Global Information Assurance Certification Paper

Creating A Secure Linux Logging System

GIAC Security Essentials Certification (GSEC)
Practical Assignment
Version 1.4c – Option 1

October 7, 2004

Submitted by: Nathaniel Hall
Location: Overland Park, Kansas

Table of Contents

Introduction

The purpose of this paper is to identify and demonstrate methods that can be used to create a secure Linux logging system that can be expanded to other types of systems for secure logging. Using logs, data can be collected to figure out why a server crashed. If the server is unrecoverable, remote logs allow you to be able to see what happened prior to the crash, even without the system running. If the crash was related to an intrusion, any information that can describe how the system was compromised can help determine the cause of the problem.

After reading this paper, the reader should have a basic understanding of how to prevent intrusions of the logging server and detect them if they occur. I will explain methods to prevent unauthorized logins to administrative accounts, control which clients are allowed to remotely log to the server, and prevent and detect intrusions to the log server. In the demonstrations, various types of free software will be used, all of which are included with the version of Linux to be installed. This ensures compatibility and ease of installation of all needed software.

How Secure Should The Remote Server Be?

Since this remote log server will hold every logged event for all equipment directed to use the server, the remote log server should be very secure of all the servers on a network. Once a server has been compromised, most hackers are able to cover their tracks my removing log file entries on the server they have hacked. This makes it almost impossible to trace how the system was compromised, creating the need for remotely placed log files. To prevent such an intrusion, the remote server will have its own firewall, password security, host-based intrusion detection, and software security, all of which will be configured in this example.

Necessary Software

Throughout this project, several pieces of software will be used. For simplicity, we will be using software included with RedHat Linux Version 9[1]. Some of the software being discussed will be Syslog, Tripwire[2], and IPTables[3]. All software referred to is assumed to be installed, but not configured.

[1] Information on current and legacy RedHat products can be found at www.redhat.com and www.fedoralegacy.org.
[2] Information on open-source Tripwire can be found at www.tripwire.org.
[3] Information on IPTables can be found at www.netfilter.org.

Some Basic Security Concepts

A basic piece of security we will use is password protection. Almost anywhere a password is required, security is emphasized. Most systems will require a minimum password length and a combination of characters, including upper- and lower-case letters, numbers, and sometimes, symbols.

In the case of the secure logging server, we want to create passwords that are almost impossible to guess. These passwords should be a combination of the above characters, a minimum length of 8 characters and should not contain any dictionary words. A minimum of 5 of these passwords should be created.

The next basic security concept to be used in our secure logging server is limiting what is installed. Many pieces of software have unknown security flaws within the code. If these are found, our "secure" logging server could be easily compromised. To prevent this, a very basic install will be used. With less software, there are fewer security holes.

The last basic security concept that will be assumed is perimeter security. Is the network around the logging server secure? If it is not, hacking of the logging server becomes much easier. If a hacker is able to directly connect to your logging server, there is not enough security. We will, however, incorporate some firewall security into the logging server.

Initial Setup

Gather Needed Information

Before the secure logging server can be setup, some information must be gathered. The IP address of the new server must be unique and preferably a private address, for example a 10, 172.16, or 192.168 network address. This will help lower the chances of a hacker gaining access to your internal server. This server should not have any means to connect to or be controlled from an outside network, or the Internet.

Plan Your Partition Layout

Since all information contained within this server must be kept from unauthorized access, all data on the server must be contained locally on the server and not transferred across the network for storage. Due to this, depending on the amount of logs generated, a large amount of disk space may be required.

Knowing this, the partition layout should be designed to accommodate for the large size and number of log files. Plan your partition layout to allow for enough space on the root partition for your install, adequate swap space for the amount of RAM in the server, a small, but appropriately sized /boot and leave the rest to the /var partition. A useful partition layout may look like the following:

/boot	100 MB
/	10 GB
/usr	10 GB
/var	Minimum of 20 GB

This configuration allows for enough disk space to install the system. The /var partition should be as large as possible to prevent filling the partition. Once a partition is completely filled, the server could lose log entries or even crash.

