

NAME

tripwire – a file integrity checker for UNIX systems

SYNOPSIS

```
tripwire { -m i | --init } [ options... ]
tripwire { -m c | --check } [ options... ] [ object1 [ object2... ] ]
tripwire { -m u | --update } [ options... ]
tripwire { -m p | --update-policy } [ options... ] policyfile.txt
tripwire { -m 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 later *Tripwire* integrity checks, this database serves as the basis for comparison.

When run in Database Initialization mode, **tripwire** reads the policy file, generates a database based on its contents, and then cryptographically signs the resulting database. Options can be entered on the command line to specify which policy, configuration, and key files are used to create the database. The filename for the database can be specified as well. If no options are specified, the default values from the current configuration file are used.

Integrity Checking Mode

After building the *Tripwire* database, the next step is typically to run **tripwire** in Integrity Checking mode. This mode scans the system for violations, as specified in the policy file. Using the policy file rules, *Tripwire* will compare the state of the current file system against the initial baseline database. An integrity checking report is printed to *stdout* and is saved in the location specified by the `REPORTFILE` setting in the *Tripwire* configuration file.

The generated report describes each policy file violation in detail, depending on whether the specified file system object was added, deleted, or changed. Each report item lists the properties of the object as it currently resides on the file system, and, if appropriate, the old value stored in the database. If there are differences between the database and the current system, the administrator can either fix the problem by replacing the current file with the correct file (e.g., an intruder replaced `/bin/login`), or update the database to reflect the new file (e.g., a fellow system administrator installed a new version of `/usr/local/bin/emacs`). The (**-I** or **--interactive**) option launches an editor that allows the user to update the database quickly. The Database Update mode of **tripwire** can also be used.

Database Update Mode

Running **tripwire** in Database Update mode allows any differences between the database and the current system to be reconciled. This will prevent the violation from showing up in future reports. If the reported change is unexpected and potentially malicious, then the changed file should be replaced with the original version. If there is a valid reason for the change, the database must be changed to match the current files.

In Database Update mode, the items to be changed are specified in a "ballot box" in the plain text report that is launched in an editor program. The entries to be updated are specified by leaving the "x" next to each policy violation. After the user exits the editor and provides the correct local passphrase, **tripwire** will update the database. Options to control this operation include the (**-Z** or **--secure-mode**) and (**-a** or **--accept-all**) flags.

Policy Update Mode

Policy update mode is used by **tripwire** to change or update the policy file and to synchronize an earlier database with new policy file information. The filename of the new clear text version of the policy file is specified on the command line. The new policy file is compared to the existing version, and the database is updated according to the new policy rules. Any changes in the database since the last integrity check will be detected and reported. How these violations are interpreted depends on the security mode specified with the (**-Z** or **--secure-mode**) option. In **high** security mode (the default), *Tripwire* will print a list of violations and exit without making changes to the database. In **low** security mode, the violations are still report-

ed, but changes to the database are made automatically.

Because the policy and database files are binary-encoded and cryptographically signed, the user will be prompted for the site and local passphrases to change the policy settings. After the database is successfully updated, the database and policy files are re-encoded and signed.

Test Mode

Test mode is used to check the operation of the *Tripwire* email notification system. When run in this mode, *Tripwire* will use the email notification settings specified in the configuration file to send a test email message. If MAILMETHOD is set to SMTP, the SMTPHOST and SMTPPORT values will be used to send email. If MAILMETHOD is set to SENDMAIL, the MAILPROGRAM value will be used. If email notification is working correctly, the address specified on the command line will receive the following message:

```
To: user@domain.com
From: user <user@domain.com>
Subject: Test email message from Tripwire
```

