

Instruction Manual for Tripwire



Developed for

**United States Computer Emergency Readiness Team
(US-CERT)**

**Government Forum of Incident Response and Security Teams
(GFIRST)**

Developed by

USmax & General Dynamics Network Systems

July 29, 2005

TABLE OF CONTENTS

Table Of Contents	i
Table Of Figures	ii
1.0 What Is Tripwire?	3
2.0 System Specifications	3
3.0 Setting Up Tripwire	4
3.1 Obtaining the Source Code	4
3.2 Installing Tripwire	4
3.3 Configuring Tripwire	5
3.3.1 Variable Declaration	5
3.3.2 Key Generation	6
3.3.3 Create the Configuration File.....	7
3.3.4 Create the Policy File.....	7
3.3.5 Securing the Tripwire Files.....	7
3.3.6 Creating a Baseline Database.....	8
3.3.7 Checking the System Against the Database.....	9
3.3.8 Tweaking the Policy File	9
4.0 Using Tripwire	10
4.1 On-Demand Integrity Checking.....	10
4.2 Scheduled, 'Cronjob' Integrity Checking.....	10
5.0 Tripwire Report Files (.twr).....	12
6.0 Maintaining Tripwire	14
6.1 Updating the Policy.....	14
6.2 Updating the Database	15
7.0 Running Tripwire on Secure Medium	17
8.0 Tripwire Options & Reference Materials	19

TABLE OF FIGURES

Figure 1 – pwd, cd, su, and rpm.....	4
Figure 2 – Key Generation.....	6
Figure 3 – Configuration File Creation.....	7
Figure 4 – Policy File Creation.....	7
Figure 5 – Original Owner and Permissions.....	8
Figure 6 – Owner and Permissions Commands.....	8
Figure 7 – Database Initialization and Check.....	9
Figure 8 – System Integrity Check and Report.....	10
Figure 9 – Sample Tripwire Report.....	13
Figure 10 – Updating the Tripwire Policy.....	14
Figure 11 – Tripwire Database Update.....	15
Figure 12 – Database Violations.....	15
Figure 13 – Initializing/rewriting the database.....	16
Figure 14 – Tripwire man page.....	19

1.0 WHAT IS TRIPWIRE?

Tripwire is an open source, host-based intrusion detection application, which is used to verify system integrity. Unlike signature-based intrusion detection systems, Tripwire is not designed to detect active intrusion attempts. Rather, Tripwire tracks changes to made to the host. To do this, Tripwire maintains an attribute database of all system files and the respective byte count of each file. The system is routinely scanned and compared against this database, and any anomalies are flagged. The results of the comparison can be provided to administrators, detailing which files have changed, and the criticality level of the offense.

2.0 SYSTEM SPECIFICATIONS

This document addresses obtaining, compiling and executing Tripwire according to the system specifications below. For other Linux-based systems (distributions or architectures), the compilation and execution procedure will be similar to the approach discussed in this document. All other platforms may use Section 3.3 Configuring Tripwire, as the Tripwire configuration is cross-platform.

Operating System	Linux SuSE 9.0 Professional
Kernel	2.4.21-291
Tripwire	2.3.1-185
Requirements	Internet Connection/Browser Privileged account access (root)
Source files	http://rpmfind.net/linux/rpm2html/search.php?query=tripwire

3.0 SETTING UP TRIPWIRE

The Tripwire installation process involves obtaining the source code and an RPM Packet Manager (RPM) installation. Following the installation, Tripwire can be configured with passwords and specific directories.

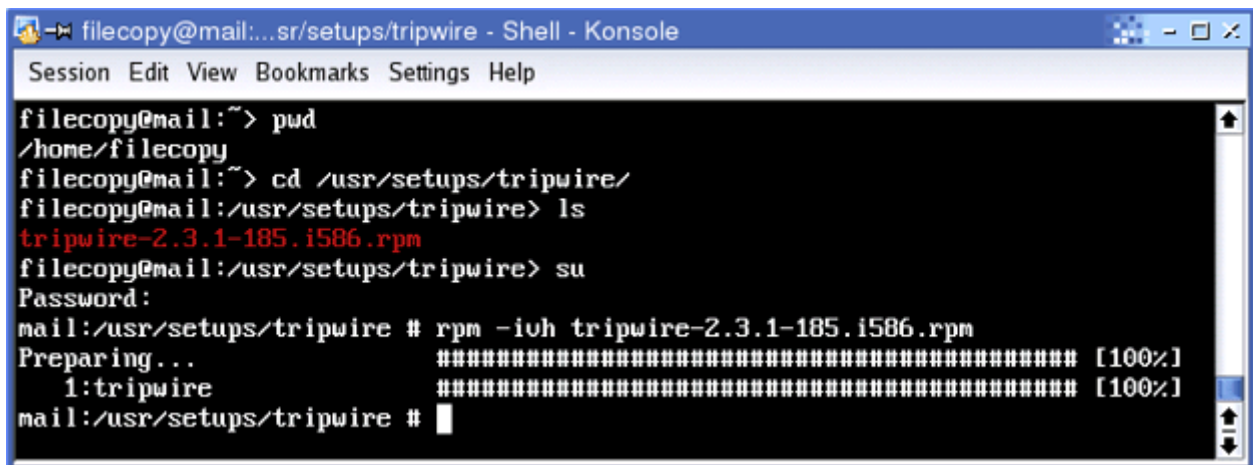
3.1 Obtaining the Source Code

- 1) Using an internet browser (Internet Explorer, Netstape Navigator, Firefox, etc.), navigate to <http://rpmfind.net/linux/rpm2html/search.php?query=tripwire>
 - a) Looking at the “Distribution” column, scroll down to ‘SuSE Linux 9.0 Updates for i386.’
 - b) After finding this distribution’s row, the last column, “Download” should be titled “tripwire-2.3.1-185.i586.rpm”
 - c) There appears to be to identical rows, but one row contains “tripwire-2.3.1-185.i586.rpm” and is actually a patch. By hovering the mouse over both links, the source and patch files can be distinguished.
 - d) Click the row’s download link that is not the patch, and when prompted, save the file to an accessible directory. (i.e. usr/setups/tripwire/). This file is in the RPM format, denoted by the rpm extension. The file should be called tripwire-2.3.1-185.i586.rpm. If the downloaded file’s title does not match, re-download the correct file.
- 2) After downloading the correct file, proceed to Section ‘3.2 Installing Tripwire.’

3.2 Installing Tripwire

With the source RPM file correctly downloaded, the system is now ready to install Tripwire.

- 1) To install Tripwire, navigate to the directory where the rpm was saved, using ‘cd’ (Figure 1).
- 2) Only a privileged user can install Tripwire. To become the root user, enter ‘su’ followed by a <return> and enter the root password, as shown in Figure 1.
- 3) Once in the correct directory and a privileged user, install the rpm using the ‘rpm’ command. The options, ivh, refer to install (i), verbose (v), and hash (h). The hash option refers to the ‘#’ progression that scrolls across during installation, as shown in Figure 1.



```
filecopy@mail:~> pwd
/home/filecopy
filecopy@mail:~> cd /usr/setups/tripwire/
filecopy@mail:/usr/setups/tripwire> ls
tripwire-2.3.1-185.i586.rpm
filecopy@mail:/usr/setups/tripwire> su
Password:
mail:/usr/setups/tripwire # rpm -ivh tripwire-2.3.1-185.i586.rpm
Preparing...      ##### [100%]
 1:tripwire      ##### [100%]
mail:/usr/setups/tripwire #
```

Figure 1 – pwd, cd, su, and rpm. This figure shows the commands to navigate from the current directory (‘pwd’) to the download directory (‘cd’), and show the directory (‘ls’) contents. The user then switches to a privileged user (su) to finish installation. The source code is then installed (‘rpm -ivh’). The installation was successful as shown by the two lines stating ‘100%.’