Begin Installation

Begin your install and follow the on screen prompts while selecting the best options for your situation. Setup the partitions as decided above. Select High Security Level when asked about enabling the built-in firewall.

When asked about a password for the root account, enter in one of the secure passwords created earlier. This will help prevent hackers from controling the system.

As stated earlier, a minimum amount of software should be installed on the secure logging system. Specifics should be selected by choosing Select Individual Packages. From this area, everything should be de-selected, except for the few needed packages, such as Tripwire, ntp, syslog, and ViM. Other packages that might be useful would include curl, ethereal, iptraf and screen. Once all packages are selected, make sure all dependencies are fulfilled. This will ensure each software package has all required software pieces.

Finish the installation and reboot when needed. Once the server is running, verify the system properly runs and has a working Internet connection.

Limiting Connections To The Secure Server

After initial installation of the server, the IPTables firewall will be setup. A firewall is “a system designed to prevent unauthorized access to or from a private network.”[4] We will use this to only allow access to the secure logging server from specified logging clients and allowed SSH clients. If help is needed to write

[4] http://www.webopedia.com/TERM/f/firewall.html.

IPTables firewall rules, please refer to Chapter 9 of *Red Hat Linux Firewalls* by Bill McCarty[5] or to Appendix A.

To secure the server using IPTables, write rules to limit traffic on UDP port 514 with source addresses of the client syslog servers. I recommend creating a syslog chain that accepts traffic destined for UDP port 514. Send traffic from approved source IP address to this chain. This will lower the chance of incorrectly written rules causing failed iptables startups.

An additional way of securing the server involves securing SSH using IPTables. First, limit connections to SSH by only accepting SSH from specified source IP addresses, similarly to the way syslog was secured. Create an SSH chain that will filter information destined for TCP port 22. Send approved source IP addresses to this chain.

To obscure the appearance of traffic destined to the secure server, create a prerouting rule to perform DNAT from a high port (i.e. 17216) to port 22/TCP and 514/UDP. This will make the server harder (or longer) to port scan. For a sample client firewall configuration file, please refer to Appendix B. If you are logging additional devices that do not allow port translation, create chains to allow traffic on port 514/UDP from the specific IP address of the device.

Restart the IPTables service with the `service iptables restart` command.

Recognizing Logging Clients

With multiple clients sending logs to the same server, visually determining which client is which can become difficult. This can be made easier by configuring the /etc/hosts file with the known clients. Enter the IP addresses of the client, as well as the name. Each entry will be used by the logging system to convert IP addresses to easily identifiable names. For a sample hosts file, please see Appendix C.

Accurately Storing Log Entries

One of the most important parts of collecting log entries from multiple servers is making sure the correct time of the log entry is entered. This allows an administrator to easily create a picture of what happened by comparing the times the entries were entered. If multiple servers are involved in an intrusion, the correct time is crucial to place an action on one server with the correct result on another.

[5] McCarty, pgs. 261-296

Maintaining the correct time is important. Instead of routinely checking the time on all servers and making changes where they are needed, the network time protocol daemon will be used. Settings for the ntp daemon will be contained within the configuration file located at /etc/ntp.conf. This file contains the location of time servers to connect to in order to synchronize time and settings to use if a network time server was created. For this logging system, time will be synchronized with existing time servers.

When determining servers to synchronize with, special consideration must be made pertaining to the server stratum. Stratum is defined as "The distance a host running the xntpd time daemon is from an external source of Coordinated Universal Time (UTC)" [6] by the SCO Group.[7] If a system has a lower stratum rating then the server should have time that is more accurate, usually within 20 milliseconds for each stratum below the root time server.[8]

Each server should synchronize with multiple time servers that are physically and logically separate. In this example, only two time servers will be used with expectations that a third time server will exist within the existing network the log server will be placed in. See Appendix D for an example ntp configuration file.

