Global Information Assurance Certification Paper

Joe Malmberg
February 11, 2004
GSEC Practical, Version 1.4b Option 2

Centralized Logging with Open Source Software in a Unix/Windows Environment

Summary
In day-to-day operation, system logs are arguably the most under-appreciated and overlooked feature of modern operating systems. When the security of your network has been compromised, their importance is quickly adjusted to a more appropriate level. When you find yourself in this situation, the importance of a trusted set of logs becomes paramount. Often, in the case of a break-in, such a log can quickly become your only trusted source of information.

As is the case today, several IT shops are running systems with a mix of Unix and Windows flavors. Powerful and disparate, they make the task of centralized logging a bit trickier than that of a "single OS" shop. This paper will explore a low-cost process of setting up, securing and configuring a centralized logging system and its clients in a "mixed" (read: Unix and Windows) environment. The following processes prove that logging in a mixed environment is feasible and can be done entirely using open source software.

The Problem
The network at my place of employment existed prior to my hiring. It was clear that the network was set up piecemeal with little foresight or attention to detail. It was poorly documented, using an unnecessarily large block of non-routable IPs and the "firewall" was riddled with security holes. After writing network documentation, switching ip blocks and tightening up the firewall, I recognized that I still did not have the ability to quickly and easily check system logs.

While each system on my network provides some kind of logging mechanism, it would be difficult, if not impossible, to sufficiently monitor their logs machine-by-machine. The obvious solution to this problem, after reading the chapters on Windows and Unix security in the SANS course material, is to institute some incarnation of a centralized logging server (here out referred to as CLS).

My Logging Requirements
My network, like most, is made up of many machines running several different operating systems. Specifically, my network has a number of Windows machines, several GNU/Linux boxes, HP Printers and a peppering of Cisco routers. The system that receives my logging messages must be capable of parsing messages from all of these devices.

Once set up, it must be easier to use than any of the existing log analysis tools included by default with the above operating systems. Daily or hourly updates via email are a necessity. A web interface would be an added bonus.

The CLS must offer me several layers of flexibility. I want flexibility in the operating systems and applications that it can and will log from. I want to have the option of adding logs from new operating systems and/or applications my

employer may acquire down the road. I would also prefer one that allows more than one method of log storage (e.g. flat file, database, etc.)

A second requirement of my CLS is that it must be free and/or open source. This "requirement" is more of a preference than a necessity. I am a regular user of several free software projects, many handling mission critical applications for my office. The cost of free software makes upper management happy and being able to examine and/or fix the innards of the software running in my organization makes me happy.

Logging Options

Even with this semi-rigid list of criteria, there are still several packages that fit the bill for this project. First and foremost is syslog. Syslog is a protocol that allows a machine to send event notification messages across IP networks to event message collectors - also known as Syslog Servers or Syslog Daemons.[1] The daemon receives syslog messages and pushes them into log files. It has the ability to collect logs from remote machines and is well supported. It is also a mature project and not likely to see great changes in the near future.

There are syslog daemons for several platforms. The most popular Windows flavors are Winsyslog by Adiscon, Kiwi Syslog Daemon by Kiwi Enterprises and SysLog Turbo by weird solutions. While all of these solutions provide a syslog daemon, none of them quite fit the bill. Winsyslog is free for home use, Kiwi Syslog Daemon is free as long as it is not run as a service and Syslog Turbo has only a free demo. The lack of a syslog daemon for Windows that meets all of my criteria causes me to search for an option that runs on GNU/Linux.

Syslog has been included by default in every GNU/Linux distribution I have used. The daemon is free to download and install. The version distributed with my preference of distributions, Debian, it is offered under the Gnu Public License (GPL) and Berkely licence as sysklogd[2]. This version of syslog is capable of running as a daemon, is open source and costs nothing. It would be the ideal choice if it were not for syslog-ng.

Syslog-ng, or Syslog Next Generation, is an actively developed syslog daemon by BalaBit IT Ltd. and is distributed under the terms of the GPL. It has an active user base and is used by several individuals for their centralized logging. Syslog-ng is syslog on steroids. The major advantage of syslog-ng is in its piping and filtering capabilities. Syslog-ng can filter logs based on conditions the user sets in its configuration file and can output those logs in a multitude of fashions.

---

[1] Rehman, Waji-ur. "Introduction to Syslog Protocol." 25 March 2003.
URL: http://www.monitorware.com/Common/en/Articles/syslog-described.asp (23 October 2003).

[2] "Sysklogd Copyright." 1.4.1-10.
URL: http://people.debian.org/~noel/changelogs/pool/main/s/sysklogd/sysklogd_1.4.1-10/copyright (15 December 2003).

Syslog-ng seems to meet all the criteria I have set forth above. It is flexible enough to take whatever syslog messages are thrown at it and process them. It can output to any format you can define and it has the bonus of customizability. For these reasons, I am choosing syslog-ng as the logging daemon for my CLS.

Prepping for installation: Bastille

I have chosen to install syslog-ng on a machine with a fresh installation of Debian stable/testing GNU/Linux on a PII 350 with 512MB of ram and mirrored 200GB hard drives. Prior to installation, I am running through some processes to harden the OS. First, I am running the Bastille GNU/Linux hardening script. The Bastille hardening system incorporates many operating system hardening recommendations including the SANS guide to securing GNU/Linux.[3]

Installation of the Bastille script is a breeze. In Debian, it is a simple:

```
apt-get install bastille
```

After apt pulls the packages and installs them, I simply type "`bastille`" and follow the prompts. The Bastille script is menu-driven and relatively painless to install. Depending on the options you choose, Bastille can set up a firewall via ipchains, apply system patches, perform an SUID root audit and deactivate or restrict unnecessary services.[4] A full walkthrough of the Bastille script is outside the scope of this paper, but to get started I am patching the system, performing the root audit and deactivating unnecessary services. I opted to forego the firewall until I have the server accepting connections. Only after it is working will I risk breaking it!

Timesync

Since I will be collecting logs from all over the network, it is imperative that I have a reliable time source with which to verify the logs. Currently, all of the machines on the network sync with the time service on GNU/Linux fileserver that I have on my network. The time server is running an NTP (Network Time Protocol) daemon. To sync the clock on my CLS with my network's time server, I am placing the following script into my /etc/cron.daily directory:

Begin timesync

/usr/sbin/ntpdate –s *timeserver*

End timesync

After verifying the script has the correct execute permissions and works, I move on to installing syslog-ng.