3.3 Configuring Tripwire

After Tripwire has been installed, it must be configured with passwords and a database. The location of files can also be altered if the user desires to use specific directories other than the default selection.

3.3.1 Variable Declaration

To change the location of file storage and other parameters, the base configuration file `/etc/tripwire/twcfg.txt` can be changed. Within this file there are five required variables that can be customized by a privileged user. The files and their descriptions are listed below.

- 1) POLFILE – Location of the policy file; `/etc/tripwire/tw.pol` is the default.
- 2) DBFILE – Location of the database file; `/var/lib/tripwire/$(HOSTNAME).twd` is the default.
- 3) REPORTFILE – Location of the report files. By default is set to `/var/lib/tripwire/report/$(HOSTNAME)-$(DATE).twr`.
- 4) SITEKEYFILE – Location of the site key file; `/etc/tripwire/site.key` is the default.
- 5) LOCALKEYFILE – Location of the local key file; `/etc/tripwire/$(HOSTNAME)-local.key` is the default.

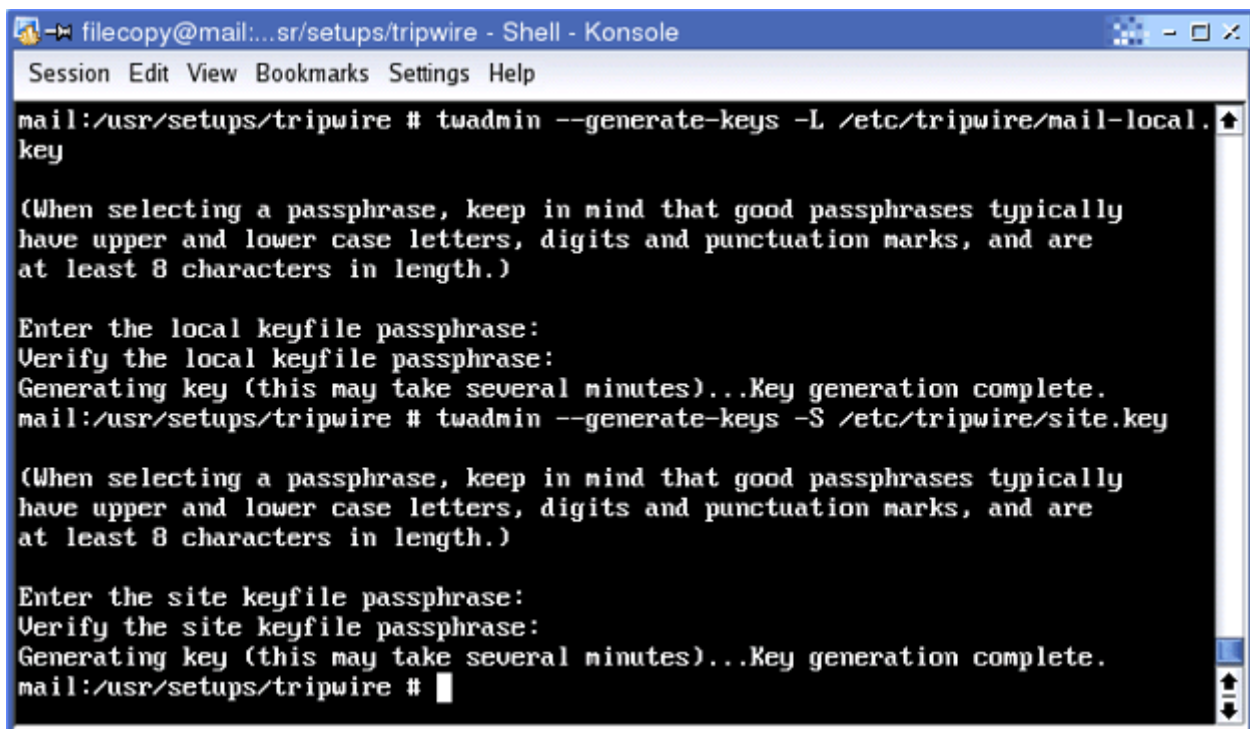
Beyond the required variables, there are several optional variables that the user may set for the Tripwire system. These variables and their description are discussed below.

- 1) EDITOR – Specifies the text editor called by Tripwire. The default value is `/bin/vi`.
- 2) LATEPROMPTING – If set to true, this variable configures Tripwire to wait as long as possible before prompting the user for a password, minimizing the amount of time the password is in memory. The default value is false.
- 3) LOOSEDIRECTORYCHECKING – If set to true, this variable configures Tripwire to report if a file within a watched directory changes, but not to report the change for the directory itself. This limits redundancy in Tripwire reports. The default value is false.
- 4) SYSLOGREPORTING – If set to true, this variable configures Tripwire to report information to the syslog daemon via the "user" facility. The log level is set to notice. See the `syslogd` man page for more information. The default value is false.
- 5) MAILNOVIOLATIONS – If set to true, this variable configures Tripwire to email a report at a regular interval regardless of whether any violations have occurred. The default value is true.
- 6) EMAILREPORTLEVEL – Specifies the level detail for emailed reports. Valid values for this variable are 0 through 4. The default value is 3.
- 7) REPORTLEVEL – Specifies the level detail for reports generated by the `twprint` command. This value can be overridden on the command line, but is set to 3 by default.
- 8) MAILMETHOD – Specifies which mail protocol Tripwire should use. Valid values are SMTP and SENDMAIL. The default value is SENDMAIL.
- 9) MAILPROGRAM – Specifies which mail program Tripwire should use. The default value is `/usr/sbin/sendmail -oi -t`.

3.3.2 Key Generation

After changing the required variables, or setting the optional variables, the Tripwire keys should now be set. The keys are used to protect the data (database, configuration files, etc.) from unauthorized access. The steps to establish these keys are listed below. Figure 2 below traces the commands used to create the keys.

