About This Guide
Document List

The Tripwire Installation Guide describes installation procedures for Tripwire Manager and Tripwire for Servers software.

The Tripwire for Servers User Guide describes configuration and operation of Tripwire for Servers software.

The Tripwire Manager User Guide describes configuration and operation of Tripwire Manager software, which is used to manage multiple installations of Tripwire for Servers software.

The Tripwire Reference Guide contains detailed information about the Tripwire configuration and policy files.

The Quick Reference Cards summarize important functionality of Tripwire for Servers software.

You can access PDF versions of the Guides from the docs directories on the Tripwire Manager and Tripwire for Servers CDs.

You can access online help from the Tripwire Manager interface.
Conventions

This Guide uses the following typographic conventions.

Bold in regular text indicates FTP and HTTP URLs, and emphasizes important issues.

Italic indicates file and directory names.

Constant in regular text shows commands and command-line options, and policy file rule attributes, directives, and variables.

Sans Serif in examples shows actual user input on the command line.

Sans Serif Italic in examples shows variables which should be replaced with context-specific values.

W denotes sections of the text that apply only to Windows installations of Tripwire software. Unless otherwise specified, all references to Windows refer to both Windows NT and Windows 2000.

U denotes sections of the text that apply only to UNIX or Linux installations of Tripwire software. Unless otherwise specified, all references to UNIX also refer to Linux.

[options] the command reference section shows optional command-line arguments in brackets.

{ 1 | 2 | 3 } the command reference section shows sets of possible options in braces, separated by the | character. Choose only one of the options.

Unless otherwise specified, command-line examples assume that the Tripwire bin directory is the current working directory.
Support

For the latest information and support for Tripwire products, visit the Tripwire website or contact Tripwire Technical Support.

Tripwire Support Website: http://www.tripwire.com/support

Tripwire Technical Support:
  e-mail: support@tripwire.com
  toll-free: 1.866.TWSUPPORT (6am-6pm Pacific)
  phone: 503.276.7663

General information: info@tripwire.com

Tripwire Professional Services

Tripwire Professional Services provides flexible service and support to meet your specific technical and deployment needs. If you would like Tripwire software deployment and implementation assistance, or additional training in using Tripwire software products, visit http://www.tripwire.com or contact your Tripwire Sales Representative.

Tripwire Educational Services

Obtain expert hands-on technical training and experience from a Tripwire Certified Instructor. Courses are offered by Tripwire Authorized Training Centers, and prepare you to install, configure, and maintain Tripwire software. Visit http://www.tripwire.com or contact your Tripwire Sales Representative for more information.
Contents

About This Guide ........................................ iii
Document List ............................................. v
Conventions .............................................. vi
Support .................................................. vii
   Tripwire Professional Services .................... vii
   Tripwire Educational Services .................... vii

Introduction to Tripwire for Servers ............... 1
Overview .................................................. 3
How Tripwire for Servers Works ..................... 3
Data and Network Integrity with Tripwire Software .... 4
   Security ............................................. 4
   System Management ................................. 5
   Risk Management .................................. 5
Using Tripwire Software ................................ 6
Tripwire Files ........................................... 8
Key Files and Passphrases ............................ 9
Tripwire Manager ....................................... 10
Changes for This Version ............................. 10

Configuring Tripwire for Servers .................... 13
Overview ............................................... 15
Editing the Configuration File ....................... 15
   Setting Up E-mail Reporting ...................... 17
## Contents

- Setting Up Log File Reporting .......................... 17
- Setting Up SNMP Logging ............................... 18
- Testing E-mail Reporting ............................... 19
- Creating the Policy File ................................. 20
  - Obtaining a Policy File ............................... 20
    - Tripwire Policy Resource Center Website .... 21
    - Tripwire for Servers CD ............................ 21
  - Editing the Policy File ............................... 22
  - Signing and Installing the Policy File .......... 23
- Initializing the Database File ......................... 23
- Tuning the Policy File ................................ 24

### Using Tripwire for Servers ......................... 29

#### Overview ........................................... 31

#### Checking Integrity ................................ 31
  - E-mailing Integrity Check Reports .................. 33
  - Selective Integrity Checks ........................... 34
  - Scheduling Integrity Checks ......................... 38

#### Viewing Report Files ............................... 39

#### Updating the Database File ......................... 40
  - Resolving Database Update Problems ............... 42

#### Updating the Policy File ........................... 43
  - The Policy Update Process ........................... 44
  - Resolving Policy Update Problems ................. 45

#### Changing Passphrases ............................... 46
Introduction to Tripwire for Servers
Overview

This chapter introduces Tripwire for Servers, an integrity assessment tool that allows you to monitor data and network integrity. If you are new to Tripwire products or to the concepts of data and network integrity, this chapter gives you the necessary background.

If you have previous experience with Tripwire software, read about the new features in this release before moving on to the next chapter.