---

[3] "Bastille Linux" 19 January 2004. URL: http://www.bastille-linux.org/ (21 January 2004).

[4] Beale, Jay. "Bastille Linux: A Walkthrough." SecurityFocus. 6 June 2000. URL: http://www.securityfocus.com/infocus/1414 (21 January 2004).

File Integrity: AIDE and Samhain
Now that the system is in a "pristine" state, I am going to install a couple of file integrity checking tools. I am choosing to install Samhain for its centralized logging feature and AIDE for redundancy. AIDE is a simple file integrity checker for one machine. Its installation is as straightforward as most in Debian.

An "apt-get install aide" downloads and installs AIDE. The file /etc/aide/aide.conf has to be edited to point to the email address I want my logs sent and define which directories to check for changes. The Debian package adds an AIDE script to /etc/cron.daily/aide. The script should be run once via the command line to populate the AIDE database.

Like AIDE, Samhain is a file integrity checker but a bit more robust. It has the capability to act as a central server, receiving Samhain reports from other Samhain clients. It logs these reports to a MySQL database for future reference. Installation of Samhain as a server on Debian GNU/Linux is not as straightforward as most installations. The program must be compiled from source, as the server option is not compiled into the Debian package. Following the superb tutorial on linuxsecurity.com[5] will install Samhain on the CLS with the server option.

I am opting to compile my own version of Samhain for my GNU/Linux based boxes. The tutorial on linuxsecurity.com outlines Samhain installation on the clients. Since all of my GNU/Linux boxes run Debian, I am going to create a package so I do not have to recompile it for each system. To accomplish this, I configure the package the same way as described on the linuxsecurity.com tutorial, but compile it with a "make deb". This will create a Debian package that I can install on my other machines. I will still have to create a new Samhain executable for each Debian machine via "samhain_setpwd", but the Debian package will still provide for a much faster installation.

After following the rest of the tutorial on linuxsecurity.com and verifying my database is populating, I can move on to the installation of syslog-ng.

Installing syslog-ng on the server
Again, Debian proves its greatest strength with the installation of Syslog-ng. A simple "`apt-get install syslog-ng`" pulls the necessary syslog-ng package and removes Debian's sysklogd. Syslog-ng is now installed. It requires some fine tuning, but, before starting with that, I would like to look at some of the additional options Syslog-ng provides.

[5] Dunston, Duane. "Centralized File-Integrity With Samhain Part I." Linux Security. 9 August 2002. URL: http://www.linuxsecurity.com/feature_stories/feature_story-116.html (11 November 2003)

According to the excellent Syslog-ng faq[6], there has already been work done on pushing acquired logs to a MySQL database. Since this is one of my criteria, I am planning on implementing it. The faq also has links to a "mini-howto" on creating a CLS with Syslog-ng and a long list of valuable Syslog-ng resources.

Configuring syslog-ng on the server

The entire configuration for syslog-ng is done in the file syslog-ng.conf. In Debian, this file is located under /etc/syslog-ng/syslog-ng.conf. The default configuration file is very different from what I want my setup to look like, so instead of editing the existing file, I am starting from scratch. There is a fantastic sample syslog-ng.conf file located on the same server as the syslog-ng faq[7]. The sample is loaded with comments and documentation on all of the parameters and options a syslog-ng configuration file can contain. I have found it to be more valuable than the included documentation.

For the sake of clarity, I am breaking up my syslog-ng.conf into pertinent sections and explaining their relevance.

```
####################
# Set log sources #
####################

source src
  {
    unix-dgram("/dev/log");
    internal();
    udp();
  };
```

The above snippet defines the input sources from which syslog-ng will process logs. Log sources are defined as internal logs ("unix-dgram" and "internal") and logs coming via the syslog port, UDP 514.

```
####################
# Set Destinations #
####################

# Organizes logs by system name and date
#
  destination hosts {

file("/var/log/HOSTS/$HOST/$YEAR/$MONTH/$DAY/$FACILITY$YEAR$MONTH$
DAY"
   owner(root) group(root) perm(0600) dir_perm(0700)
create_dirs(yes));
  };

  log {
        source(src);
        destination(hosts);  };
```

[6] Campi, Nate. "Syslog-ng FAQ." Campin dot Net.
URL: http://www.campin.net/syslog-ng/faq.html (9 January 2004).
[7] Campi, Nate and Szalay Atilla. "Sample syslog-ng.conf."
URL: http://www.campin.net/syslog-ng.conf (9 January 2004).

The above section of the configuration is taken directly from the syslog-ng *Central Loghost Mini-Howto*[8]. It drops logs coming from other machines into directories sorted by hostname and date. The resulting directory structures are easier to navigate and search via grep.

```
## Log syslog-ng to mysql database
##
    destination d_mysql {
        pipe("/tmp/mysql.pipe"
        template("INSERT INTO logs (host, facility, priority,
level, tag, date,
        time, program, msg) VALUES ( '$HOST', '$FACILITY',
'$PRIORITY', '$LEVEL', '$TAG',
        '$YEAR-$MONTH-$DAY', '$HOUR:$MIN:$SEC', '$PROGRAM', '$MSG'
);\n") template-escape(yes));
        };
    log {
          source(src);
          destination(d_mysql);
    };
```

One of my requirements for a CLS is the ability to log to a database. The above snippet from vermeer.org pipes syslog data to a MySQL database that I will define in the next section.

The remainder of the configuration file defines what logs to watch, how to process them and where to pipe the output. I want to take advantage of the fact that the CLS and its monitor are located in my office, so I will push all emergency log messages to the console. First, I define the console:

```
destination console_all { file("/dev/tty8"); };
```

Then I ask syslog-ng to push all emergency logs to the defined console:

```
log { source(src); filter(f_emergency); destination(console_all); };
```

The net result is that, should any emergency logs come through, they will appear on the screen of the CLS. This can come in very handy in quickly diagnosing errors when problems arise on the network.

LogRotate

Since I have changed the syslog-ng configuration file so drastically and have new sets of logs being created daily, I need to reconfigure logrotate to process the new logs. I simply edit the file `/etc/logrotate.d/syslog-ng` and add the logs that I want logrotate to process. After saving the file, there is no service to restart, as logrotate is run as a cron job every night.

[8] Campi, Nate. "Central Loghost Mini-HOWTO." Campin dot Net. URL: http://www.campin.net/newlogcheck.html (9 January 2004).