- 1) The first key to set is the local key, by using the following steps.
 - a) Execute `twadmin --generate-keys -L /etc/tripwire/HOSTNAME-local.key`
 - b) Here, `HOSTNAME` is replaced with the hostname of the Tripwire system (in this case, `mail` is the hostname). Note: the file location can be changed to any desired location.
 - c) Enter (and re-enter for validation) the passphrase for the local key.
 - d) After key generation, the local key has been created in `/etc/tripwire/`
 - e) The key owner and permissions should be set so that only root can read or write the key file, as discussed in Section 3.3.5.
- 2) The second key to create is the site key, using the following steps.
 - a) Execute `twadmin --generate-keys -S /etc/tripwire/site.key` (Note: the file location can be changed to any desired location)
 - b) Enter (and re-enter for validation) the passphrase for the site key.
 - c) After generation, the site key has been created in `/etc/tripwire/`
 - d) The key file owner and permissions should be set so that only root can read or write the key file, as discussed in Section 3.3.5.



```
filecopy@mail:...sr/setups/tripwire - Shell - Konsole
Session Edit View Bookmarks Settings Help
mail:/usr/setups/tripwire # twadmin --generate-keys -L /etc/tripwire/mail-local.
key

(When selecting a passphrase, keep in mind that good passphrases typically
have upper and lower case letters, digits and punctuation marks, and are
at least 8 characters in length.)

Enter the local keyfile passphrase:
Verify the local keyfile passphrase:
Generating key (this may take several minutes)...Key generation complete.
mail:/usr/setups/tripwire # twadmin --generate-keys -S /etc/tripwire/site.key

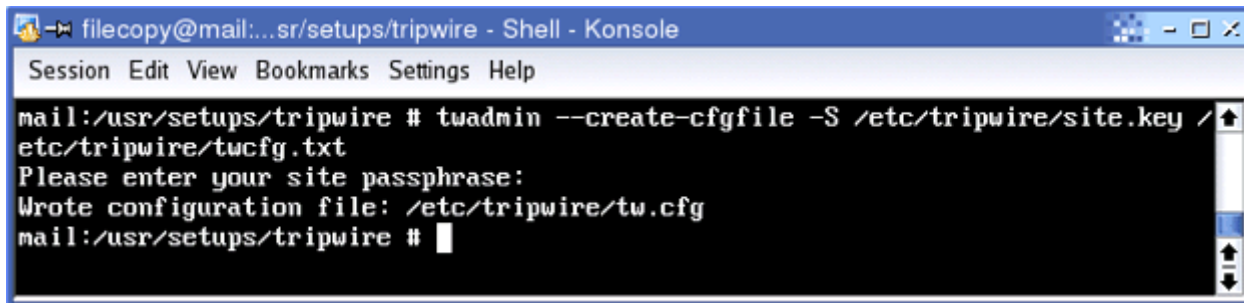
(When selecting a passphrase, keep in mind that good passphrases typically
have upper and lower case letters, digits and punctuation marks, and are
at least 8 characters in length.)

Enter the site keyfile passphrase:
Verify the site keyfile passphrase:
Generating key (this may take several minutes)...Key generation complete.
mail:/usr/setups/tripwire #
```

Figure 2 – Key Generation. This figure shows the process to create a Local key (first command) and Site key (second command). The keys are protected by passwords that may or may not be the same.

3.3.3 Create the Configuration File

Because Tripwire is a system integrity intrusion detector, the configuration files and other Tripwire files should not be stored in clear text. By using the keys generated in the previous section, the configuration template file can be encrypted. As shown in Figure 3, the command 'twadmin --create-cfgfile -S /etc/tripwire/site.key /etc/tripwire/twcfg.txt' creates a file named 'tw.cfg' in /etc/tripwire.

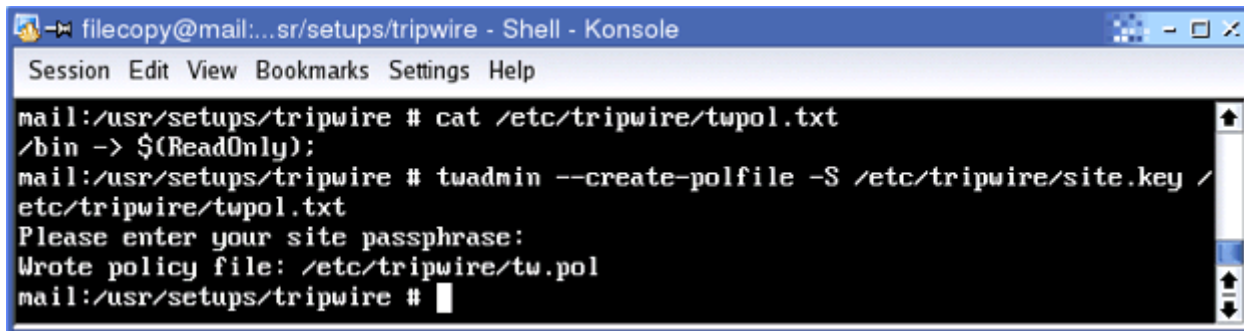


```
filecopy@mail:...sr/setups/tripwire - Shell - Konsole
Session Edit View Bookmarks Settings Help
mail:/usr/setups/tripwire # twadmin --create-cfgfile -S /etc/tripwire/site.key /
etc/tripwire/twcfg.txt
Please enter your site passphrase:
Wrote configuration file: /etc/tripwire/tw.cfg
mail:/usr/setups/tripwire #
```

Figure 3 – Configuration File Creation. This command creates the encrypted configuration file tw.cfg using the Site key and the twcfg.txt plaintext configuration file.

3.3.4 Create the Policy File

For the purposes of this document, the policy file will only consist of a single rule. A complex example file can be viewed in '/usr/share/doc/packages/tripwire/twpol.txt.' The cleartext policy file template, twpol.txt, must be created and saved in '/etc/tripwire/.' The only rule in the test policy file is "/bin -> \$(ReadOnly);", as shown in Figure 4 below. After creating this file, the encrypted policy file is created using the Site key and the twpol.txt plaintext file. The output is the encrypted tw.pol file, shown in Figure 4.



```
filecopy@mail:...sr/setups/tripwire - Shell - Konsole
Session Edit View Bookmarks Settings Help
mail:/usr/setups/tripwire # cat /etc/tripwire/twpol.txt
/bin -> $(ReadOnly);
mail:/usr/setups/tripwire # twadmin --create-polfile -S /etc/tripwire/site.key /
etc/tripwire/twpol.txt
Please enter your site passphrase:
Wrote policy file: /etc/tripwire/tw.pol
mail:/usr/setups/tripwire #
```

Figure 4 – Policy File Creation. This figure shows the single rule in the plaintext policy file, twpol.txt. The second command shows the creation of the encrypted policy file, tw.pol, using the Site key and plaintext twpol.txt policy file.

3.3.5 Securing the Tripwire Files

Now that the keys, encrypted configuration and encrypted policy file have been created, it is important to protect these files. To do this, each file should have its owner and group changed to 'root,' and its permissions altered so that only 'root' can read or write to the files (in this example, the owner and group are already 'root,' but will be performed as a useful exercise). The permissions command uses the Unix triad of Read-Write-Execute in binary and will be set to 600. The list of files and their respective permissions prior to any change are shown in Figure 5, followed by the altered permissions shown in Figure 6.

```

filecopy@mail:...sr/setups/tripwire - Shell - Konsole
Session Edit View Bookmarks Settings Help
mail:/etc/tripwire # ll
total 46
drwxr-xr-x  2 root  root    216 2005-07-28 11:43 .
drwxr-xr-x 57 root  root   6192 2005-07-27 17:08 ..
-rw-r--r--  1 root  root    931 2005-07-28 11:26 mail-local.key
-rw-r--r--  1 root  root    931 2005-07-28 11:26 site.key
-rw-r--r--  1 root  root   4586 2005-07-28 11:39 tw.cfg
-rw-r--r--  1 root  root   1017 2005-07-28 11:38 twcfg.txt
-rw-r--r--  1 root  root   4159 2005-07-28 11:41 tw.pol
-rw-r--r--  1 root  root   9278 2005-07-28 11:41 twpol.txt
mail:/etc/tripwire #

