits

This project had a swift remedial effect on the issues we had been facing. As we configured more and more hosts to use Syslog the solution increasingly impacted our environment. All of our networked devices could now be monitored regularly and then managed promptly. Although technically we were still being reactive our reaction time was reduced to mere seconds. We were beginning to learn about our performance problems from the hosts themselves instead of from our users, which was a valuable improvement.

All of our design criteria were satisfied by this deployment. Each and every device's log data was now being captured in one central repository. That data was safely stored and archived with a high level of integrity. The ability to correlate log data across multiple devices was accomplished due to the common time source and log format that we established. The near real-time alerting exceeded our expectations and has proven effective in increasing security and reducing downtime.

This is the sort of project that never really ends but will constantly evolve. Even though our initial design has been fully realized there are some further ideas which we intend to explore that might improve this project. Here is a list of features that we hope to include in the future:

- We would like to deploy a redundant Syslog server to protect us from system failure and to achieve high availability.

- We are interested in ensuring that the Syslog services on our hosts function reliably. We are considering using Servers Alive[12] or perhaps Nagios[13] for this purpose.

- We have heard about increasing the security of a "receive only" host by breaking the transmit wire pair on an Ethernet cable. This technique would apply nicely to Syslog and so we intend to consider it.

I am thankful to the SANS community for the wealth of its shared knowledge. I hope the information that is included in this case study will enrich your project.

References

[1] Kiwi Syslog Daemon

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

URL: http://www.kiwisyslog.com/products.htm#syslog

[2] Wilkins, Brian R. "Effective Logging & Use of the Kiwi Syslog Utility"
URL: http://www.sans.org/rr/paper.php?id=201

[3] Request For Comments: 3164 – The BSD Syslog Protocol
URL: http://www.ietf.org/rfc/rfc3164.txt?number=3164

[4] Microsoft Network Security Hotfix Checker (Hfnetchk.exe) Tool
URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;303215

[5] The Center For Internet Security – Win2k Pro Gold Standard Template
URL: http://www.cisecurity.com/bench_win2000.html

[6] Kiwi Syslog Daemon For Windows – Online Help File
URL: http://www.kiwisyslog.com/help/Syslogd/index.html

[7] Backlog – Windows NT Event Redirection – by Intersect Alliance
URL: http://www.intersectalliance.com/projects/BackLogNT/archive.html

[8] Bay Networks – Configuring Syslog on the Router
URL:http://support.baynetworks.com/library/tpubs/html/switches/bstream/115412A/P_38.HTM#MARKER-2-455

[9] Brockman, Belinda "A Forensic Argument for Network Time Synchronization"
URL: http://www.sans.org/rr/legal/time_sync.php

[10] MD5 Command Line Message Digest Utility
URL: http://www.fourmilab.ch/md5/

[11] CDRTools – Command line tools for CDROM burning.
Files: ftp://ftp.berlios.de/pub/cdrecord/alpha/win32/
Documentation:
http://www.fokus.fhg.de/research/cc/glone/employees/joerg.schilling/private/man/

[12] Woodstone: Servers Alive
URL: http://www.woodstone.nu/salive/

[13] Nagios: formerly called NetSaint
URL: http://www.nagios.org/

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

Last Updated: May 25th, 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