MySQL

I like GUIs for some things. The program, php-syslog-ng at vemeer.org[9] provides a web GUI for the syslog-ng logs that have been pushed to a MySQL database. To run, the program requires MySQL and Apache. I am not a master of MySQL via the commandline, so I am also going to install the nifty MySQL GUI, PHPMyAdmin. The command:

```
apt-get install apache-ssl phpmyadmin mysql-server
```

will install all three of those packages, as well as their dependencies.

php-syslog-ng also requires php. The command: "`apt-get install php4`" will install php4 and make the necessary modifications to the apache-ssl httpd.conf.

Once the programs are installed, the MySQL database I defined in the previous section must be created. Fortunately, the authors of php-syslog-ng have included a script for creating the MySQL table structure in their installation instructions at http://vermeer.org/syslog/. After creating the table structure, I need to create a fifo pipe for the logs to run through. As defined in the MySQL portion of syslog-ng.conf, MySQL will be waiting for logs from /tmp/mysql.pipe. To create this pipe, simply type "`mkfifo /tmp/mysql.pipe`" and restart syslog-ng with /etc/init.d/syslog-ng restart.

To make sure that the fifo is initialized each time the computer is restarted, I make the following script from vermeer.org run at startup:

```
#
# Created by Tadghe Patrick Danu
#


#!/bin/bash

if [ -e /tmp/mysql.pipe ]; then
        while [ -e /tmp/mysql.pipe ]
                do
                        mysql -u mysqluser --password=password syslog <
/tmp/mysql.pipe
        done
else
        mkfifo /tmp/mysql.pipe
fi
```

I save it as `php-syslog-check.sh` in `/etc/init.d/` and make it executable with

```
chmod 755 /etc/init.d/php-syslog-check.sh
```

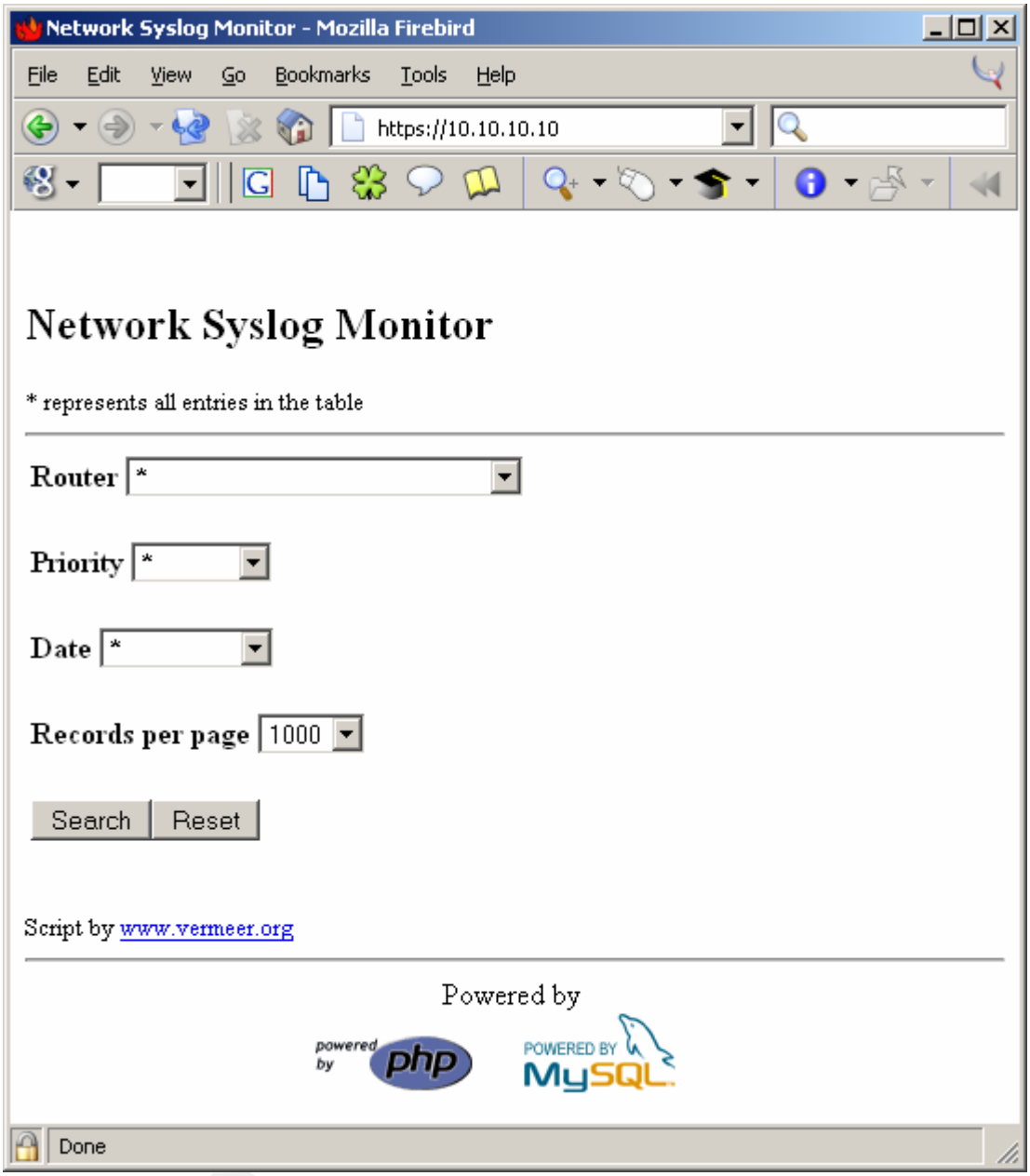
[9] "Centralized Syslog-ng to MySQL Database." 18 May 2002. URL: http://vermeer.org/syslog/ (10 January 2004).

I then create a link to the script in the appropriate run level directory. My GNU/Linux machine starts at run level 2.

```
ln -s /etc/init.d/php-syslog-check.sh /etc/rc2.d/S21phpsyslog
```

This will help to ensure no logs are dropped from the MySQL logs database at reboot.

The final steps of php-syslog-ng installation involve editing and copying the web files from the php-syslog-ng-1.4.tgz archive to the apache web server. In the directory /php-syslog-ng-1.4/web, I edit the file dbinfo.inc.php to point the php scripts to the correct MySQL database. I then copy the files to the root folder of the web server, /var/www by default. Entering the ip address of the CLS into a browser gives me the following page:



I will test this out more thoroughly in the "Viewing the logs" section below, after the CLS has begun receiving logs from its clients.