```

Figure 5 – Original Owner and Permissions. This figure shows the list of the files prior to any changes made to owner or permissions. The key files are readable by the group and world, which poses a risk to signed-but-not-encrypted report files (users who have read access to reports and keys can view signed files).

```

filecopy@mail:...sr/setups/tripwire - Shell - Konsole
Session Edit View Bookmarks Settings Help
mail:/etc/tripwire # chown root.root *
mail:/etc/tripwire # chmod 600 *
mail:/etc/tripwire # ll
total 46
drwxr-xr-x  2 root  root    216 2005-07-28 11:43 .
drwxr-xr-x 57 root  root   6192 2005-07-27 17:08 ..
-rw-----  1 root  root    931 2005-07-28 11:26 mail-local.key
-rw-----  1 root  root    931 2005-07-28 11:26 site.key
-rw-----  1 root  root   4586 2005-07-28 11:39 tw.cfg
-rw-----  1 root  root   1017 2005-07-28 11:38 twcfg.txt
-rw-----  1 root  root   4159 2005-07-28 11:41 tw.pol
-rw-----  1 root  root   9278 2005-07-28 11:41 twpol.txt
mail:/etc/tripwire #

```

Figure 6 – Owner and Permissions Commands. This figure shows the commands to change the owner and group permissions for all files in this directory using ‘chown’ and ‘chmod.’

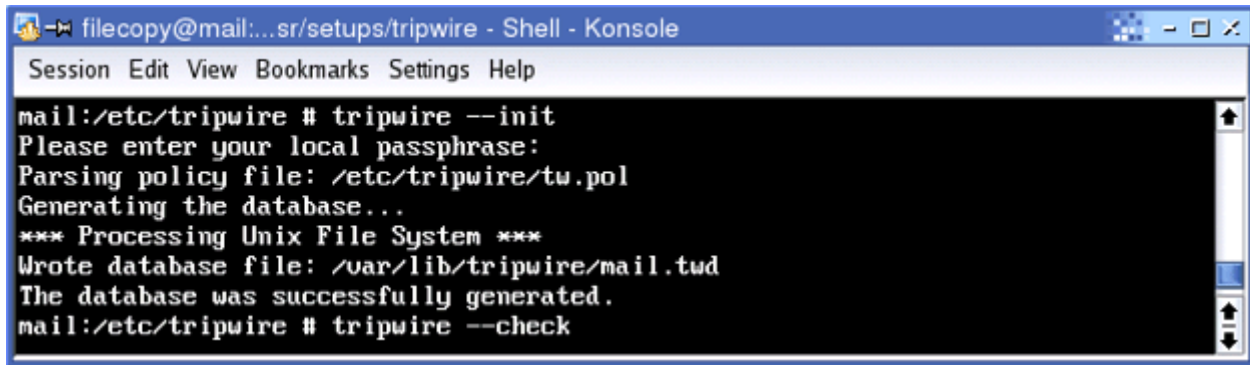
In the figure above, all of the file permissions were modified. This includes the Tripwire keys (site and local), configuration file (tw.cfg) and policy file (tw.pol), as well as their templates (twcfg.txt and twpol.txt). The final permissions are set so that only ‘root’ can read and write these files (-rw-----).

3.3.6 Creating a Baseline Database

Using the policy file previously created, the database will be initialized with a current snapshot of the objects specified in the policy rules. The baseline database will be stored in /var/lib/tripwire/<HOSTNAME>.twd. The initialization command is shown in Figure 7 of the next section.

3.3.7 Checking the System Against the Database

When Tripwire was initialized, a snapshot of the system was taken and stored in a database to be used as a reference. When it is time to compare the current system state against this snapshot database, the 'check' parameter is used. This will compare the attributes and files of the current system against that of the database snapshot, and inform the user of the differences, if any, between the two sets of information. The command to check the current system state against the saved database snapshot is 'tripwire --check', as shown in Figure 7.

A screenshot of a terminal window titled "filecopy@mail:...sr/setups/tripwire - Shell - Konsole". The terminal shows the following text:

```
mail:/etc/tripwire # tripwire --init
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
Wrote database file: /var/lib/tripwire/mail.twd
The database was successfully generated.
mail:/etc/tripwire # tripwire --check
```

Figure 7 – Database Initialization and Check. This figure shows the command to initialize the database (--init) and then compare the current system against the initialized database (--check).

Because the initialization and check were executed in a short amount of time, there were no changes made to the files that were checked, so no errors were reported. The results of the check were sent to the standard output (screen), and logged to a file. This file is located in '/var/lib/tripwire/report/<HOSTNAME>-YYYYMMDD-HHMMSS.twr.' The user may review any report at any time using the command "twprint --print-report -r /var/lib/tripwire/report/<HOSTNAME>-YYYYMMDD-HHMMSS.twr." For this filename, YYYY refers to the four-digit year (e.g. 2005), MM refers to the numeric month (e.g. 02 for February), DD refers to the numeric day of the month (e.g. 24th), HH refers to the hour in military time (e.g. 18 is 6 PM), MM refers to the minute, and SS refers to the seconds.

3.3.8 Tweaking the Policy File

Depending on the policy file and associated rules used to create the Tripwire database, there may be warnings and alerts noting files that are not necessarily on the Tripwire system, or should not be monitored. In order to reduce the number of alerts relating to frequently changing files and directories, the policy file needs to be updated. By comparing the twpol.txt file against the report generated from the Tripwire check, rules can be added or removed, and files commented out so that inappropriate alerts are avoided. A side-by-side comparison of the policy file (/etc/tripwire/twpol.txt) and the Tripwire report (/var/lib/tripwire/report/<HOSTNAME>-YYYYMMDD-HHMMSS.twr) will allow the user to edit the policy file to avoid receiving alerts relating to frequently changing files, such as logs in the /var directory.

After editing the policy file to handle over-zealous alerts and warnings, the policy file needs to be re-encrypted (twadmin --create-polfile -S /etc/tripwire/site.key /etc/tripwire/twpol.txt) and the database reinitialized (tripwire --init). After establishing the final policy file, clear text files (twpol.txt and twcfg.txt) should be removed to prevent unauthorized viewing of Tripwire policies in the event of a system breach, which may allow the attacker to cover their tracks.

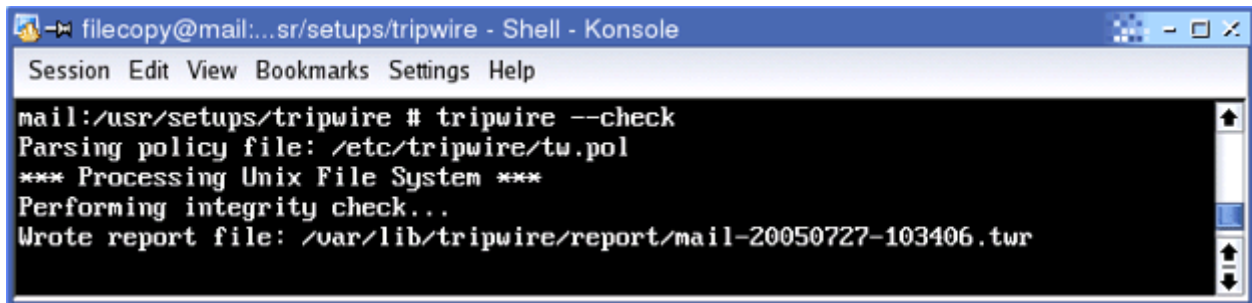
4.0 USING TRIPWIRE

There are typically two methods of using Tripwire, after the setup and configuration is complete. These methods include ad-hoc file system integrity checking on an on-demand basis, as well as a regularly scheduled ‘cronjob’ check to make sure that the system is in proper order. This section discusses these two methods and their implementation.