This chapter describes:

- how Tripwire for Servers works
- how you can use Tripwire for Servers to maintain Data and Network Integrity
- the components of Tripwire software
- cryptographic protection for Tripwire files
- new features for this version of Tripwire for Servers
- Tripwire Manager, an application for managing multiple installations of Tripwire for Servers

How Tripwire for Servers Works

Tripwire for Servers tells you how your system has changed from a known, good state. It does this by first scanning your file system, based upon pre-established rules, and creating a baseline. Once this baseline is created, you can run Tripwire for Servers periodically to determine how the system has changed. If changes are detected, Tripwire for Servers generates a report of the changes, and can send alerts via email, syslog, or SNMP.
Tripwire for Servers uses a user-defined policy, which specifies the objects in a system, and the attributes of those objects, to check. The policy can be tuned to eliminate the noise of day-to-day system changes due to normal operation, and only report significant, actionable events. The policy also characterizes objects or groups of objects according to function and relative severity. When Tripwire software finds multiple integrity violations, it sorts the output based upon the criteria you define, allowing you to easily see and address the most serious issues first.

Tripwire for Servers includes a comprehensive policy for each operating system that it is supported on. These policies can be used out of the box or customized by the user.

Tripwire for Servers includes software that enables it to connect to one or more Tripwire Managers. Tripwire Manager allows you to manage and view reports from thousands of Tripwire for Servers machines across your network from a GUI-based console. See page 10 for more information on Tripwire Manager.

**Data and Network Integrity with Tripwire Software**

Effective security, system management, and risk management depend upon the ability to assess the state of Data and Network Integrity (DNI).

**Security**

Many malicious intrusions involve changes to critical infrastructure components, when intruders replace or modify system files to gain control of systems. Tripwire for Servers detects intrusions and saves administrators vast amounts of recovery time by quickly showing which components have changed.
Detecting intrusions in this way has several advantages. First, Tripwire software detects misuse whether it comes through the firewall or originates inside it. Second, Tripwire software does not rely upon attack signatures, which are based on historical attacks and cannot detect constantly-evolving methods. Third, reports from Tripwire software can be used as a forensics tool to establish a chain of evidence when prosecuting miscreants.

Tripwire software is not meant to replace other security measures such as firewalls or network intrusion detection tools. Instead, it is an integral part of a comprehensive security strategy. Tripwire compliments other security tools and protects the platforms on which they run.

### System Management

Tripwire software is often used for day-to-day management of an information infrastructure. When important applications and services cease to function properly, the first, most fundamental question is: what changed? Answering that question quickly and easily frees administrators from time-consuming and tedious diagnosis and recovery tasks.

In addition to detecting unwanted changes, Tripwire software can also be used to detect when things should change, such as when installing patches to the operating system, or during new software installations.

### Risk Management

To resist misuse of your information infrastructure, you must ensure that security measures do not themselves become altered. You can use Tripwire software to monitor the integrity of firewalls and network security appliances, as well as the platforms on which they run.
Using Tripwire Software

The chart on the next page outlines the process of integrity assessment with Tripwire for Servers. You can perform all of these tasks from the command line on each Tripwire for Servers machine, or from a central location with the Tripwire Manager (see page 10).

1. Edit the rules in the default policy file, or create a custom policy file for your system, to specify the directories, files, or registry objects that you want Tripwire software to monitor.

2. Using the rules in the policy file, Tripwire software collects data from the file system and generates a database file. For most Tripwire implementations, this step only needs to be done once, when the software is first installed.

3. Tune the rules in your policy file to remove noise and false positives from Tripwire report files.

4. After tuning the policy file, you can run regular integrity checks. During a check, Tripwire software compares the data in the database file to the current state of the system and creates a report of changes. This report can be sent to recipients via e-mail.

5. Using the information in the report file, you can decide if changes to the system are authorized.

6. If you discover unauthorized changes, you should take appropriate measures, including restoring files from backup, or changing security procedures to prevent further intrusions.

7. If you discover authorized changes, you should update the database file to reflect the changed state of the system. This prevents these changes from being flagged as violations in the future.

   After you resolve all of the changes, you can run another integrity check to verify the integrity of the system.

8. After an integrity check, you may want to update the existing policy file to monitor new files, or to change rules that are generating noise in Tripwire report files.
1. Install software & create policy file

2. Initialize database file

3. Tune policy file

4. Run integrity check

5. Examine report file

6. Take appropriate security measures

7. Update database file

8. Update policy file
Tripwire for Servers uses a number of files to assess system security:

The **policy file** enables you to specify how Tripwire software monitors your system. The policy file consists of a list of rules which specify system objects (directories, files, or registry objects) to monitor, and describe which changes to the objects should be reported and which ones can be ignored.

The **database file** is at the center of integrity assessment. When Tripwire software is first installed, it uses the rules in the policy file to create a snapshot of your computer system in a known secure state. During an integrity check, the software compares this baseline database file against the current state of the system to determine if any changes have occurred.

**Report files** record the changes detected during an integrity check that violate the rules in the policy file. You can configure Tripwire software to e-mail all or part of a report file to administrators after an integrity check.

The **configuration file** stores system-specific information that controls Tripwire operation, including the location of Tripwire files, and the parameters used for e-mail notification.

The **site key file** and **local key file** store public and private keys used to sign Tripwire files cryptographically. To modify signed Tripwire files, you must provide the correct site or local passphrase. See the next section for more information on key files and passphrases.