To correctly set the time prior to automatically synchronizing the time with the time servers, we need to manually synchronize them. A manual synchronization can be initiated by entering the command `ntpdate –u timeserver`, where timeserver is the time server you want to synchronize with. This command causes the system "to rapidly force it to synchronize" as stated by Linux Home Networking[9]. This command should be run twice for each time server to which the system is being synchronized. To verify the system is synchronizing with other time servers, use the `ntpq –p` command. This will let you view the offset, delay and jitter. If offset and delay are zero and jitter is 4000, the server is not correctly synchronizing and the configuration must be checked.

Restricting Access

In order to be able to track what users access the log server, user accounts will be configured. Each user will have an unprivileged account and an administrative account, creating two additional levels of security before administrative access is granted. Administrative accounts should be protected with stronger passwords and have names that would not easily distinguish them from normal user accounts.

---

[6] http://docsrv.sco.com/NET_tcpip/ntpC.ntp_terms.html.
[7] More information on the SCO Group can be found at http://www.sco.com.
[8] http://www.wilsonmar.com/1clocks.htm
[9] http://www.siliconvalleyccie.com/linux-hn/ntp.htm

Once the accounts have been created, the accounts that have been created for administrative access need to be modified. Enter the command `vipw` to access the /etc/passwd file. This file contains information about each user and the primary group they belong to. Within the file, the administrative accounts can be found. Locate the username and modify the user identification number and the group identification number. These should be set to a user id of 0 and a group id of 0. These two numbers represent the root administrative account and group, allowing the user to perform actions while acting as the root user.

While accessing the file, ensure only accounts that need to login have a shell set. All others should be set to /sbin/nologin. Additionally, ensure all accounts have passwords by viewing the file located at /etc/shadow. If any account that contains a shell also contains two exclamation marks within the same line, a password must be set or change the shell to /sbin/nologin.

Allowing Remote Access

Once all user accounts have been secured, a secure shell should be made available for remote access. Using SSH is a universally recognized practice for a secure method of accessing the server command interface. By editing the /etc/ssh/sshd_config file, remote access can be denied to root accounts through SSH. When editing the sshd_config file, entries that are commented are the default settings. For *PermitRootLogin*, the default setting is yes. Uncomment this line and change the setting to no. This prevents any root account from logging into the server directly. The sshd service will have to be started and configured to start automatically at boot by using the command `service sshd start` and `chkconfig sshd on`, respectively.

As stated earlier, only authorized IP addresses should be allowed to access the log server. These workstations should be accessed by a minimal number of users, but can be the primary workstation of the system administrators. These workstations are defined within the iptables configuration file, explained above. In order to connect, the administrator must connect on the port specified in the iptables configuration file, 17216 within this example, and connect using the normal user account created earlier. Once the user is logged in, to gain administrative access the user must use the command `su – adminuser` where adminuser is the name of the administrative username created earlier. To exit the administrative account, simply type `exit` and the command line for the initial user will be given.

Detecting Local Intrusions

Protecting the log server is important for maintaining validity of the log files. If there is an intrusion to the server, data should not be trusted as valid. However,

the ability to verify data integrity after an intrusion is important. Tripwire is a host-based intrusion detection software package included with Red Hat Linux Version 9.[10] Free and commercial versions of Tripwire[11] are available. When used correctly, Tripwire can help an administrator determine if a server has been compromised and, if so, determine the files that were added, modified, or removed.

Multiple instances of Tripwire will be configured and run to insure database integrity. One will perform an integrity check nightly and update the database with changes. The other will perform the same integrity check; however, it will not update the changes made. This allows for easy comparison between databases to see which files have been modified from the initial setup of the server.

The initial install of Tripwire will not be used for detection. Run the twinstall.sh script located in /etc/tripwire. This will create the configuration files needed for further setup. During the install, passphrases will need to be entered. For the initial setup, any password will suffice as the files will be removed and reconfigured. Once Tripwire is finished installing, run the command `tripwire --init` to initialize the database.