4.1 On-Demand Integrity Checking

The ‘On-Demand’ type of Tripwire use is convenient for integrity checking when a system has possibly been breached or compromised, and an immediate check is required. It is also useful when generating a real-time report of the current system state.

The steps to execute ‘On-Demand’ integrity checking are the same as that of Section 3.3.7. By simply executing ‘tripwire --check’, the system will be compared against the initialized database according to the rules set in the Tripwire policy file (Figure 8). After the check has completed its comparison, the report is stored in ‘/var/lib/tripwire/report/<HOSTNAME>-YYYYMMDD-HHMMSS.twr.’ The ‘check’ results are discussed in Section 5.0.



```
filecopy@mail:...sr/setups/tripwire - Shell - Konsole
Session Edit View Bookmarks Settings Help
mail:/usr/setups/tripwire # tripwire --check
Parsing policy file: /etc/tripwire/tw.pol
*** Processing Unix File System ***
Performing integrity check...
Wrote report file: /var/lib/tripwire/report/mail-20050727-103406.twr
```

Figure 8 – System Integrity Check and Report. This figure shows the command to check the current system state against the Tripwire database using the rules in the policy file ‘/etc/tripwire/tw.pol.’ The report file has been written to ‘/var/lib/tripwire/report’ as shown.

4.2 Scheduled, ‘Cronjob’ Integrity Checking

An excellent feature of Tripwire is its ability to be scripted and scheduled, so that the system checking can be performed regularly and automatically. Like many other Linux tools, the scheduling daemon, cron, can be setup to execute Tripwire commands. This section discusses creating a ‘cronjob’ to check the system with Tripwire, and email the results at set intervals.

To schedule the Tripwire check and distribute the results via email, the cronjob must be set up by editing the crontab. The crontab is the scheduler for cron to perform specific tasks at specific times. To edit the crontab using vi, execute ‘crontab –e’ and press Enter. Creating an entry in the crontab is very easy, as there is a specific format and syntax to follow as shown below. For proper execution, the six fields must be completed in the order listed.

Field	Description	Values
Minute	Minute of execution	0-59
Hour	Hour of execution	0-23
M-day	Day of month of execution	1-31
Month	Month of execution	1-12 or names (April-July)
W-day	Day of week of execution	0-7 (0,7 is Sunday) or names (Monday-Thursday)
Command	Command line program	/usr/sbin/tripwire

The fields mentioned above are completed in top-down order when written into the crontab. The values mentioned also have format restrictions, which are explained below. The restrictions apply to the minute, hour, M-day, Month and W-day fields.

Value	Description
*	Matches all values (* in minutes means every minute, * in month means every month)
x-y	Matches the x to y range (2-4 in Mday means 2 nd , 3 rd , 4 th of the month)
x/n	In x range with n frequency (2-6/2 in Month means every other month from Feb-June)

As an example, the following crontab entry will be used to show the fields, values and command.

```
01 02 * 2-4 * /usr/sbin/tripwire --check
```

This entry will check the system integrity against the database at 2:01AM every day for the months of February, March and April, using the command `/usr/sbin/tripwire --check`.

After establishing the schedule that Tripwire will follow for regular system checks, the cronjob can also be set to email the results to a specified email address after complete. The example below shows that Tripwire is called to check a specific folder at the scheduled time, and email the results.

```
05 14 27 July * /usr/sbin/tripwire --check --email-report > /dev/null
```

This example runs at 2:05PM on July 27, and checks the system for changes against the database according to the policy rules. After it is complete, the results are emailed to the address stored in the policy file. Sections of the report file can be emailed to different accounts, in the case of segmented administration of a system.

The report sent via email (or even posted to a website) is the same report generated when the Tripwire check utility is run by itself. The user can view and assess the results, and decide to take action, update the database, or do nothing. The following section discusses the report file generated during a check.

5.0 TRIPWIRE REPORT FILES (.TWR)

As discussed in earlier sections, Tripwire creates a report file that compares the Tripwire database against the current system state, whenever the `--check` is invoked. The standard location for the report files is `/var/lib/tripwire/report/` using a file name with the following format: `<HOSTNAME>-YYYYMMDD-HHMMSS.twr`. The generated report is stored in compressed format, and not readable using standard editors. If the report were encrypted (tripwire was invoked using the `--signed-report` option), the encryption would have to be removed in order to see the report. Because the reports are only compressed, users with read access to the Tripwire configuration and report file will be able to read the reports. However, since the key, configuration and policy file permissions were changed to only allow read/write access from root, other users cannot view these files.

In order to view the compressed Tripwire report file, the following command must be executed: `twprint --print-report --twrfile /var/lib/tripwire/report/filename`. This command will print the report to stdout (screen). If the user wishes to print the report to a clear text file viewable by others, the following command would be used, which pipes the report to a user-specified file. `twprint --print-report --twrfile /var/lib/tripwire/report/filename /path/to/report/output.file`.

The format of the uncompressed, unencrypted report contains four (4) major sections, detailing the results of the Tripwire check. The sample report shown in Figure 9 illustrates the main sections, including Report Summary, Rule Summary, Object Detail and Error Report. The information contained in each section is as follows:

Report Summary - This lists general system and Tripwire configuration information

Rule Summary - This lists the rules violated, the severity and whether violated rule was caused by the addition, removal or modification of the violated object.

Object Detail - This section details the violated objects, provided detailed statistics on the changes, including times, sizes and properties.

Error Report - This lists any errors encountered while Tripwire was executed.

In the sample report, the Report Summary details that the command used to invoke Tripwire was `'tripwire --check --signed-report /data/temp/'` which checks the `'/data/temp/'` folder against the initialized database. For this session, the output report will be signed with the Local key. There has also been one rule violation noted, according to the Rule Summary. This rule notes that `'/data'` has had a violation (modification in this case), with severity 100, this highest of all. The Object Detail section identifies three violations, consisting of size and modification and change times. Finally, no error was reported in this check.

```

Tripwire(R) 2.3.0 Integrity Check Report

Report generated by:      root
Report created on:       Wed 27 Jul 2005 03:34:25 PM EDT
Database last updated on: Never
=====
Report Summary:
=====
Host name:               mail
Host IP address:         192.168.0.3
Host ID:                 None
Policy file used:        /etc/tripwire/tw.pol
Configuration file used: /etc/tripwire/tw.cfg
Database file used:      /var/lib/tripwire/mail.twd
Command line used:       tripwire --check --signed-report /data/temp/
=====
Rule Summary:
=====
-----
Section: Unix File System
-----
-----
Rule Name                Severity Level   Added   Removed   Modified
-----
* System Files           100              0       0          1
  (/data)

Total objects scanned:  1
Total violations found: 1
=====
Object Detail:
=====
-----
Section: Unix File System
-----
-----
Rule Name: System Files (/data)
Severity Level: 100
-----
-----
Modified Objects: 1
-----
Modified object name:  /data/temp

Property:                Expected          Observed
-----
* Size                   312              344
* Modify Time            Wed 27 Jul 2005 02:22:14 PM EDT
                        Wed 27 Jul 2005 03:31:20 PM EDT
* Change Time           Wed 27 Jul 2005 02:22:14 PM EDT
                        Wed 27 Jul 2005 03:31:20 PM EDT
=====
Error Report:
=====
No Errors
-----
*** End of report ***

Tripwire 2.3 Portions copyright 2000 Tripwire, Inc. Tripwire is a registered
trademark of Tripwire, Inc. This software comes with ABSOLUTELY NO WARRANTY;
for details use --version. This is free software which may be redistributed
or modified only under certain conditions; see COPYING for details.
All rights reserved.