Installation and configuration of syslog-ng on the clients

All of the GNU/Linux machines on my network run Debian. As shown in the server installation section, installation of syslog-ng is a breeze. A simple "apt-get install syslog-ng" installs syslog-ng and removes sysklogd. After this, the only part of installation remaining is editing syslog-ng.conf.

Since I have already figured out what I want to log in the syslog-ng.conf on my CLS, I will use that configuration file to configure my clients. It will be necessary

to change a couple of sections to get it to push logs to the CLS. Again, as the configuration file is quite lengthy, I will analyze appropriate snippets of the configuration file.

First, the log sources section should be modified:

```
source src
  {
    unix-dgram("/dev/log");
    internal();
  };
```

Since I am only retrieving local logs, I am removing UDP as a source for logs. I also am removing the sections that push log data to a MySQL database, and the section that organizes system logs into directories.

```
# Send to logserver
destination loghost       { udp("10.10.10.10" port(514)); };
```

In addition to /var/log, the syslog-ng client will be sending its logs to the CLS. In the example above, I define the loghost as the ip or DNS name of the CLS.

After making these changes and saving the file, I restart syslog-ng via "/etc/init.d/syslog-ng restart". I return to the CLS and check the /var/log/HOSTS/ directory. There should be a new directory created with the name or IP of the client I just configured. Navigating the directory structure for the December 25 2003 logs of a machine named "DanielSon" would look something like this:

```
CLS# ls –al /var/log/HOSTS/DanielSon/2003/12/25/
total 296
-rw-------    1 root     root        50209 Dec 25 23:55 auth20031225
-rw-------    1 root     root           59 Dec 25 06:25 authpriv20031225
-rw-------    1 root     root        17539 Dec 25 23:53 cron20031225
-rw-------    1 root     root       204608 Dec 25 23:59 mail20031225
-rw-------    1 root     root         9360 Dec 25 23:57 syslog20031225
```

The directory structure will repeat itself for every day of the year.

A Syslog client for Windows

By default, Windows pushes all system logs to its "Event Viewer", a format incompatible with syslog. To get Event Viewer logs (Event Logs) to a syslog server, they must be converted to syslog format. Fortunately, the need for centralized logging via syslog has existed long enough for several Event Log to syslog translators to arrive on the scene.

Loganalysis.org, a library of log analysis information, lists a number of Event Log to syslog translators.[10] Of the seven or so listed on the loganalysis page, only NTSyslog, evlogsys.pl and Snare were fully functional and distributed under the

[10] Bird, Tina and Marcus Ranum. "syslog Client Configs for Windows/Non-UNIX." Loganalysis.org. URL: http://www.loganalysis.org/sections/syslog/windows-to-syslog/index.html (10 January 2004).

terms of the GPL. Evlogsys.pl is a perl script that is run as a scheduled task. It parses existing Event Logs into syslog format. I do not like the fact that it does not push event logs in real-time. Both NTSyslog and Snare claim to push Event Logs in real-time. I understand that NTSyslog has a fairly large userbase, but I was unable to get it to work as advertised. I also could not find an organized mailing list or user forum. Enter Snare for Windows.

The sticking point of this installation could have been Windows were it not for Snare for Windows. Snare is freely distributed under the terms of the GPL by its creator, Intersect Alliance[11]. Rather than storing and then pushing event logs to a syslog server, Snare for Windows converts and transmits event logs in real-time to a defined syslog server. Snare is actively developed and maintains an active discussion forum on Sourceforge[12].

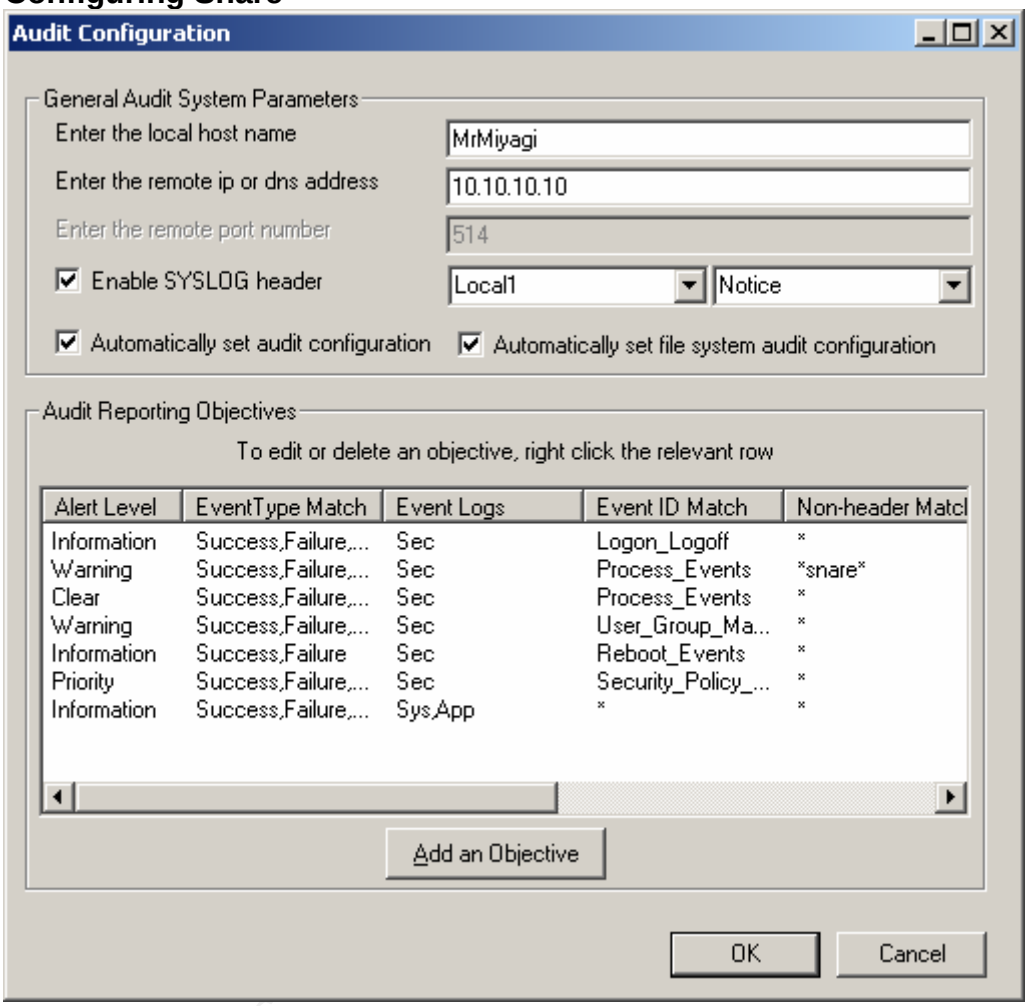
Installing Snare

Installation of Snare is a snap. After downloading the latest version (2.2 as of this writing) and loading the executable, you are presented with a standard Windows-fare installation wizard.