```
If you receive this message, email notification
from Tripwire is working correctly.
```

Test mode only tests email notification for the address specified on the command-line, and does not check for errors in the syntax used with the emailto attribute in the policy file.

OPTIONS

Database Initialization mode:

-m i	--init
-v	--verbose
-s	--silent, --quiet
-c <i>cfgfile</i>	--cfgfile <i>cfgfile</i>
-p <i>polfile</i>	--polfile <i>polfile</i>
-d <i>database</i>	--dbfile <i>database</i>
-S <i>sitekey</i>	--site-keyfile <i>sitekey</i>
-L <i>localkey</i>	--local-keyfile <i>localkey</i>
-P <i>passphrase</i>	--local-passphrase <i>passphrase</i>
-e	--no-encryption

-m i, --init

Mode selector.

-v, --verbose

Verbose output mode. Mutually exclusive with (-s).

-s, --silent, --quiet

Silent output mode. Mutually exclusive with (-v).

-c *cfgfile*, --cfgfile *cfgfile*

Use the specified configuration file.

-p *polfile*, --polfile *polfile*

Use the specified policy file.

-d *database*, --dbfile *database*

Write to the specified database file.

-S *sitekey*, --site-keyfile *sitekey*

Use the specified site key file to read the configuration and policy files.

-L *localkey*, --local-keyfile *localkey*

Use the specified local key file to write the new database file. Mutually exclusive with (-e).

-P *passphrase*, --local-passphrase *passphrase*

Specifies passphrase to be used with local key to sign the new database. Mutually exclusive with (-e).

-e, --no-encryption

Do not sign the database being stored. The database file will still be compressed and will not be human-readable. Mutually exclusive with **(-L)** and **(-P)**.

Integrity Checking mode:

-m c	--check
-I	--interactive
-v	--verbose
-s	--silent, --quiet
-c <i>cfgfile</i>	--cfgfile <i>cfgfile</i>
-p <i>polfile</i>	--polfile <i>polfile</i>
-d <i>database</i>	--dbfile <i>database</i>
-r <i>report</i>	--twrfile <i>report</i>
-S <i>sitekey</i>	--site-keyfile <i>sitekey</i>
-L <i>localkey</i>	--local-keyfile <i>localkey</i>
-P <i>passphrase</i>	--local-passphrase <i>passphrase</i>
-n	--no-ty-output
-V <i>editor</i>	--visual <i>editor</i>
-E	--signed-report
-i <i>list</i>	--ignore <i>list</i>
-l { <i>level</i> <i>name</i> }	--severity { <i>level</i> <i>name</i> }
-R <i>rule</i>	--rule-name <i>rule</i>
-x <i>section</i>	--section <i>section</i>
-M	--email-report
-t { 0 1 2 3 4 }	--email-report-level { 0 1 2 3 4 }
[<i>object1</i> [<i>object2...</i>]]	

-m c, --check

Mode selector.

-I, --interactive

At the end of integrity checking, the resulting report is opened in an editor where database updates can be easily specified using the ballot boxes included in the report.

-v, --verbose

Verbose output mode. Mutually exclusive with **(-s)**.

-s, --silent, --quiet

Silent output mode. Mutually exclusive with **(-v)**.

-c *cfgfile*, --cfgfile *cfgfile*

Use the specified configuration file.

-p *polfile*, --polfile *polfile*

Use the specified policy file.

-d *database*, --dbfile *database*

Use the specified database file.

-r *report*, --twrfile *report*

Write the specified report file.

-S *sitekey*, --site-keyfile *sitekey*

Use the specified site key file to read the configuration and policy files.

-L *localkey*, --local-keyfile *localkey*

Use the specified local key file to read the database file and, if **(-E)** is specified, to write the report file.

- P** *passphrase*, **--local-passphrase** *passphrase*
 Specifies passphrase to be used with local key to sign the database when (**-I**) is used, and to sign the report when (**-E**) is used. Valid only with (**-I**) or (**-E**).
- n**, **--no-tty-output**
 Suppress the report from being printed at the console.
- V** *editor*, **--visual** *editor*
 Use the specified editor to edit the update ballot boxes. Meaningful only with (**-I**).
- E**, **--signed-report**
 Specifies that the *Tripwire* report will be signed. If no passphrase is specified on the command line, **tripwire** will prompt for the local passphrase.
- i** *list*, **--ignore** *list*
 Do not compute or compare the properties specified in *list*. Any of the letter codes (abcdgimnprstulCHMS) specified in *propertymasks* can be excluded. Use of this option overrides information from the policy file. The format to be used for *list* is a double-quoted, comma-delimited list of properties (e.g. `--ignore "p,c,m"`).
- l** { *level* | *name* }, **--severity** { *level* | *name* }
 Check only policy rules with severity greater than or equal to the given level. The level may be specified as a number or as a name. Severity names are defined as follows:
