# Tripwire – An Integrity Assessment Tool

Paula McKeehan

January 24, 2001

## Overview

In your effort to protect your system against hackers, it is critical that you have a comprehensive picture of your entire system. Tripwire is not designed to keep out intruders, however it does provide an inventory of what files may have been tampered with. "System files; such as the kernel, library files, OS binaries, and configuration files; are prime targets for attackers." Tripwire is a tool designed to provide that initial snapshot. Gene H. Kim and Eugene H. Spafford developed tripwire while working at the COAST Laboratory at Purdue University in 1992. They wanted to create a tool that would monitor a designated set of files and directories for any changes. When used on a regular basis, Tripwire will let a system administrator know if any files have been modified or corrupted. In 1997, Gene Kim and W. Wyatt Starnes founded Tripwire Inc. One of their goals was to release an upgraded commercial version that could be used on more platforms. The non-commercial version 1.2 Tripwire package is available at ftp://coast.cs.purdue.edu/pub/tools/unix/ids/tripwire. Version 1.2, however, is out-dated and no longer supported. The commercial release of Tripwire 2.2.1 is available at http://www.tripwiresecurity.com for $495 per host. There is also a Tripwire 1.3.1 Academic Source Release available at http://www.tripwiresecurity.com/downloads/.

At the time Tripwire was first developed in 1992, there were only a handful of UNIX security tools available:

COPS (Computer Oracle and Password System) are a collection of security tools that was designed to support a large number of UNIX platforms. It has been distributed freely since 1989, and monitors files using a CRC checksum. However, the COPS program had several problems, such as its inability to monitor all fields in the file inode structure.

TAMU is a set of scripts that scan a UNIX system looking for security problems, in the same manner as COPS. "TAMU is shipped with a database of signatures for system binaries of popular operating systems. TAMU compares signatures of critical system files against those stored in its database to determine whether they match any of the known versions."(1) The TAMU scripts were unable to scan an entire file system and required that its database be updated with every new release of the operating system and patches.

Hobgoblin checks file system consistency against a template description. When Hobgoblin runs a scan it compares the system files against the descriptions stored in its database. However, there is no measure for updating the database when file systems change.

ATP takes a snapshot of the system and creates a database of file attributes. It verifies files using a 32-bit CRC and a MD5 checksum. ATP was also developed with an action list that would automatically change the ownership to root of any changed files it's detected. This particular feature made ATP more unique when compared against COPS, TAMU, and Hobgoblin, since they merely reported potential problems.

However, most of these tools did not provide the capabilities or portability that was desired in an integrity checker.

**Configuring Tripwire**

The Tripwire package reads a configuration file that lists all of the files and directories you want to monitor. Both the free and commercial versions come with an initial configuration file for your use. The freeware configuration file is called tw.config, and the commercial version is twpol.txt. However, do not depend solely on the contents of that file, because it is rather generic to each operating system, and may not be all-inclusive depending on the release level of your operating system. For example, it is important that you monitor all files that are SUID or SGID on your system. To locate all SUID files on your system use the following command:

```
find / -type f -perm -4000 -print
```

To locate all SGID files on your system use the following command:

```
find / -type f -perm -2000 -print
```

You can now take the results from both of these commands and place them in your configuration/policy file. It is also important to know what startup files are located in the home directory of your users. It's a good idea to do a search for the following files and include them in your configuration/policy file:

| **Name** | **Description** |
|---|---|
| .cshrc | C-shell initialization file |
| .forward | Tells /usr/lib/sendmail where to forward your electronic mail |
| .kshrc | Korn shell initialization file |
| .login | C-shell initialization command script, which executes at login |
| .netrc | FTP initialization and macros; can also store clear text passwords |
| .profile | Bourne and Korn shell initialization commands |
| .rhosts | Contains the names of remote accounts that can log in using rsh and rlogin without a password |
| .xinitrc | Start-up file for xinit |

For example, to locate all .netrc files on your system, use the following command:

```
find / -name '.netrc' -print
```

Again, after locating all of the important dot files in your user's home accounts, place them in your configuration/policy file for monitoring. It would be a good idea at this time to also make sure there is a business need for some of these files, such as .rhosts and .netrc. If there is a valid need, then make sure the permissions are also set to 600 to prevent anyone but the owner of these files access.

"A signature function is any function that takes an arbitrary file as input and yields a fixed-sized output called the signature or the secure hash."(5) However, it is possible for an intruder to break a signature for a given file whereas the file size remains the same. With this in mind, it became necessary to add signature routines to the integrity checker software. Message digest signature routines are setup to use one-way functions that are difficult to reverse. The freeware version of Tripwire supports the following signature routines:
MD5 - the RSA Data Security, Inc. Message Algorithm that generates a 128-bit signature.
Snefru - the Xerox Secure Hash Function
MD4 - the RSA Data Security, Inc. Message Algorithm that was designed to exploit 32-bit RISC architectures.
MD2 - the RSA Data Security, Inc. MD2 Message Algorithm that generates a 128-bit signature.
SHA - the Secure Hash Algorithm.
Haval - a 128-bit signature message-digest algorithm.
CRC-16 - a 16-bit Cyclic Redundany Checkcode
CRC-32 - a 32-bit Cyclic Redundany Checkcode

The commercial version of Tripwire supports the following signature routines:
CRC-32 - a 32-bit Cyclic Redundany Checkcode
SHA - the Secure Hash Algorithm.
Haval - a 128-bit signature message-digest algorithm.
MD5 - the RSA Data Security, Inc. Message Algorithm that generates a 128-bit signature.