[11] "Snare Agent for Windows." October 2003.
URL: http://www.intersectalliance.com/projects/BackLogNT/index.html (15 January 2004).
[12] "Snare-Users Forum." Sourceforge.net. 22 January 2004.
URL: http://sourceforge.net/forum/forum.php?forum_id=134533 (22 January 2004).

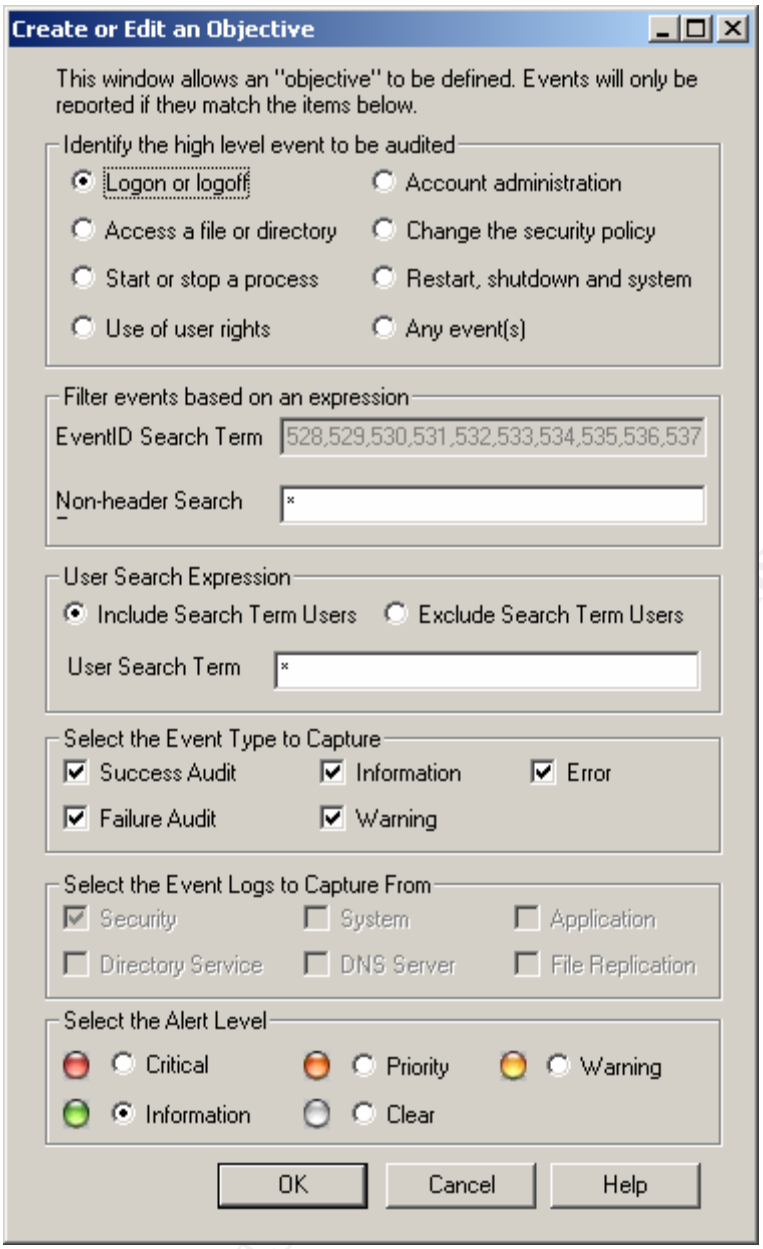
The final step of the installation starts up the Snare service and asks if you want Snare to take control of your EventLog configuration. Saying yes to this allows you to configure more of your logging options from within Snare. You can turn it off later from within the program if you choose.

Configuring Snare



The Snare configuration menu is pretty straightforward. It allows you to modify the hostname of your computer or simply allow your CLS to identify it via DNS. The remote ip is the address of the CLS. When the "Enable SYSLOG header" is checked, Snare will append a syslog prefix to the Windows logs before they are fired off to the CLS. The disadvantage to this Syslog header is that in the example above, all logs, regardless of their priority, will be prefixed as "Notice". I have not found a way around this. I have all Windows logs going to Local1 and have defined a rule in syslog-ng.conf to push all those logs to /var/log/winnt.log. I have also added a rule to logrotate.conf to process this new log file.

Under the "Audit Reporting Objectives" you are given the opportunity to define which EventLog alerts will be forwarded on to the CLS. Right-clicking on an objective and clicking "Edit Objective" will open up the following window:



Snare gives you the opportunity to set the kind of event logged and to filter it from within snare. The predefined high level events have done a lot of research work for you, by defining the appropriate Windows EventIDs to their relevant high level events. You are given more filtering options further down. Since my CLS will be handling filtering, I am not going to filter this rule down more than it already is. From here, simply clicking on “OK” twice and restarting the service will begin forwarding logs to your CLS.

Syslog clients for other Operating Systems

Now that I have logs flowing from both my GNU/Linux machines as well as my Windows machines, I am going to complete the circle by pointing the remainder of my mission-critical log-capable machines to my CLS

Cisco Router

My Cisco routers can be set to log to my CLS by adding the following lines to their running configuration[13]:

```
logging 10.10.10.10 #CLS IP Address
logging on
```

HP Printer

All of the network-enabled HP printers in my company's offices can also send their logs to a centralized server. This option can be set via their network configuration page by setting "Syslog Server" to the IP address of the CLS.