The **Agent configuration file** stores information that each machine uses to communicate with the Tripwire Manager (see page 10).
Key Files and Passphrases

To protect against unauthorized modification, all important Tripwire files are stored on disk in a binary-encoded and signed form. Tripwire policy, configuration, database, and (optionally) report files are protected with El Gamal asymmetric cryptography with a 1024 bit signature.

The El Gamal signature process uses a paired set of keys, one public key and one private key. In Tripwire software's cryptographic system, the public and private keys are generated and stored together in a key file.

Two of these sets of keys, the site key file and the local key file, are used to protect important files. The site key is used to protect the policy and configuration files, which can be used across an entire site. The local key is used to protect database and (optionally) report files, which are specific to a particular system.

To edit or replace a signed Tripwire file, you must provide the passphrase for the key file used to sign the file. You choose a passphrase at the time that a key file is generated, and it is important that you remember the passphrases that you choose. For security reasons, passphrases are not stored on the system, and Tripwire, Inc. cannot help you recover lost passphrases.

Tripwire software uses cryptographic signatures to prevent unauthorized writing of files, rather than reading of files. Only the public key is required to read files, and since the public key is available to all users, anyone can view these files.

Warning: Cryptographic techniques do not protect against all attacks, such as the deletion of Tripwire data files. For maximum security, important files should be protected by regularly verifying their hash using the Tripwire siggen utility, comparing to known reliable backups, or storing on read-only media.
Introduction to Tripwire for Servers

Tripwire Manager

Tripwire Manager is a Java-based application with a graphical user interface that allows you to manage multiple installations of Tripwire software from a central location. You can also operate the Tripwire for Servers software on each machine from the command line.

For more information on Tripwire Manager, see the Tripwire Manager User Guide, or contact Tripwire, Inc. toll-free at (877) TRIPWIRE.

Changes for This Version

- You can now specify global e-mail addresses that always receive notification when violations are found during an integrity check. See page 13 of the Tripwire Reference Guide for more information.
- You can now update individual objects in the database file, instead of the whole file. See page 40 for more information.
- Tripwire for Servers now supports SNMP traps of integrity check reports that can be used by other applications. See page 18 for more information.
- On UNIX systems, you can now configure Tripwire for Servers to cross mount points during an integrity check. See page 9 of the Tripwire Reference Guide for more information.
- On Windows systems, you can now specify a remote host for Event Log reporting. See page 17 for more information.
- You can now specify permissions for Tripwire database, policy, and report files at file creation time. See page 6 of the Tripwire Reference Guide for more information.
- You can now specify a from address for Tripwire e-mail reports. See page 11 of the Tripwire Reference Guide for more information.
Configuring Tripwire for Servers
Overview

This chapter explains how to configure Tripwire for Servers for standalone operation.

If you are using Tripwire Manager to manage Tripwire installations, you should use the Manager GUI to perform these tasks (see chapter 3 of the Tripwire Manager User Guide).

To configure Tripwire for Servers for routine operation, you need to perform the following tasks:

- Edit the configuration file to control Tripwire software operation.
- Test e-mail reporting parameters in the configuration file.
- Create a policy file that is customized for your operating system and the system objects that you want to monitor.
- Initialize the Tripwire database file to create the initial snapshot that is used for later integrity checks.
- Tune the policy file to reduce noise and false positives when you run integrity checks.

Editing the Configuration File

The configuration file controls many aspects of Tripwire software operation. For security reasons, the configuration file is stored on the system in an encoded and signed form.

The encoded configuration file is named `tw.cfg`, and is located in the Tripwire `bin` directory. A plain text copy of the same file, named `twcfg.txt`, is in the same directory. To make changes to the configuration file, you edit and save a plain text version of the file, and then encode and sign that file with the `twadmin` command.
You need to edit the configuration file parameters to change:

- the location of Tripwire files
- the parameters used to send e-mail reports
- the information that Tripwire writes to log files
- how directories are parsed during an integrity check
- the default level of detail for Tripwire report files

**Warning:** After editing, you should delete any plain text copies of the configuration file, or store them in a secure location to prevent unauthorized access.

**To edit the configuration file:**

1. Create a plain text copy of the configuration file with the `twadmin` print configuration file mode.

   ```
   twadmin --print-cfgfile > twcfg.txt
   ```

2. Open the text file with an editor, and change the values for the configuration parameters.

   See page 17 for specific information on configuring e-mail reporting, log file reporting, and SNMP logging. Consult the Tripwire Reference Guide for information on other important configuration file parameters.

3. Save the plain text configuration file.

4. Use the `twadmin` create configuration file mode to encode and sign the plain text file and install it as the new configuration file.

   ```
   twadmin --create-cfgfile --site-keyfile ..../key/site.key twcfg.txt
   ```

See page 71 for more information on the `twadmin` command.
Setting Up E-mail Reporting

You can configure Tripwire to send e-mail reports of violations every time it runs an integrity check.