The following is a portion of a sample policy file for Tripwire version 2.2.1:

```
# Critical configuration files
(
#     rulename = "Critical configuration files",
      severity = 100
)
{
/etc                          -> $(ReadOnly) (rulename="/etc - critical config files")  ;

/etc/default                  -> $(ReadOnly) (rulename="/etc/default - critical config files")
;
/etc/dfs/dfstab               -> $(ReadOnly) (rulename="/etc/dfs/dfstab - critical config
files")  ;
/etc/dfs/sharetab             -> $(ReadOnly) (rulename="/etc/dfs/sharetab - critical config
files")  ;
/etc/hosts                    -> $(ReadOnly) (rulename="/etc/hosts - critical config files")  ;
/etc/inet/inetd.conf          -> $(ReadOnly) (rulename="/etc/inet/inetd.conf - critical
config files")  ;
```

```
/etc/inet/protocols          -> $(ReadOnly) (rulename="/etc/inet/protocols - critical config
files") ;
/etc/inet/services           -> $(ReadOnly) (rulename="/etc/inet/services - critical config
files") ;
```

Lets breakdown the line for /etc in the above policy file example. The "ReadOnly" variable is predefined within the software. This particular variable translates to the following information:

**Check the following file properties:**
File permissions
Inode number
Number of links
User id of owner
Group id of owner
File size
File type
Modification time stamp
Device number of the disk where the inode is stored
Number of blocks allocated
CRC-32 signature
MD5 signature

**Ignore the following file properties :**
Device number of devices
Access time stamp
Inode creation/modification time stamp
Growing files
SHA signature
Haval signature

The rulename attribute is a symbolic name that can be unique for each directory or file on your system. The rulename information is displayed as a description in the report file.

After completing the customization of your configuration file, you are ready to create the system database that will store your baseline information. To initialize the database on release 1.3.1 use the command 'tripwire -initialize', whereas the commercial version uses './tripwire --init'. These commands will read the configuration/policy file and generate a database based on its contents.

At this point, Tripwire is now ready to detect any changes made to files. To run release 1.3.1 in integrity checking mode, use the command 'tripwire'. In release 2.2.1, use the command './tripwire --check --interactive', which will bring the resulting report into an editor, such as vi, for performing database updates. "Tripwire detects intrusions by watching for changes in the filesystem. It also lets you know exactly what has changed, so you know which programs you can trust once you have been hacked."(2) However, some files changes can be legitimate due to normal system operations, such as system log files. The administrator should confirm the change was valid and then update the database.

Finally, after completing your installation, configuration file setup, and database initialization, it is important to protect your Tripwire binaries and database files. Make sure they are in a protected directory by removing all unnecessary permissions to your Tripwire directories, or by placing the software and data on removable disks. If an intruder can change your database then your entire integrity checker will be rendered useless.

**Differences Between Commercial and ASR release** (6)

| **Tripwire 2.2.1 Commercial Release** | **Tripwire 1.3.1 Academic Source Release** |
|---|---|
| Workstation or site license fees | Free license per workstation |
| • Email tech support included with license fee.<br>• Extended support option available: Four business-hour turnaround on email, same version software upgrades, bug fixes, access to knowledge base.<br>• Telephone support option available. | Support limited to on-line knowledge base and long-term turn around email support. |
| • Fast, easy, simple installation.<br>• Configuration information easily changed with Tripwire commands. | • Complex installation<br>• User compiled<br>• Extensive installation configuration |
| • User manual<br>• "man" page documentation | "man" page documentation only |
| • Optimized for performance.<br>• Approximately 50% better performance than ASR version. | Resource use intensive. |
| • Critical files in binary format and cryptographically signed.<br>• Safely stored on target machine.<br>• Facilitates automated operation. | • Critical Tripwire files stored in plain-text format.<br>• Requires use of read-only media for database and configuration files. |
| • Email reporting<br>• Reports are organized by user-defined rule names and severity levels. | • No email reporting.<br>• On-host text-based reports only. |

Clearly, Tripwire is a necessary tool for any system administrator who has one system or many to monitor. The interactive update mode allows the system administrator to easily update the database and ensures that no files are inadvertently updated without review.(1) The software has been tested on more than 28 UNIX platforms, thus proving it highly portable. In addition, the commercial version has been created for the Windows NT and Windows 2000 platforms.

**References:**

1. Kim, Gene H. and Spafford, Eugene H. "The Design and Implementation of Tripwire: A File System Integrity Checker." 23 February 1995.
URL: http://sunsite.bilkent.edu.tr/pub/security/cerias/papers/Tripwire

2. Beale, Jay. “Tripwire – The Only Way to Really Know.” 17 October 2000.
URL: http://www.earthweb.com/dlink.resource-jhtml.72.1084.|repository||networking|content|article|2000|10|11|NTSPTripwire|NTSPTripwire~xml.0.jhtml?cda=true

3. Rogers, Russ. “Understanding and Installing Tripwire.” 15 August 2000.
URL: http://www.securityhorizon.com/whitepapers/tripwire.html

4. Fennelly, Carole. “Tripwire: The Next Generation of Security Tools.” 29 February 2000.
URL: http://www.unixinsider.com/swol-02-2000/swol-02-security.html

5. Kim, Gene and Spafford, Gene. “Monitoring File System Integrity with Tripwire.” July 1992.
http://www.cerias.purdue.edu/homes/spaf/tech-reps/gkim

6. Garfinkel, Simson and Spafford, Gene. Practical UNIX & Internet Security. April 1996.

7. Tripwire, Inc. Tripwire 2.2.1 for UNIX. January 2000.