- | | |
|--------|-----|
| Low | 33 |
| Medium | 66 |
| High | 100 |
- Mutually exclusive with (**-R**).
- R** *rule*, **--rule-name** *rule*
 Check only the specified policy rule. Mutually exclusive with (**-I**).
- x** *section*, **--section** *section*
 Only check the rules in the specified section of the policy file. For *Tripwire 2.3.1*, FS is the only meaningful argument for this flag.
- M**, **--email-report**
 Specifies that reports be emailed to the recipient(s) designated in the policy file.
- t** *level*, **--email-report-level** *level*
 Specifies the detail level of email reports, overriding the EMAILREPORTLEVEL variable in the configuration file. *level* must be a number from 0 to 4. Valid only with (**-M**).
- [*object1* [*object2*...]]
 List of files and directories that should be integrity checked. Default is all files. If files are specified for checking, the **--severity** and **--rule-name** options will be ignored.

Database Update mode:

-m u	--update
-v	--verbose
-s	--silent, --quiet
-c <i>cfgfile</i>	--cfgfile <i>cfgfile</i>
-p <i>polfile</i>	--polfile <i>polfile</i>
-d <i>database</i>	--dbfile <i>database</i>
-r <i>report</i>	--twrfile <i>report</i>
-S <i>sitekey</i>	--site-keyfile <i>sitekey</i>
-L <i>localkey</i>	--local-keyfile <i>localkey</i>
-P <i>passphrase</i>	--local-passphrase <i>passphrase</i>
-V <i>editor</i>	--visual <i>editor</i>

- P** *passphrase* **--local-passphrase** *passphrase*
- Q** *passphrase* **--site-passphrase** *passphrase*
- Z** { low | high } **--secure-mode** { low | high }
- policyfile.txt*
- m p, --update-policy**
Mode selector.
- v, --verbose**
Verbose output mode. Mutually exclusive with (-s).
- s, --silent, --quiet**
Silent output mode. Mutually exclusive with (-v).
- c** *cfgfile*, **--cfgfile** *cfgfile*
Use the specified configuration file.
- p** *polfile*, **--polfile** *polfile*
Write the specified policy file.
- d** *database*, **--dbfile** *database*
Use the specified database file.
- S** *sitekey*, **--site-keyfile** *sitekey*
Use the specified site key file to read the configuration file, and read and write the policy file.
- L** *localkey*, **--local-keyfile** *localkey*
Use the specified local key file to read and write the database file.
- P** *passphrase*, **--local-passphrase** *passphrase*
Specifies passphrase to be used with local key to sign the database.
- Q** *passphrase*, **--site-passphrase** *passphrase*
Specifies passphrase to be used with site key to sign the new policy file.
- Z** { low | high }, **--secure-mode** { low | high }
Specifies the security level, which affects how certain conditions are handled when the existing filesystem does not match the database information. Since the database produced at the end of a policy update becomes the baseline for future integrity checks, this consistency-checking ensures that no substantive filesystem changes have occurred since the last integrity check.

High: In **high** security mode, if a file on the filesystem does not match the properties in the database file, Tripwire reports the differences as warnings, and exits without changing the database or the policy file.

Low: In **low** security mode, inconsistencies are reported as warnings, but the changes are still made to the database and policy file.
- policyfile.txt*
Specifies the text policy file that will become the new policy file.

Test mode:

- m t** **--test**
- e** *user@domain.com* **--email** *user@domain.com*
- m t, --test**
Mode selector.
- e** *user@domain.com*, **--email** *user@domain.com*
Use the specified email address. This parameter must be supplied when test mode is used. Only one address may be specified.

VERSION INFORMATION

This man page describes **tripwire** version 2.3.1

AUTHORS

Tripwire, Inc.

COPYING PERMISSIONS

Permission is granted to make and distribute verbatim copies of this man page provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this man page under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this man page into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by Tripwire, Inc.

Copyright 2000 Tripwire, Inc. Tripwire is a registered trademark of Tripwire, Inc. in the United States and other countries. All rights reserved.

SEE ALSO

twintro(8), twadmin(8), twprint(8), siggen(8), twconfig(4), twpolicy(4), twfiles(5)

The Design and Implementation of Tripwire: A UNIX File Integrity Checker by Gene Kim and Eugene Spafford. Purdue Technical Report CSD-TR-93-071.