During the initialization of Tripwire, various filenames will be displayed on the screen. Modify the file twpol.txt by commenting out the lines containing these filenames with the exception of the line containing *"$(TWLKEY)/$(HOSTNAME)-local.key*". This filename will be different. Once the modification is finished, remove all files in the /etc/tripwire directory with the exception of twcfg.txt, twpol.txt and twinstall.sh and rerun the twinstall.sh script. If files are still listed besides the previous line, additional modifications are necessary. If no additional files are listed, remove the same files from above, as well as any files in /var/lib/tripwire/, but do not rerun twinstall.sh.

For the security of the server, two separate Tripwire databases will be maintained; therefore, two separate configurations must be made. For ease of installation, two directories should be made, one called daily and the other called life. Copy the files twpol.txt, twcfg.txt and twinstall.sh to each directory. Make additional directories with the same names in /var/lib/tripwire, each containing the directory reports.

First, modify twpol.txt to reflect the changes. At the top of the file, variables for the policy file can be found. For any line that contains tripwire in the variable content, add daily to the path within the variable. Change the hostname to HOSTNAME-Daily, where HOSTNAME is the name of the logging server. Save

[10] Information on current and legacy RedHat products can be found at www.redhat.com and www.fedoralegacy.org.
[11] Information on open-source and commercial Tripwire products can be found out www.tripwire.org and www.tripwire.com.

and close the file. See Appendix E for an excerpt from twpol.txt, including these settings.

Second, modify twcfg.txt similarly to the way twpol.txt was modified. Add daily to the path for any variable containing tripwire. Any variable containing the text $(HOSTNAME) add –daily after $(HOSTNAME). Save and close the file. See Appendix F for an example.

Lastly, modify twinstall.sh with the new variables. Similar to the others, add daily to the path of all needed variables. As with twcfg.txt, add –daily after *$(HOSTNAME)* to reflect the changes. Save and close the file. See Appendix G for an excerpt from twinstall.sh, including these settings.

Once all files have had the appropriate variables replaced, run the script ./twinstall.sh. This will setup similarly to the initial install. Input a password that is different than any other password as it protects the database from unwanted database updates. Separate passwords should be used for the daily and life installations. Once Tripwire is finished installing, run the command `tripwire --init` to initialize the database. After initialization is finished, run `tripwire --check`.

Repeat these steps for the files located under the *life* directory except use *life* instead of *daily*. This will setup the *life* configuration files and database after installation and initialization.

To perform daily checks, modify the tripwire-check configuration file located at /etc/cron.daily/. This file checks to see if the Tripwire database exists and, if so, checks the existence of tw.cfg, the Tripwire configuration file. If both exist, Tripwire will be executed to inventory the system. Modifications must be made to this file to use the new configuration settings, prior to the script executing daily. As with the other Tripwire configuration files, any line that points to the tripwire directory must add */daily/* to the path. In addition, one line will contain *$(HOSTNAME)*. This line should have –daily added after *$(HOSTNAME)* to point to the correct database.

Since this database will be updated daily, a single line will be added after the database has been checked. This line will automatically update the database after the check. Use the command `tripwire –m i –c /etc/tripwire/daily/tw.cfg –P PASSWORD` where PASSWORD is the password used for the daily Tripwire installation. Exit and save the file.

Move *tripwire-check* to the filename *tripwire-daily-check* to help distinguish it from the next file that will be made. Copy *tripwire-daily-check* to *tripwire-life-check*. Edit *tripwire-life-check* and change all occurrences of /daily/ to /life/. Lastly, remove the line that updates the database. This will make all file changes

appear in the report from the beginning of the installation. Verify the script is executable by all before continuing.

Visually inspecting the Tripwire reports can be time consuming, however, if changes are made to the system and the reports are not inspected, an intrusion could go unnoticed. To prevent such an event, modify the file /etc/aliases. This file contains multiple e-mail addresses to be used for different services. Once address is for the root account. Add e-mail addresses of personnel who would inspect the Tripwire reports. Save and close the file and run the command `newaliases`. This will cause the reports to be sent to each address within the root alias.