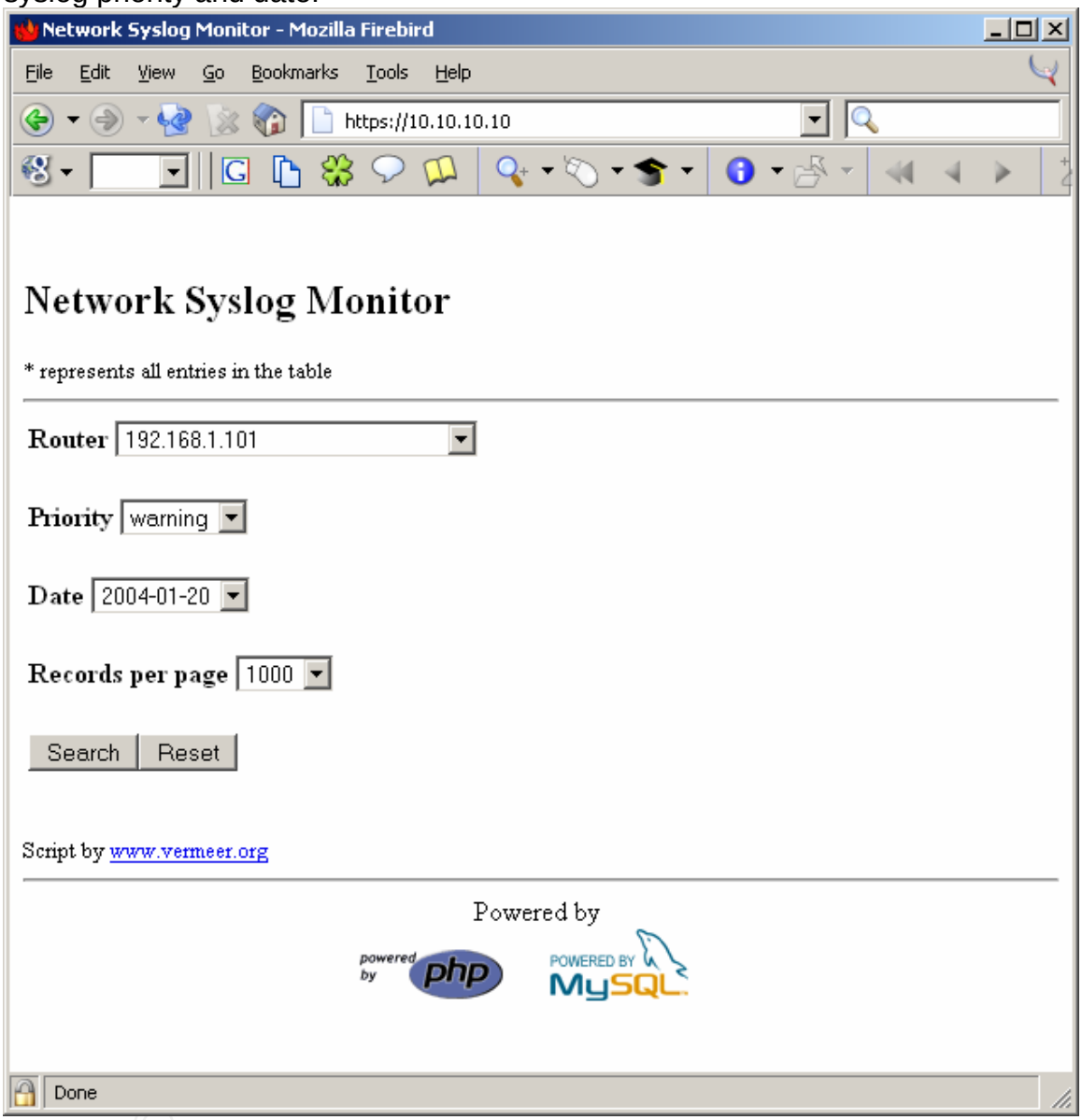
After adding these lines, the routers' and printers' informational logs begin populating their appropriate directories on the CLS.

this space intentionally left blank

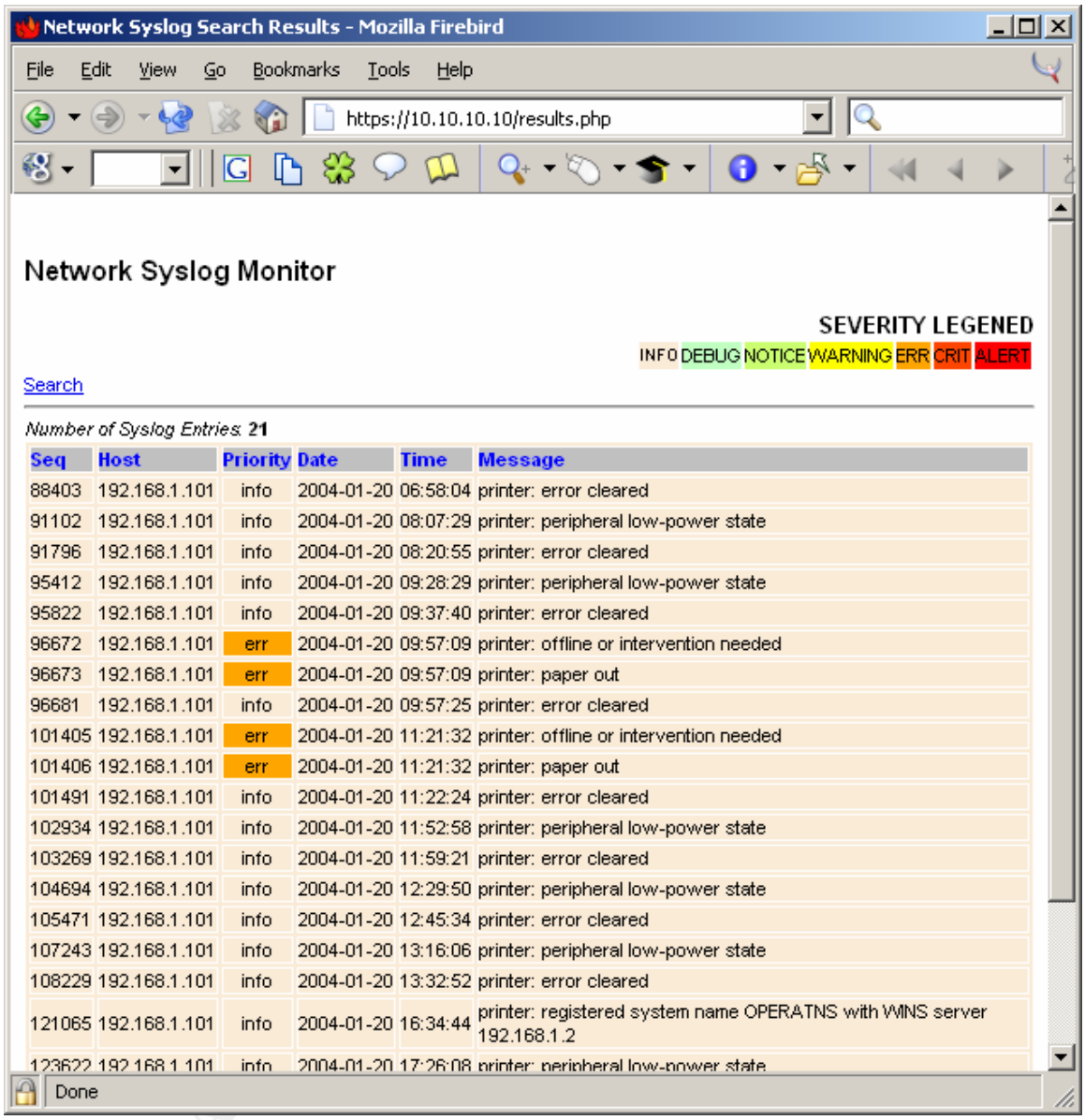
[13] "Cisco Debug Commands" Cisco Systems. 1997.
URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios11/dbook/dintro.htm (3 February 2004).