To set up e-mail reporting:

1. Set the e-mail configuration file parameters based on the protocol you want to use.

<table>
<thead>
<tr>
<th>To send e-mail using</th>
<th>Change these parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMTP</td>
<td>MAILMETHOD = SMTP</td>
</tr>
<tr>
<td></td>
<td>SMTPHOST = (hostname or IP address)</td>
</tr>
<tr>
<td></td>
<td>(optional) SMTPPORT = (port #)</td>
</tr>
<tr>
<td>SENDMAIL</td>
<td>MAILMETHOD = SENDMAIL</td>
</tr>
<tr>
<td></td>
<td>MAILPROGRAM = (path to sendmail executable)</td>
</tr>
<tr>
<td>MAPI</td>
<td>MAILMETHOD = MAPI</td>
</tr>
</tbody>
</table>

2. Set GLOBALEMAIL to the address to which you want to send all e-mail reports of violations. You can specify multiple recipients as user1@domain.com;user2@domain.com.

There are many other configuration file parameters that control e-mail reporting with Tripwire software. See page 10 of the Tripwire Reference Guide for more information on these parameters.

Setting Up Log File Reporting

You can send notification of integrity checks and other Tripwire events to the log file on your machine with the SYSLOGREPORTING and SYSLOGREPORTLEVEL parameters in the configuration file.

On Windows systems, events are posted to the Application Event Log. On UNIX systems, events are posted to the syslog.
To set up syslog reporting on UNIX systems:

1. Set SYSLOGREPORTING to true in the configuration file.

2. Set SYSLOGREPORTLEVEL to a value from 0 (least detail) to 2 (full detail) to specify the detail level for syslog reports.

   See page 14 of the Tripwire Reference Guide for more information on syslog report formats.

3. Edit your syslog configuration file to add a rule for Tripwire log messages. See the syslogd(8) and syslog.conf(5) man pages for more information on editing this file.

To set up Event Log reporting on Windows systems:

1. Set SYSLOGREPORTING to true in the configuration file.

2. Set SYSLOGREPORTLEVEL to a value from 0 (least detail) to 2 (full detail) to specify the detail level for syslog reports.

   See page 14 of the Tripwire Reference Guide for more information on syslog report formats.

3. If you want to send syslog reports to the Event Log on a remote machine, set SYSLOGHOST to `\machine_name`. You can specify multiple machines as `\machine1 \machine2 \machine3`.

Setting Up SNMP Logging

You can send SNMP traps with the results of each integrity check to a machine that you specify in the configuration file. See page 16 of the Tripwire Reference Guide for more information on SNMP reporting parameters.

A Management Information Base (MIB) file containing information for Tripwire for Servers SNMP V1 traps is located on the Tripwire for Servers CD in the SNMP directory.
To set up SNMP logging:

1. Set SNMPHOST to the IP address of the host you want to use for SNMP logging.
2. Set SNMPPORT to the port number that the SNMP manager is listening on. The default port is 162.
3. Set SNMP_COMMUNITY to the string that is used for the community name in the trap message. The default is “public”.

Testing E-mail Reporting

After you set e-mail reporting options in the configuration file, you can test these parameters with the tripwire test mode.

When you execute either of the following commands, Tripwire software sends a test e-mail message to the user specified on the command line.

The tripwire test mode only tests e-mail notification for the address specified on the command line, and does not check for syntax errors with the emailto attribute in the policy file.

To test your e-mail parameters in the configuration file:

```
tripwire --test --email user@domain.com
```

To test MAPI e-mail addresses:

```
tripwire --test --email “Joe Admin”
```
Creating the Policy File

As part of the Tripwire for Servers installation process, a default policy file for your operating system is also installed. The default policy file only monitors basic components that are common to all versions of each operating system. We strongly recommend that you create a customized policy file for each machine you are monitoring.

There are 3 steps to creating a customized policy file for your system.

- Obtain an OS version-specific policy file from the Tripwire Policy Resource Center website or the Tripwire for Servers CD.
- Edit this policy file to specify additional applications or system objects that you want to monitor.
- Cryptographically sign the policy file, and install it on your machine.

The following sections describe each of these steps.

Obtaining a Policy File

You can get a fully-featured policy file, customized for your version of your operating system, in two ways:

- by creating it on the Policy Resource Center website (recommended)
- by copying it from the Tripwire for Servers CD

Both of these approaches are described in greater detail below.

Note: Because Tripwire for Servers only supports HPUX version 11.0 and AIX version 4.3, it installs complete policy files on machines running these operating systems by default. If you are using Tripwire for Servers on either of these platforms, skip to Editing the Policy File on page 22.
Tripwire Policy Resource Center Website

Tripwire Policy Resource Center is an online resource that helps you to develop an effective policy file for your system. By answering questions about your operating system and the applications that you have installed, you can develop a policy file that is secure, but that does not create false alarms. After you download this file to your local machine, you can edit it to add other system objects that you want Tripwire software to monitor.

To obtain a policy file from the Policy Resource Center website:

2. Follow the prompts on the screen to create a policy file for your system.
3. At the end of the policy creation process, download the policy file as a text file.
4. Save the text policy file to the Tripwire policy directory on your local machine.

See Editing the Policy File on page 22 for information on customizing your policy file.

Tripwire for Servers CD

The Tripwire for Servers CD contains fully-featured policy files for all versions of the operating systems that Tripwire for Servers supports.