Retaining Log History

Logrotate is used to move and rename files for archival and to prevent overly large log files. By default, Logrotate will rotate message and secure logs, as well as many others. The default number of rotations is four, meaning Logrotate will keep four weeks worth of logs. The amount of time logs are kept depends on the security policy of the company. For this example, 52 weeks will be the number of log rotations. Edit the file /etc/logrotate.conf. This file contains the settings for Logrotate. Within the file will be the setting for the number of rotations to set Logrotate to. Change this number to 52. Save and exit the file.

Preparing for Remote Logs

In order to tell syslogd to accept remote logs, remote logging must be turned on in two configuration files. The two configuration file settings are similar, so the same change will be performed on both files. Modify the files /etc/init.d/syslog and /etc/sysconfig/syslog. The *SYSLOGD_OPTIONS* variable by default should be set to "-m 0". Add to this the –r option. This will make the line look like *SYSLOGD_OPTIONS="-m 0 –r"*. Save and exit the file and restart syslog using the command *service syslog restart*.

Client Setup

The initial client setup will consist of firewall modification. Edit the file /etc/sysconfig/iptables. Create a prerouting rule to perform DNAT from port 514/UDP to port 17216/UDP for traffic destined to the log server. View Appendix B for a sample client firewall configuration file.

After the firewall setup, Syslog setup is needed. Unlike the server configuration, there is only a single file to modify. This file is /etc/syslog.conf. This file determines where to send log file entries. All messages, with the exception of

mail, are by default placed in the /var/log/messages file, as long as they are of info or higher levels. To send all entries to the remote log server, place an entry at the top of the file similar to *.* @###.###.###.### where ###.###.###.### is the IP address of the remote logging server. Save and exit the file and restart syslog.

Use the same configuration for ntp as was used for the logging server. This will keep log time on the client the same as on the server.

Testing

Prior having all servers log to the logging server, tests should be performed to make sure the security and remote logging is working correctly. To make sure SSH is working correctly; try to SSH to the server on port 22. The server should refuse the connection. Try again to SSH to the server on port 17216. This time, the server should accept the connection as long as the attempt is from an approved SSH client. If an attempt is made from an unapproved SSH client, the connection should be refused.

Next, try to SSH from an approved client to the server on port 17216 using a root account. The server should refuse the connection. If the server allows the login, configuration needs to be modified.

To test the remote logging capability of the server, log into a logging client and use the *logger* command. This can be used to send messages to syslog to see if the entry was sent to the remote server. If the command `logger test` is used, the remote server should show an entry similar to *Sep 26 14:19:57 client root: test*. If the server does not have the entry, there are several possible breaking points. The syslog configuration on both the client and server should be checked, as well as the firewall configuration on both the client and server.

Why Use A Remote Logging Server?

The importance of this server cannot be understated. Without this server, the possibility of compromising the network without being noticed is very high. Many experienced hackers know to remove logs from a compromised system to prevent anybody from tracking the harmful acts back to them. With a remote logging server, any logs that are removed from the client will still be available on the logging server, allowing the administrator to track the activities of the hacker, even if the server is compromised. Without the remote logging server being secure, the possibility of a hacker covering up their tracks in all locations is possible.

Conclusion

Even though this system only has Linux clients, many types of systems use syslog as a logging daemon. With software almost any logging device could log to the remote logging server, including switches, routers, and servers, with minimal configuration on the remote side. When security becomes a factor, knowing what has happened across the network or set of servers is required. To protect this server a firewall was configured to prevent attacks on the server. A secure method of remotely accessing the server was implemented for administrative purposes. Finally, an intrusion detection system was configured to monitor unauthorized changes on the server. All of this information can be used to help track down problems with any equipment logging to the remote logging server, therefore minimizing unnecessary downtime.

References

Fedora Legacy Project, The. 27 Sept, 2004 www.fedoralegacy.org

McCarty, Bill. Red Hat Linux Firewalls. Indianapolis: Wiley, 2003.

Netfilter/iptables project homepage – The netfilter/iptables project. 27 Sept, 2004 <www.netfilter.org>

"NTP glossary". *The SCO Group, Inc. | The Power of UNIX*. 27 Sept, 2004. Online. Available: http://docsrv.sco.com/NET_tcpip/ntpC.ntp_terms.html.