Viewing the logs

Now that the CLS is receiving and storing logs, how can I make the most of them? The php-syslog-ng network syslog monitor is a good GUI to the log database. The drop-down menus allow you to sort log information by machine, syslog priority and date.



A sample result of logs from a printer on my network looks like this:



A second, less colorful way to view filtered archived logs is via command line with the command “grep”. In this example, I want to see how many times Windows media player was run from the machine MrMiyagi on January 25, 2004. I issue the following command:

```
grep wmplayer.exe /var/log/HOSTS/MrMiyagi/2004/01/25/*
```

Php-syslog-ng and grep are powerful tool for processing logs; however, they do have their drawbacks. In order to view the logs you must visit a web page or

issue a command on the terminal. They are powerful tools for forensics, but inappropriate for daily log checks. This is where newlogcheck comes in.

Newlogcheck

Logcheck was created by Psionic's Craig Rowland as a means to view a digest of system logs in an easy-to-read format. Psionic was purchased by Cisco in October of 2002.[14] Since then, the project has been modified and packaged for several GNU/Linux distributions. The version for Debian GNU/Linux can be acquired via: `apt-get install logcheck`. The most unique program included with logcheck is "logtail". Logtail remembers the last position it read in a log file[15], so the next time it reads the same file, it will begin where it last left off. This is exceptionally useful in a situation where daily log reports are desired.

Newlogcheck is a version of the program logcheck that has been modified by Nathan Campi, maintainer of the syslog-ng FAQ, to work with syslog-ng logs coming from multiple sources. It requires that the original version of logcheck be installed, specifically, the program "logtail". Newlogcheck can be downloaded from Nate's newlogcheck page at http://www.campin.net/newlogcheck.html. Newlogcheck needs to be configured to parse logs for your server. For the sake of clarity, I will highlight the sections I have modified.

This is the first and most crucial change:

```
SYSADMIN=my@email.address.org
```

This needs to be changed to the email address you want your log digests sent. By default it is set to Nate Campi's. (I wonder how many logs from other people Nate has gotten as a result of defaulting SYSADMIN to his email address?)

```
# make sure this is where logtail lives
LOGTAIL=/usr/sbin/logtail
TMPDIR=/var/tmp/logcheck/tmp
GREP=egrep
MAIL=mail
HACKING_FILE=/etc/logcheck/logcheck.cracking
VIOLATIONS_FILE=/etc/logcheck/logcheck.violations
VIOLATIONS_IGNORE_FILE=/etc/logcheck/logcheck.violations.ignore
IGNORE_FILE=/etc/logcheck/logcheck.ignore
```

The Debian logcheck package places all of the definition files in `/etc/logcheck`. The "hacking" and "violations" files contain lists of keywords that newsyslog will search for in the log files. The violations.ignore file and ignore files are lists of keywords that logcheck should ignore when processing reports. Any matches in the "hacking" file will make the subject of your log digest "ACTIVE SYSTEM ATTACK!"

---

[14] Hochmuth, Phil. "Cisco buys Psionic Software." Network World. 22 October 2002. URL: http://www.nwfusion.com/news/2002/1022psi.html (3 February 2004).

[15] "Project: logcheck: Summary." 1 February 2004. URL: http://sourceforge.net/projects/logcheck (3 February 2004).

For this configuration file, the "TMPDIR" directory /var/tmp/logcheck/tmp must also exist. The remainder of the file can be left untouched. Then I simply copy newlogcheck.sh and sort_logs.pl to /usr/sbin and run it via /usr/sbin/newlogcheck.sh.

About twenty seconds later I have a massive log digest in my inbox. Using this file, I can begin to pare down the types of alerts I want sent to me by editing definition files. The GNU/Linux alerts and ignores files are pretty well covered with the default definition files. Definitions for Windows eventlog alerts are completely unwritten in the Debian logcheck definition files. The process of writing Windows eventlog rules is a bit out of the scope of this paper, but here are some pointers that helped me while writing mine. Securityfocus has a good list of "Events to look for"[16]. The breadth and scope of these rules depends on the environment that is being logged. In a smaller environment it may be worthwhile to know when a user enters a bad password. In a larger environment, like mine, this option makes my digest so long it ceases being useful.