```

Figure 9 – Sample Tripwire Report. This example shows a simple report generated from a Tripwire check on a specific object (/data/temp). There has been one object violated, consisting of three property violations.

6.0 MAINTAINING TRIPWIRE

It is highly unlikely that a system put in place will never be tuned, patched or upgraded in the course of its service. When systems are updated, reconfigured, or tweaked, Tripwire will likely need to also be updated to incorporate the modifications to the system so errors are not unnecessarily reported. There are two potential areas of update for Tripwire; the policy file containing the rules and directories to monitor, and the database snapshot of a “clean” system. These two updates are discussed in the next two subsections.

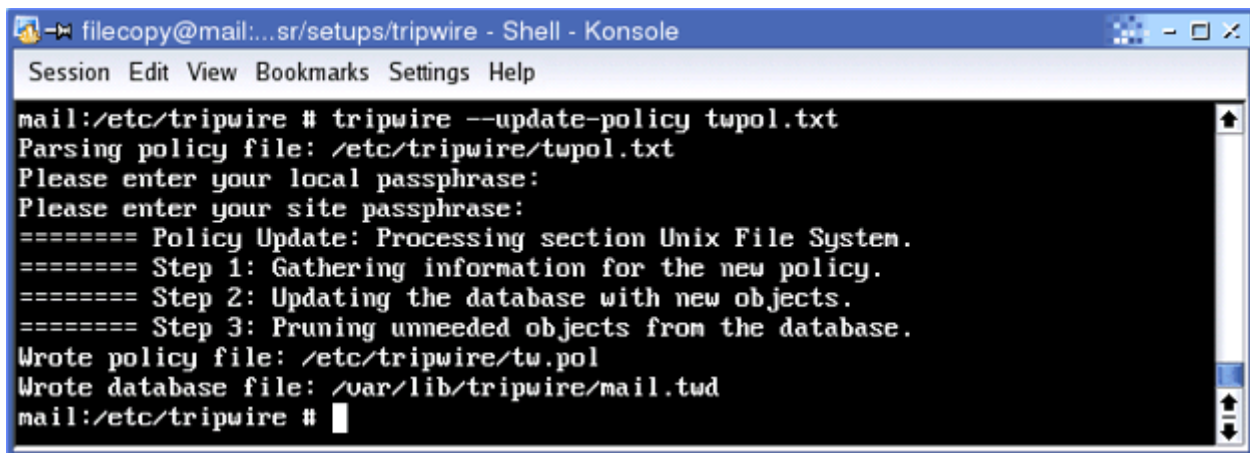
6.1 Updating the Policy

A main source of alerts and errors is an incorrectly implemented policy file. If the system is monitoring modification dates of log files for instance, numerous errors will be reported very frequently because these files are routinely updated and created. In this case, perhaps a less stringent rule should be invoked to only monitor for ownership or permission changes. To change these rules, the policy file must be updated and reloaded for Tripwire to use.

In order to update the policy file, the root user (because of policy file permissions) needs to use a favorite editor to open the twpol.txt file in the /etc/tripwire directory. Additional files and directories that should be monitored can be added, entries can be removed to reflect the current system status, and the severity rules can also be changed. The root user makes the appropriate changes and saves the file prior to closing.

After updating the plaintext file, a new encrypted tw.pol file must be created as was done during Tripwire setup. The command ‘twadmin --create-polfile --polfile tw.pol twpol.txt’ will create the encrypted tw.pol file, and save the old version as tw.pol.bak.

An alternative way to update the policy file is to use a Tripwire option at the command line. The ‘--update-policy’ option is a one-step method of updating the policy file, and performs database pruning to reflect the policy change. Figure 10 below shows the command and the password prompts.



```
filecopy@mail:...sr/setups/tripwire - Shell - Konsole
Session Edit View Bookmarks Settings Help
mail:/etc/tripwire # tripwire --update-policy twpol.txt
Parsing policy file: /etc/tripwire/twpol.txt
Please enter your local passphrase:
Please enter your site passphrase:
===== Policy Update: Processing section Unix File System.
===== Step 1: Gathering information for the new policy.
===== Step 2: Updating the database with new objects.
===== Step 3: Pruning unneeded objects from the database.
Wrote policy file: /etc/tripwire/tw.pol
Wrote database file: /var/lib/tripwire/mail.tud
mail:/etc/tripwire #
```

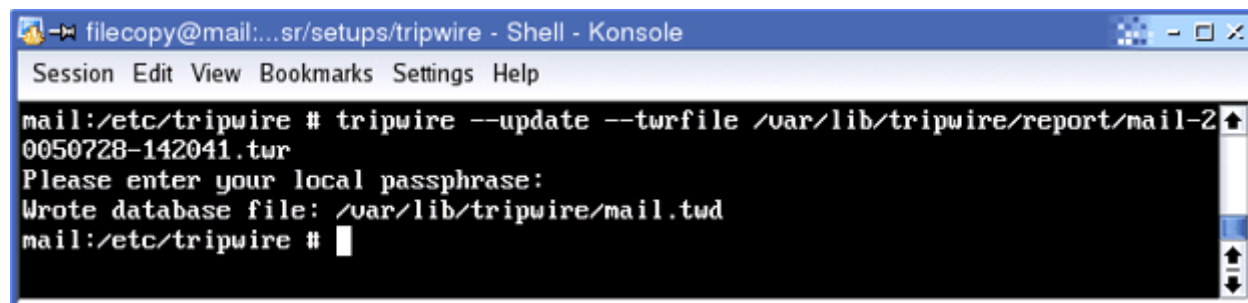
Figure 10 – Updating the Tripwire Policy. This shows the command and prompts to update the Tripwire policy. A new policy file was created, and the database was pruned accordingly.

Now that the policy file has been updated, attempts to check the system integrity will likely fail until the Tripwire database has also been updated to handle the changes. Database updates are discussed in the following subsection.

6.2 Updating the Database

As a first note, the database update does not always have to follow a policy update. If there are rules within the policy file to only monitor /var/tmp for ownership or permissions changes, files within this directory can be modified, added or deleted without requiring a policy update. The footprint database should be updated periodically to accurately capture the system state, so that the initial database does not become stale and obsolete.