"Keeping Time On Windows Machines". *Welcome to Wilsonmar.com*. 7 Oct, 2004. Online. Available: http://www.wilsonmar.com/1clocks.htm

"Quick HOWTO: Configuring Linux NTP Servers". *Linux Home Networking*. 27 Sept, 2004. Online. Available: http://www.siliconvalleyccie.com/linux-hn/ntp.htm

Red Hat | The Open Source Leader. Red Hat, Inc. 27 Sept, 2004 www.redhat.com

Tripwire, Inc. – Tripwire is the leading provider of Change Monitoring and Analysis software. Tripwire, Inc. 27 Sept, 2004 www.tripwire.com

Tripwire.org – Home of the Tripwire Open Source Project. 27 Sept, 2004 www.tripwire.org

"What is firewall? – A Word Definition From the Webopedia Computer Dictionary". *Webopedia*. 27 Sept, 2004. Online. Available: http://www.webopedia.com/TERM/f/firewall.html

Appendix A

Server Firewall Configuration
/etc/sysconfig/iptables

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
:SYSLOG - [0:0]
:SSH - [0:0]
:SWITCHES - [0:0]


-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -s 192.168.0.237 -j SSH
-A INPUT -s 192.168.0.243 -j SSH
-A INPUT -s 192.168.1.60 -j SSH
-A INPUT -s 192.168.0.6 -j SYSLOG
-A INPUT -s 192.168.0.77 -j SYSLOG
-A INPUT -s 192.168.3.2 -j SYSLOG
-A INPUT -s 192.168.24.6 -j SYSLOG
-A INPUT -s 192.168.5.138 -j SYSLOG
-A INPUT –s 192.168.20.200 –j SWITCHES
-A INPUT -j LOG --log-prefix "LOG-Unknown-IN: " --log-level=3
-A INPUT -j DROP

-A FORWARD -j DROP

-A OUTPUT -j ACCEPT

-A SSH -d 192.168.0.80 -p tcp -m tcp --dport 22 -j LOG --log-prefix "LOG-SSH-IN: " --log-level=3
-A SSH -d 192.168.0.80 -p tcp -m tcp --dport 22 -j ACCEPT

-A SYSLOG -d 192.168.0.80 -p udp -m udp --dport 514 -j ACCEPT

-A SWITCHES -d 192.168.0.80 -p udp -m udp --dport 514 -j ACCEPT

COMMIT

*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]

-A PREROUTING -d 192.168.0.80 -s 192.168.1.60 -p tcp -m tcp --dport 17216 -j DNAT --to-destination 192.168.0.80:22
-A PREROUTING -d 192.168.0.80 -s 192.168.0.237 -p tcp -m tcp --dport 17216 -j DNAT --to-destination 192.168.0.80:22
-A PREROUTING -d 192.168.0.80 -s 192.168.0.243 -p tcp -m tcp --dport 17216 -j DNAT --to-destination 192.168.0.80:22
-A PREROUTING -d 192.168.0.80 -s 192.168.0.6 -p udp -m udp --dport 17216 -j DNAT --to-destination 192.168.0.80:514
-A PREROUTING -d 192.168.0.80 -s 192.168.0.77 -p udp -m udp --dport 17216 -j DNAT --to-destination 192.168.0.80:514
-A PREROUTING -d 192.168.0.80 -s 192.168.3.2 -p udp -m udp --dport 17216 -j DNAT --to-destination 192.168.0.80:514
-A PREROUTING -d 192.168.0.80 -s 192.168.24.6 -p udp -m udp --dport 17216 -j DNAT --to-destination 192.168.0.80:514
-A PREROUTING -d 192.168.0.80 -s 192.168.5.138 -p udp -m udp --dport 17216 -j DNAT --to-destination 192.168.0.80:514


COMMIT
```

Appendix B

Client Firewall Configuration
/etc/sysconfig/iptables

```
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]

-A PREROUTING -d 192.168.0.80 -s 192.168.0.6 -p udp -m udp --dport 514 -j DNAT --to-destination 192.168.0.80:17216