To obtain a policy file from the Tripwire for Servers CD:

1. In the policyfiles directory of the CD, find the policy file that corresponds to the correct version of your operating system.
2. Copy the text file in this directory to the Tripwire policy directory on your local machine.

See the next section for information on customizing your policy file.
Configuring Tripwire for Servers

Editing the Policy File

After copying a file to your local machine, you can edit this file to match the specific configuration of your system, and to add other files or system objects that you want Tripwire software to monitor.

**Warning:** When you first set up the policy file, you edit a new policy file text as described in this section. After you initialize the database file and run an integrity check, you should always update the existing policy file (see page 43) instead of using the procedure below.

To customize the policy file for your machine:

1. Open your text policy file with the text editor of your choice.
   **Read through the policy file. Tripwire software interprets any text that follows the # character as a comment, and does not parse it. See the Tripwire Reference Guide for a complete description of the components of the policy file.**

2. Add or remove the # character, based on your system configuration and the files that you want to monitor.

3. Using the rules in the policy file as a guide, add additional rules to protect important files or applications on your machine. **See the Tripwire Reference Guide for more information on policy file rules.**

4. Save the policy file as a text file.

See the next section for information on cryptographically signing and installing your policy file.
Signing and Installing the Policy File

After you have created a customized policy file, you cryptographically sign the file to prevent unauthorized modification. During this process, Tripwire software creates an encoded and signed version of the plain text file. The signed version of the file is saved in the Tripwire policy directory as `tw.pol`, by default.

To sign and install the policy file for your system:

Use the `twadmin` create policy file mode to encode and sign the text policy file you have created, and install it as the new policy file.

```
twadmin --create-polfile policy file.txt
```

Initializing the Database File

After you edit the configuration and policy files, the last step in the configuration process is to create the baseline Tripwire database file. For most installations of Tripwire software, you only need to initialize the database file once, during the initial configuration process.

During database initialization, the `tripwire` executable reads the rules in your policy file, collects information from your system based on these rules, and stores this information in a database file. The database file is binary-encoded and signed to prevent unauthorized modification.

**Warning:** Because the database file serves as the baseline for all later integrity checks, make sure that you generate the database file on a machine that has not been compromised. For maximum security, you should create your baseline database immediately after installing your operating system and application files from original media.
Configuring Tripwire for Servers

To initialize the Tripwire database file:

```
tripwire --init
```

This command saves the database file to the Tripwire `db` directory or the location specified by the `DBFILE` parameter in the configuration file.

**Tuning the Policy File**

After initializing the database file, you need to tune the rules in the policy file, to ensure that Tripwire software is effectively monitoring your system without creating unnecessary noise in report files.

The chart on the next page shows the steps in the policy tuning process. After running an integrity check, you examine the resulting report file and edit the rules in your policy file that are causing errors and false positive violations. You update the policy file, then repeat the process until all errors and violations are eliminated.

After you have tuned the policy file to match your system, Tripwire software will generate reports that contain only information you want, without false positives or noise. You will only need to change the database and policy files when you make a change to your system.

**Warning:** The following procedure should only be used to run your first integrity checks and tune your policy file. After you have tuned your policy file, use the procedure on page 44 when you update the policy file.

**To tune your policy file:**

1. Run an integrity check with the following command.

```
tripwire --check --interactive
```
The first integrity check may produce warnings about files that Tripwire software could not access. This is normal for the first integrity check.

After the integrity check is finished, a report file opens in your text editor.

2. Create a plain-text copy of your policy file.

```
twadmin --print-polfile > pol_tune.txt
```
Configuring Tripwire for Servers

3. Open this text policy file in an editor next to the report file. Scroll to the section labelled Errors at the end of the report file. If there are no errors, skip to step 8.
   
   Errors in report files are usually caused when Tripwire software cannot find system objects specified in the policy file, or cannot monitor system objects because of access permissions.
   
   If Tripwire software cannot access files that you want to monitor, you need to run integrity checks from an administrator account.
   
4. Edit the rules in the policy file that are causing errors. You may need to change the paths in rules or comment them out. See the Tripwire Reference Guide for more information on policy file rules.
   
5. After making changes, save the text policy file.
   
6. Update the Tripwire policy file with the edited text file.

   ```
   tripwire --update-policy --secure-mode low pol_tune.txt
   ```

7. Repeat step 1 to step 6 until you eliminate all errors from your report files.

8. Scroll to the section of the report file labelled Object Summary. If there are no violations in this section, your policy file is tuned. Skip to the end of this procedure.
   
   The Object Summary section of the report file lists details of each violation found during an integrity check. If you do not want to see these violations during each integrity check, you need to edit the rules in your policy file to change the way that Tripwire software collects data for these objects.
   
9. Edit the policy file in the text editor. You may need to:
   
   - add special rules for files that change frequently
   - change the properties for objects
   - add stop points for some objects
   - comment out rules
See the Tripwire Reference Guide for more information on editing the rules in your policy file.