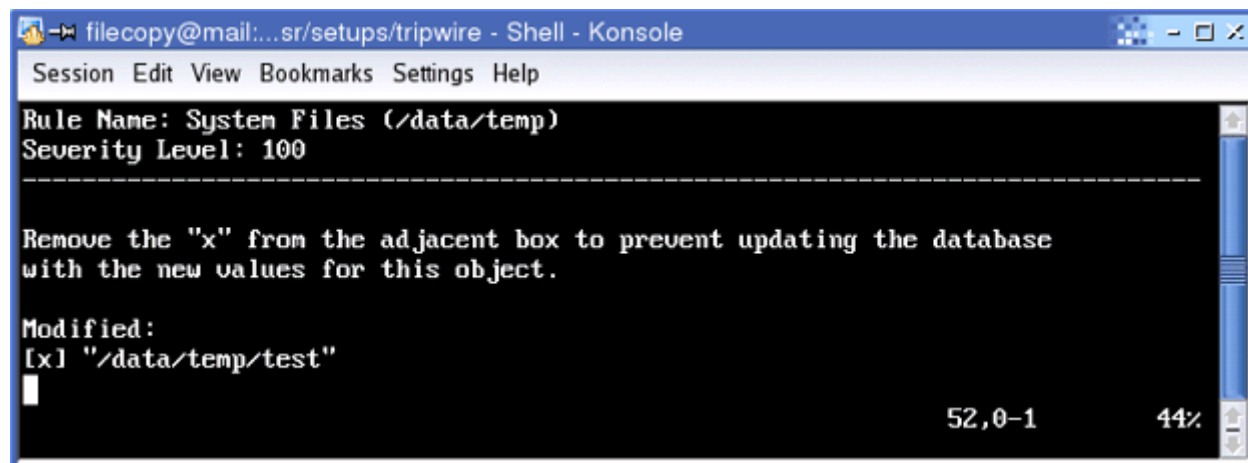
Typically, updates to the database are provided from a report file, which shows the differences between the initialized database and the current system state. Figure 11 shows the command that will update the database using a Tripwire report file.



```
filecopy@mail:...sr/setups/tripwire - Shell - Konsole
Session Edit View Bookmarks Settings Help
mail:/etc/tripwire # tripwire --update --twrfile /var/lib/tripwire/report/mail-20050728-142041.twr
Please enter your local passphrase:
Wrote database file: /var/lib/tripwire/mail.tud
mail:/etc/tripwire #
```

Figure 11 – Tripwire Database Update. This figure shows the command to update the Tripwire database, using a report file located in /var/lib/tripwire/report, with a file name including the date and time.

What is not shown in the above figure is that after executing the first command, the text editor specified in the twcfg.txt and tw.cfg files is called and opens a modified report file. Within this report file, all discrepancies between the database and the report file are marked with an ‘X’ as shown in Figure 12 below. Valid violations should have the X removed, whereas all discrepancies that are to be used to update the database should be left with the X.



```
filecopy@mail:...sr/setups/tripwire - Shell - Konsole
Session Edit View Bookmarks Settings Help
Rule Name: System Files (/data/temp)
Severity Level: 100
-----
Remove the "x" from the adjacent box to prevent updating the database
with the new values for this object.

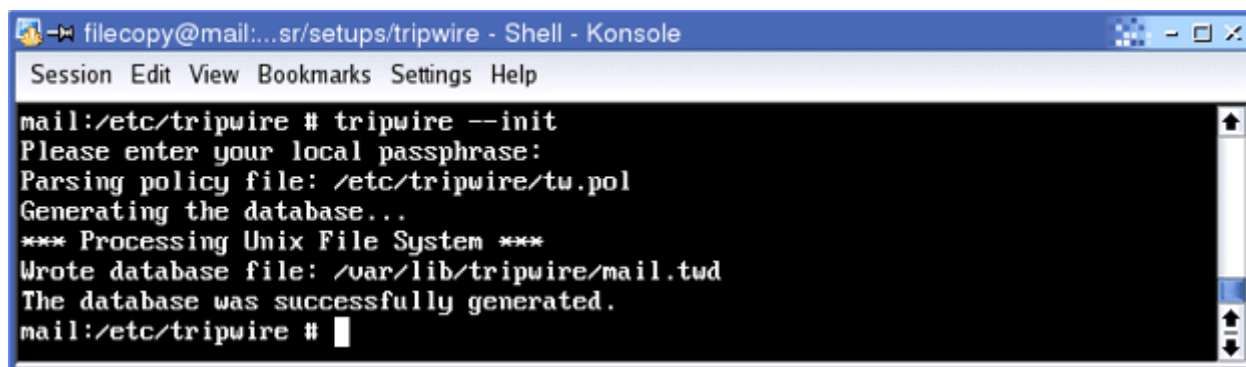
Modified:
[x] "/data/temp/test"

52,0-1 44%
```

Figure 12 – Database Violations. This figure shows a violation flagged to update the Tripwire database. If this violation is valid and should always be reported, the X should be removed.

After reviewing the changes, the file is saved, and the user is prompted for the local password to update the database. After updating, the violations that were previously reported with an X will not appear unless altered again.

There are some instances where a complete rewrite of the database is necessary, as there are too many changes in the system to accept or reject changes one by one. The process to rewrite the database is the same as that of initialization; in fact, the same command is used, as shown in the figure below.



```
filecopy@mail:...sr/setups/tripwire - Shell - Konsole
Session Edit View Bookmarks Settings Help
mail:/etc/tripwire # tripwire --init
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tu.pol
Generating the database...
*** Processing Unix File System ***
Wrote database file: /var/lib/tripwire/mail.tud
The database was successfully generated.
mail:/etc/tripwire #
```

Figure 13 – Initializing/rewriting the database. This figure shows the command (`tripwire --init`) used to take a complete snapshot of the system again. The snapshot follows the rules set forth in the Tripwire policy file.

7.0 RUNNING TRIPWIRE ON SECURE MEDIUM

Tripwire is an excellent utility to verify the integrity of all or some of the files of a system. What happens, however, if Tripwire is checking the same system on which it is currently installed? If the system could be compromised to alter other files, why not the Tripwire files? By modifying Tripwire files so that system changes to not appear, Tripwire has become useless. There is a solution however, that can guarantee the integrity of the Tripwire files, which in turn can be used to very accurately determine the integrity of the actual system.

By running Tripwire from a read-only medium such as a CD-ROM, the Tripwire files cannot be altered without authorization. If the user places key files onto a read-only device, the system snapshot (database) can be compared to the live system and report discrepancies.

In order to run Tripwire from a device such as a CD-ROM, specific files and directory structures must be copied. In all, there are eleven files to copy in a directory structure that is shown below.

```
/etc
/---/tripwire
/---/---<hostname>-local.key
/---/---site.key
/---/---tw.cfg
/---/---tw.pol
/pathfixer.sh
/twcheck.sh
/usr
/---/sbin
/---/---siggen
/---/---tripwire
/---/---twadmin
/---/---twprint
/var
/---/lib
/---/---/tripwire
/---/---/---<hostname>.twd
```

There are two files listed above that are actually scripts to alter environment variables, and execute the Tripwire checking functionality. These files should be created before they are written to CD. The first file, `pathfixer.sh`, is a brief script that adds the CD-ROM mount point to the root users' path. In the script code that follows, the mount point for the Tripwire CD-ROM is presumed to be `"/media/cdrom/tw-20050731."`

```
#!/bin/sh
# Change path to Tripwire CD
TWMNT=/media/cdrom/tw-20050731
PATH=$TWMNT/usr/sbin:$TWMNT/usr/bin:$PATH
export PATH
export TWMNT
```

This code should be run prior to the Tripwire check script, by running `'source pathfixer.sh'` at the prompt. As shown, this script adds the CD-ROM to the users path, so that when tripwire is called from the `twcheck.sh` script, the correct CD structure is used.