-A POSTROUTING –j ACCEPT

-A OUTPUT –j ACCEPT

COMMIT
```

Appendix C

/etc/hosts

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       logger localhost.localdomain localhost logger.testdomain
192.168.0.6     srv1 srv1.testdomain
192.168.0.42    srv2 srv2.testdomain
192.168.0.77    srv3 srv3.testdomain
192.168.3.2     srv4 srv4.testdomain
192.168.0.80    srv5 srv5.testdomain
192.168.24.6    srv6 srv6.testdomain
192.168.5.138   srv7 srv7.testdomain
```

Appendix D

/etc/ntp.conf

```
server ntp-cup.external.hp.com
server time.nist.gov

fudge   127.127.1.0 stratum 10
#
restrict 127.0.0.1
restrict 0.0.0.0 mask 255.255.255.255 nomodify nopeer noquery notrap

driftfile /var/lib/ntp/drift
multicastclient             # listen on default 224.0.1.1
broadcastdelay  0.008

authenticate no
```

Appendix E

Excerpt from /etc/tripwire/daily/twpol.txt

```
@@section GLOBAL
TWROOT=/usr/sbin;
TWBIN=/usr/sbin;
TWPOL="/etc/tripwire/daily";
TWDB="/var/lib/tripwire/daily";
TWSKEY="/etc/tripwire/daily";
TWLKEY="/etc/tripwire/daily";
TWREPORT="/var/lib/tripwire/daily/report";
HOSTNAME=Logger;
```

Appendix F

/etc/tripwire/daily/twcfg.txt

```
ROOT                    =/usr/sbin
POLFILE                 =/etc/tripwire/tw.pol
DBFILE                  =/var/lib/tripwire/daily/$(HOSTNAME)-daily.twd
REPORTFILE              =/var/lib/tripwire/daily/report/$(HOSTNAME)-daily-$(DATE).twr
SITEKEYFILE             =/etc/tripwire/daily/site.key
LOCALKEYFILE            =/etc/tripwire/daily/$(HOSTNAME)-daily-local.key
EDITOR                  =/bin/vi
LATEPROMPTING           =false
LOOSEDIRECTORYCHECKING  =false
MAILNOVIOLATIONS        =true
EMAILREPORTLEVEL        =3
REPORTLEVEL             =3
MAILMETHOD              =SENDMAIL
SYSLOGREPORTING         =false
MAILPROGRAM             =/usr/sbin/sendmail -oi –t
```

Appendix G

/etc/tripwire/daily/twinstall.sh

```
##---------------------------------------------------------
## Set HOST_NAME variable
##---------------------------------------------------------
HOST_NAME='Logger-daily'
if uname -n > /dev/null 2> /dev/null ; then
      HOST_NAME=`uname -n`
Fi

##---------------------------------------------------------
## Program variables - edited by RPM during initial install
##---------------------------------------------------------

# Site Passphrase variable
TW_SITE_PASS=""

# Complete path to site key
SITE_KEY="/etc/tripwire/daily/site.key"

# Local Passphrase variable
TW_LOCAL_PASS=""

# Complete path to local key
LOCAL_KEY="/etc/tripwire/daily/${HOST_NAME}-local.key"

# If clobber==true, overwrite files; if false, do not overwrite files.
CLOBBER="false"

# If prompt==true, ask for confirmation before continuing with install.
PROMPT="true"

# Name of twadmin executeable
TWADMIN="twadmin"

# Path to twadmin executeable
TWADMPATH=/usr/sbin

# Path to configuration directory
CONF_PATH="/etc/tripwire/daily"

# Name of clear text policy file
TXT_POL=$CONF_PATH/twpol.txt

# Name of clear text configuration file
TXT_CFG=$CONF_PATH/twcfg.txt

# Name of encrypted configuration file
CONFIG_FILE=$CONF_PATH/tw.cfg

# Path of the final Tripwire policy file (signed)
SIGNED_POL=`grep POLFILE $TXT_CFG | sed -e 's/^.*=\(.*\)/\1/'`
```