10. After making all necessary changes, save the text policy file.
11. Update the Tripwire policy file with the edited text file.

```
tripwire --update-policy --secure-mode low pol_tune.txt
```

12. Repeat step 1 to step 11 until you eliminate all errors and violations from your report files.

After you have tuned your policy file, you have finished the configuration process for Tripwire software, and are ready to begin running regular integrity checks. See chapter 3 for information on operating Tripwire software.
Using Tripwire for Servers
Overview

This chapter explains how to operate Tripwire for Servers software from the command line. If you have the optional Tripwire Manager, you can perform these operations for many Tripwire for Servers installations from its interface. See the Tripwire Manager User Guide for more information.

The diagram on the next page summarizes the routine operation of Tripwire software, after you have configured the software as described in chapter 2. Each of these steps is described in greater detail in this chapter.

The steps for routine operation of Tripwire software include:

- running an integrity check
- viewing report files
- updating the database file
- updating the policy file

This chapter also describes actions that you may need to perform outside of routine Tripwire operation, including:

- changing passphrases
- configuring the Tripwire Agent (if you are using Tripwire Manager)

Checking Integrity

After you have tuned your policy file, you can use Tripwire software to check the integrity of your system at any time. Most users schedule integrity checks at regular intervals.

During an integrity check, the tripwire executable compares the current state of the file system with the values stored in the database file to find any violations.
Using Tripwire for Servers

- Run integrity check (page 31)
  - Changes found?
    - Yes
      - Update database file (page 40)
    - No
      - Changes permitted?
        - Yes
          - Policy file working properly?
            - Yes
            - Take appropriate security measures
            - No
              - No
            - No
              - No
              - Update policy file (page 43)
        - No
          - Policy file working properly?
            - Yes
            - Take appropriate security measures
            - No
              - No
              - Update policy file (page 43)
For each integrity check, you can check all of the rules in the policy file, or only a subset of these rules. You can send the results of the integrity check via e-mail or SNMP, or write them to the syslog.

If an integrity check finds changes, you should take necessary steps to diagnose the problem. If the changes are potentially malicious, you can replace the affected files and take appropriate security measures. If the changes are valid (for example, another administrator installed new software), you should update the database file (page 40) to reflect the new state of the system.

**To run an integrity check on all objects in the policy file:**

```
tripwire --check
```

After an integrity check, Tripwire software generates a report file that summarizes violations discovered during the check. By default, Tripwire software sends a plain text copy of this report to `stdout`. In addition, a binary copy of the report is saved to the Tripwire report directory, with the filename `($HOSTNAME)-($DATE).twr`.

**To specify an alternate filename or destination for a report file:**

```
tripwire --check --report-file /directory/report_name.twr
```

You can specify the destination for the report file using either a fully-qualified path or a relative path.

**E-mailing Integrity Check Reports**

You can send the results of an integrity check to one or more recipients via e-mail. Using the `mailto` attribute in the Tripwire policy file, you can send the entire report, or only specific sections, to each recipient.
To send e-mail notification, you must:

- set the e-mail reporting parameters in the configuration file (page 17)
- specify email recipients using the `emailto` rule attribute for rules in the policy file (page 56 of the Tripwire Reference Guide) or the `GLOBALEMAIL` configuration file variable (page 13 of the Tripwire Reference Guide)
- include the `-M` or `--email-report` option of the `tripwire integrity check mode` (page 61) when running an integrity check

**To run an integrity check and send a report file via e-mail:**

```bash
tripwire --check --email-report
```

Tripwire report files can be large and detailed, so you may want to specify a lower level of detail for reports that are e-mailed. You can specify a report level from 0 for a single line summary to 4 for a very detailed report. See Appendix B of the Tripwire Reference Guide for samples of each level of report file. The default e-mail report level is 3.

**To run an integrity check and specify the level for e-mail reports:**

```bash
tripwire --check --email-report --email-report-level 2
```

**Selective Integrity Checks**

During a basic integrity check, Tripwire software uses all of the rules in the policy file to check system integrity.

If you want to check parts of your system (the web server software, for example) more frequently than others, you can run an integrity check using only a subset of these rules, or only a single rule.
To check only a specific rule during an integrity check:

If you use the `rulename` rule attribute (page 51 of the Tripwire Reference Guide) to name rules in your policy file, you can run an integrity check for a specific rule or rule block. For example, suppose one of these rules is in your policy file.

```
W
D:\project -> &size &write &haval (rulename="My Project");

U
/usr/project -> +smH (rulename="My Project");
```

You could run an integrity check using only that rule.

```
tripwire --check --rule-name "My Project"
```

If you want to run several rules, you can group them into a rule block (page 56 of the Tripwire Reference Guide) in the policy file.

```
W
(rulename="My Project")
{
    D:\project\test -> &size &write &haval;
    D:\project\secret -> &write &sdc &sha &haval;
}

U
(rulename="My Project")
{
    /usr/project\test -> +smH;
    /usr/project\secret -> +mSH;
}
```

With either of these rule blocks, if you run an integrity check with `--rulename "My Project"`, both of the rules in brackets are used.
To check only specific objects during an integrity check:

Usually, Tripwire software checks all system objects specified by the rules in the policy file. However, you can check only specific objects.

```
tripwire --check object object object...
tripwire --check /bin /usr
```

```
tripwire --check section: object1 object2 ... section: object1 object2 ...
tripwire --check NTFS: C:\winnt D:\special NTREG: \HKEY_LOCAL_MACHINE
```

To check rules based on their severity level:

If you use the `severity` rule attribute (page 52 of the Tripwire Reference Guide) to categorize the rules in your policy file by importance, you can run an integrity check using only the rules at the specified severity level or higher.