The second script, `twcheck.sh`, actually performs the Tripwire integrity check. Because all of the Tripwire files are on the CD-ROM, several options are called to specify the location of the database, keys and policy. This code is run by executing `'./twcheck.sh'` from the prompt.

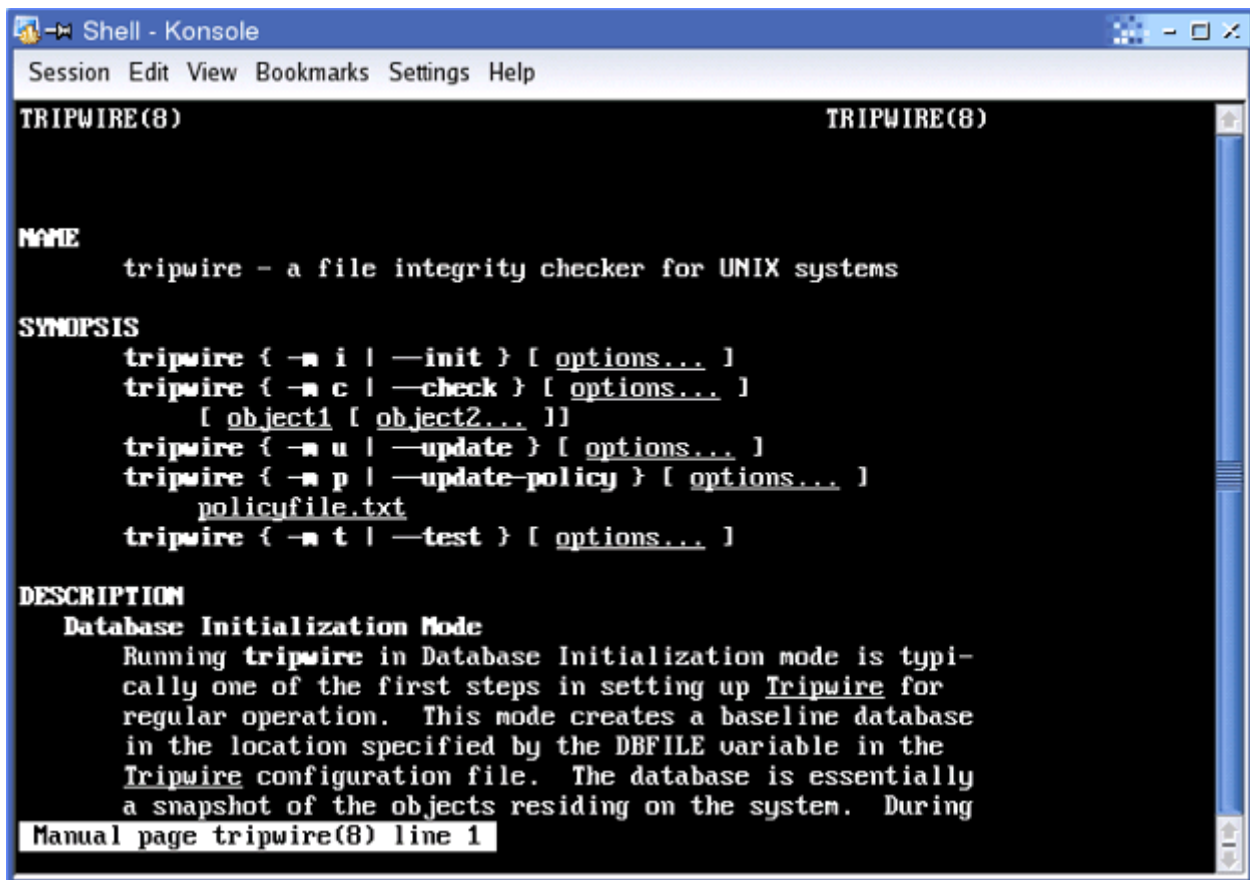
```
#!/bin/sh
chmod +x $TWMNT/usr/sbin/tripwire
chmod +x $TWMNT/usr/sbin/siggen
chmod +x $TWMNT/usr/sbin/twadmin
chmod +x $TWMNT/usr/sbin/twprint
$TWMNT/usr/sbin/tripwire \
  --check \
  --polfile $TWMNT/etc/tripwire/site.key \
  --site-keyfile $TWMNT/etc/tripwire/site.key \
  --local-keyfile $TWMNT/etc/tripwire/<hostname>-local.key \
  --dbfile $PREFIX/var/lib/tripwire/<hostname>.twd \
  --cfgfile /etc/tripwire/tw.cfg \
  --email-report
```

This command specifies the location of the Tripwire files on the CD-ROM, as opposed to using the files on the system. This will ensure that the files have not been tampered with, and that the Tripwire check is valid. The file path was changed using the pathfixer script, so that the files are referenced on the CD, such as `‘/media/cdrom/tw-20050731/etc/tripwire/site.key.’`

While placing the Tripwire files on read-only medium is not required, it makes significant strides to make the Tripwire integrity check even more secure. Using MD5 checksums takes integrity checking a step further, so that the user can check the MD5sum of the CD-ROM files to ensure that they have not been altered. Also, with the increase in the use of USB storage devices, these small devices could be used for Tripwire. By setting the physical lock on the device, data cannot be written to the drive, but provides the same functionality as that of the read-only CD-ROM.

8.0 TRIPWIRE OPTIONS & REFERENCE MATERIALS

Whether running Tripwire for the first time, or for an on-demand check, users often need to know the options available to them. This document has covered a large portion of the commands Tripwire uses to create, update, and check the integrity of a system. There are, however, other options and sub-options that can be used for specific instances. To cover those additional options would be a restatement of existing documentation. An excellent source for this existing information regarding options, command syntax, and help is available to the user on the live system in the form of 'man pages.' These 'man pages' should be considered a primary resource of information for Tripwire. To view the 'man pages' execute the command 'man tripwire.' This will return the documentation of Tripwire in great detail, explaining options and capabilities. The first screen of the Tripwire man page is shown in Figure 14 below. Other commands used in this document were 'twadmin' and 'twprint.' These two commands also have their own 'man pages,' viewable by executing 'man twadmin' and 'man twprint,' respectively. Other 'man page' documents are found with under other Tripwire commands, such as twconfig, twpolicy, twfiles, siggen, and twintro (e.g. man twconfig).



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
TRIPWIRE(8)                                TRIPWIRE(8)

NAME
tripwire - a file integrity checker for UNIX systems

SYNOPSIS
tripwire { -i | --init } [ options... ]
tripwire { -c | --check } [ options... ]
    [ object1 [ object2... ] ]
tripwire { -u | --update } [ options... ]
tripwire { -p | --update-policy } [ options... ]
    policyfile.txt
tripwire { -t | --test } [ options... ]

DESCRIPTION
Database Initialization Mode
Running tripwire in Database Initialization mode is typically one of the first steps in setting up Tripwire for regular operation. This mode creates a baseline database in the location specified by the DBFILE variable in the Tripwire configuration file. The database is essentially a snapshot of the objects residing on the system. During
Manual page tripwire(8) line 1
```

Figure 14 – Tripwire man page. The figure shows the first page of the Tripwire manual that can be used for reference.