After editing the definition files to exclude insignificant events, my newlogcheck syslog digest looks something like this:

```
BEGIN REPORT

-------------------------------
Syslog Report on MrMiyagi

Nothing of interest to report on MrMiyagi

-------------------------------
Syslog Report on src@DanielSon

Nothing of interest to report on src@DanielSon

-------------------------------
Syslog Report on src@JohnnyLawrence

Security Violations
=-=-=-=-=-=-=-=-=-=-=

Jan 30 - 2 times(s): src@JohnnyLawrence dccproc: continue not asking DCC
5 seconds after failure Jan 30 - 2 times(s): src@JohnnyLawrence dccproc:
continue not asking DCC 6 seconds after failure Jan 30 - 3 times(s):
src@JohnnyLawrence /USR/SBIN/CRON: (user) CMD ((/home/user/bin/process-
spam.sh)) Jan 30 - 4 times(s): src@JohnnyLawrence dccproc: continue not
asking DCC 8 seconds after failure

Unusual System Events
=-=-=-=-=-=-=-=-=-=-=

Jan 30 - 2 times(s): src@JohnnyLawrence dccproc: continue not asking DCC
5 seconds after failure Jan 30 - 2 times(s): src@JohnnyLawrence dccproc:
```

[16] Scott, Cory L."Dealing with Windows NT Event Logs, Part Two." Securityfocus. 1 May 2000. URL: http://www.securityfocus.com/infocus/1335 (6 February 2004).

```
continue not asking DCC 6 seconds after failure Jan 30 - 3 times(s):
src@JohnnyLawrence /USR/SBIN/CRON: (user) CMD ((/home/user/bin/process-
spam.sh)) Jan 30 - 4 times(s): src@JohnnyLawrence dccproc: continue not
asking DCC 8 seconds after failure
------------------------------
Syslog Report on src@JoeEsposito

Nothing of interest to report on src@JoeEsposito

------------------------------
Syslog Report on CobraKai

Security Violations
=-=-=-=-=-=-=-=-=-=

Jan 30 - 2 times(s): CobraKai MSWinEventLog     1     Application 48892
      Fri Jan 30 13:08:04 2004      213   LicenseService    N/A   N/A
      Warning     WFSAS1      None  0000: 17 00 02 c0 00 00 00 00
........    Replication of license information failed because the
License Logging Service on server JohnKreese could not be contacted.
      2397

Unusual System Events
=-=-=-=-=-=-=-=-=-=-=

Jan 30 - 2 times(s): CobraKai MSWinEventLog     1     Application 48892
      Fri Jan 30 13:08:04 2004      213   LicenseService    N/A   N/A
      Warning     WFSAS1      None  0000: 17 00 02 c0 00 00 00 00
........    Replication of license information failed because the
License Logging Service on server JohnKreese could not be contacted.
      2397
END OF REPORT
```

This is all useful information. There is a broken script in a user's directory on JohnnyLawrence and CobraKai is unable to contact the Windows license server on JohnKreese. I will have to look into both of these issues.

The final step of the newsyslog.sh process is to set it up to run at a predetermined interval. I have it running hourly on my log server at the times I am most likely going to be there to view it. To accomplish this, I create a cron job via the command: `crontab –e`, then add the following line:

```
0 8-18 * * 1-5 /usr/sbin/newlogcheck.sh
```

This will run newlogcheck.sh every hour on the hour from 8am to 6pm, Monday through Friday.

What is next

Now that the CLS is up and running, there is a short todo/wishlist.
What good do a load of logs do if the drive they live on fails or gets reformatted? I will be setting up rsync to replicate the logs to a large file server in the office. The file server is backed up every night via tape for offsite storage. A dedicated tape backup is also on order for the CLS.

Another machine in my office is running the open source IDS software, Snort. I would like to configure that machine to pump its logs to the CLS as well. It will make checking logs extremely handy if I can pull them all from a single machine.

I would like to explore the possibilities for parsing web logs. Intersect Alliance, the writers of Snare, have created a version for IIS that I would like to try out. By applying certain modules there are syslog options for Apache. I would like to explore those options to create a more robust CLS. Finally, I will continue to fine-tune the logcheck definition files for new threats and not-so-important system messages.

Conclusion

Creating a centralized logging server using entirely open source software is within the grasp of anybody with intermediate knowledge of GNU/Linux. Aside from the low software and hardware costs, the added redundancy and security the server offers fills a necessity of any networked business. The simplicity of the web-based GUI tools ensures non-tech types can view and make sense of system logs without requiring extensive training. In addition, the process of customizing this system educates the installer about the practice of logging more than any off-the-shelf system can.

List of References

1. Rehman, Waji-ur. “Introduction to Syslog Protocol.” 25 March 2003. URL: http://www.monitorware.com/Common/en/Articles/syslog-described.asp (23 October 2003).

2. “Sysklogd Copyright.” 1.4.1-10. URL: http://people.debian.org/~noel/changelogs/pool/main/s/sysklogd/sysklogd_1.4.1-10/copyright (15 December 2003).

3. “Bastille Linux” 19 January 2004. URL: http://www.bastille-linux.org/ (21 January 2004).

4. Beale, Jay. “Bastille Linux: A Walkthrough.” SecurityFocus. 6 June 2000. URL: http://www.securityfocus.com/infocus/1414 (21 January 2004).

5. Dunston, Duane. “Centralized File-Integrity With Samhain Part I.” Linux Security. 9 August 2002. URL: http://www.linuxsecurity.com/feature_stories/feature_story-116.html (11 November 2003)

6. Campi, Nate. “Syslog-ng FAQ.” Campin dot Net. URL: http://www.campin.net/syslog-ng/faq.html (9 January 2004).

7. Campi, Nate and Szalay Atilla. “Sample syslog-ng.conf.” URL: http://www.campin.net/syslog-ng.conf (9 January 2004).

8. Campi, Nate. “Central Loghost Mini-HOWTO.” Campin dot Net. URL: http://www.campin.net/newlogcheck.html (9 January 2004).

9. “Centralized Syslog-ng to MySQL Database.” 18 May 2002. URL: http://vermeer.org/syslog/ (10 January 2004).

10. Bird, Tina and Marcus Ranum. “syslog Client Configs for Windows/Non-UNIX.” Loganalysis.org. URL: http://www.loganalysis.org/sections/syslog/windows-to-syslog/index.html (10 January 2004).

11. “Snare Agent for Windows.” October 2003. URL: http://www.intersectalliance.com/projects/BackLogNT/index.html (15 January 2004).

12. “Snare-Users Forum.” Sourceforge.net. 22 January 2004. URL: http://sourceforge.net/forum/forum.php?forum_id=134533 (22 January 2004).

13. “Cisco Debug Commands” Cisco Systems. 1997.
URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios11/dbook/dintro.htm (3 February 2004).

14. Hochmuth, Phil. “Cisco buys Psionic Software.” Network World. 22 October 2002.
URL: http://www.nwfusion.com/news/2002/1022psi.html (3 February 2004).

15. “Project: logcheck: Summary.” 1 February 2004.
URL: http://sourceforge.net/projects/logcheck (3 February 2004).

16. Scott, Cory L.“Dealing with Windows NT Event Logs, Part Two.” Securityfocus. 1 May 2000.
URL: http://www.securityfocus.com/infocus/1335 (6 February 2004).