For example, suppose that the following rules are in your policy file.

```
C:\winnt -> &write (severity=90);
C:\projects -> &write &sdc (severity=75);
```

```
/bin -> +m (severity=90);
/projects -> +mH (severity=75);
```

If you run an integrity check with a severity level of 80,

```
tripwire --check --severity 80
```

Tripwire software checks the first rule, but ignores the second rule.
To ignore properties during an integrity check:

When running an integrity check, collecting data for some properties—particularly hashes—can be time-consuming. To run an integrity check that ignores specific properties for all objects, list the properties to ignore (see page 47 of the Tripwire Reference Guide).

When Tripwire software runs the integrity check, it does not collect the current data for the properties you specify, and therefore does not compare these properties to the information stored in the database file. This can greatly reduce the time required for an integrity check.

For example, if you run an integrity check with the following command.

```
tripwire --check --ignore "H,S,M"
```

Tripwire software does not calculate the Haval, SHA/SHS, or MD5 checksums for any objects that it checks. All other properties specified in the policy file are checked.

For Windows systems, you can specify different properties to ignore for file system and registry objects.

```
tripwire --check --ignore "NTFS:haval,sha,md5,NTREG:md5"
```

In this example, Tripwire software does not calculate the Haval, SHA/SHS, or MD5 checksums for any file system objects, and does not calculate the MD5 checksum for any registry objects. All other properties are checked normally.

See page 61 for a complete listing of command-line options for the `tripwire` integrity check mode.
Scheduling Integrity Checks

The most common ways to schedule Tripwire integrity checks are with the `at` command on Windows systems, or with `cron` on UNIX systems.

When scheduling recurring integrity checks, make sure that only one check is running at a time. For example, if you are running a full integrity check nightly, and also running a check with only a few rules every hour, make sure that the daily check begins and ends between the hourly integrity checks.

To schedule an integrity check on a Windows system:
(to run every evening at 1:00 am, and send an e-mail report)

```
W
at 01:00 /every:M,T,W,Th,F,S,Su cmd /c path\tripwire --check -M
```

where `path` is the location of the Tripwire executables.

When using `at` to schedule an integrity check:

• place the `tripwire` command-line sequence after `cmd /c`, which tells Windows to launch a Command Prompt (`cmd.exe`)

• use the `/interactive` option of the `at` command to run Tripwire software on the desktop of the user who is logged in when the job occurs

• the user who schedules the `at` job must have Administrator privileges

For more information on the `at` command, consult the Windows help system or type the following at a command prompt:

```
at /?
```
To schedule an integrity check on UNIX systems using crontab:
(to run once an hour every day, and send an e-mail report)

Insert the following lines in your crontab file.

```
U
# Run tripwire hourly.
0 */1 * * * /usr/local/tripwire/bin/tripwire --check -M
```

For more information about scheduling integrity checks with the crontab command, see the man pages for crontab(1) and cron(8).

Viewing Report Files

When you run an integrity check, Tripwire software prints a copy of the report file to the screen, and saves a binary copy of the report file to the Tripwire report directory, or to the location specified by the REPORTFILE parameter in the configuration file.

By default, Tripwire report files are named ($HOSTNAME)-($DATE).twr. For example, a report generated on January 27, 2001 at 2:15:01 PM for a computer with the hostname GARNET would be named GARNET-20010127141501.twr.

You can examine any report file in detail with the twprint command. The report is displayed at the level of detail specified by the REPORTLEVEL parameter in the configuration file, but you can specify a different report level (0 to 4) on the command line.

To print an existing report file to the screen:

```
twprint --print-report --twrfile ..\report\report.twr
```
To print an existing report file as a text file:

```
twprint --print-report --twrfile ..\report\report.twr > myreport.txt
```

To specify the report level when printing a report file:

```
twprint --print-report --report-level 4 --twrfile ..\report\report.twr
```

**Updating the Database File**

If Tripwire software finds changes during an integrity check, you should update the database file to reflect the current state of the system. You do not want authorized changes to be interpreted as violations in future integrity checks.

In database update mode, Tripwire opens a report file using the editor specified by the EDITOR parameter in the configuration file. If the report file contains legitimate changes, you can approve them, then update the database file to reflect the new information.

If you know that only certain parts of your system have changed, you can update only those parts of the database file. You can specify particular system objects, rule names, or sections of database file information that you want to update.

To update the database file with changes from a report file:

1. To update the database file from an existing report file, enter the name of the report file on the command line.

```
tripwire --update --twrfile ..\report\reportfile.twr
```
Or, to update the database file immediately after an integrity check, run an integrity check with the --interactive option.

```
tripwire --check --interactive
```

2. Scroll through the report to the list of violations. Each violation in the file is displayed with a corresponding ballot box. At the start of the database update, each ballot box contains an x.

```
Modified:
[x] "d:\temp"
```

```
Modified:
[x] "/usr/local/tripwire"
```

3. To approve a change, leave the x next to each policy violation. If you remove the x from the box, the database file is not updated with the new value for the object.

4. Save the edited file and exit the editor.

5. Provide the local passphrase, and the tripwire executable updates and saves the database file.

**To update only specific database entries:**

1. Enter the name of the report file on the command line, and specify the rule name (see page 51 of the Tripwire Reference Guide) that you want to update.

```
tripwire --update --twrfile ..\report\20010102154534.twr --rule-name "My Project"
```
Using Tripwire for Servers

Or, you can update only one section (page 33 of the Tripwire Reference Guide) of the database file.

```
tripwire --update --twrfile ..\report\20010102154534.twr --section NTFS
```

Or, you can specify files or other system objects to update. For example:

```
tripwire --update --twrfile ../report/20010102154534.twr /bin /usr/local
```

```
tripwire --update --twrfile ../report/20010102154534.twr
   NTFS: C:\\winnt D:\\special NTREG: \HKEY_LOCAL_MACHINE
```

2. Scroll through the violations, leaving the x next to each violation that you want to approve.

3. Save the edited file and exit the editor.

4. Provide the local passphrase, and the `tripwire` executable updates and saves the database file.

See page 64 for a complete listing of command-line options for the `tripwire` database update mode.

### Resolving Database Update Problems

You may encounter problems when updating the database file if:

- the report file you specify has already been used to update the database file
- the report file you specify was generated using a different database file
- the database file has been updated with a more recent report file
Tripwire software responds to database update errors according to the security mode you specify on the command line. Because database update errors can corrupt the database file, always update the database file in high security mode.

- In high security mode (the default), any error causes the `tripwire` executable to exit without updating the database file.
- In low security mode, a warning is printed, but the database file is still updated with the new information.

If you cannot resolve the errors in any other way, you can update the database file in low security mode.

```
tripwire --update --twrfile reportfile.twr --secure-mode low
```

### Updating the Policy File

By changing the rules in the policy file, you can change the way that Tripwire software monitors your system. For example, you may want to change the rules in the policy file to:

- monitor new files or software on a machine
- reduce false positive results or report-file noise
- send e-mail reports to different people
- group policy file rules differently

The `tripwire` policy update mode allows you to edit the policy file and synchronize the existing database file with the new information in the policy file. Policy update mode is the only way to change the rules in the policy file securely without re-initializing the database file. See page 44 for more information on the policy update process.
To update the policy file:

1. Create a plain text copy of the policy file.

   twadmin --print-polfile > policy.txt

2. Edit and save the text policy file.

3. Apply the changes to the existing policy file.

   tripwire --update-policy policy.txt

The new encoded and signed policy file is named *tw.pol*, located in the directory specified by the POLFILE parameter in the configuration file. You can confirm policy file changes by running *tripwire* in integrity checking mode.

See page 66 for a complete listing of command-line options for the *tripwire* policy update mode.

The Policy Update Process

The policy update process follows these steps:

1. The *tripwire* executable compares the new, plain text policy file specified on the command line to the existing version of the policy file.

2. The *tripwire* executable reads the rules in the new policy file, and runs an integrity check to gather information about the current state of the system.

3. As data are collected, any violations (additions, deletions, or changes) of the rules in the old policy file that are also covered by rules in the new policy file are detected and reported.

4. These violations are interpreted based on the security mode specified on the command line with the -Z or --secure-mode option.
In high security mode (the default), `tripwire` prints a list of violations and exits without changing the database file.

In low security mode, `tripwire` reports the violations, but still changes the database file.

5. After the database file is updated with new data, the old version of the policy file is replaced with the new version. The new database file now reflects the current state of the system.

**Warning:** Conflicts discovered when updating the policy file should be treated with the same seriousness as integrity checking violations. For this reason, we recommend that you always update the policy file in high security mode, so that these situations can be detected, and appropriate actions taken.

### Resolving Policy Update Problems

By default, `tripwire` policy update mode runs in high security mode, as described above. You may encounter errors when running in high security mode if the file system has changed since the last database update, and if the changes still violate the rules in the new policy file. This could happen if another administrator is modifying files during the policy update process, for example.

To resolve this, determine whether all of the violations reported in high security mode are authorized, then update the policy file in low security mode.

```
tripwire --update-policy --secure-mode low policy.txt
```
Changing Passphrases

Important Tripwire files are cryptographically signed to protect them against unauthorized modification. Policy and configuration files are signed with the site key file, and database and (optionally) report files are signed with the local key file. Each key file is inextricably linked with a passphrase, so you must change the key file to change your passphrases.

You can change your key files and passphrases at any time. It is a good security practice to change your keys periodically, or you may want to change the keys after staff changes, or if you think your passphrases have been compromised.

See page 9 for more information on key files and passphrases.

Warning: Deleting or overwriting the key file used to sign a Tripwire file makes that file permanently unusable. Always make backup copies of key files before changing encryption.

To change the site key file for all configuration and policy files:

1. If you do not know what site key file signed a file, use the `twadmin` examine encryption command to find out.

   `twadmin --examine file1 file2 ...`

2. Remove the cryptographic signatures from the configuration and policy files. You must enter the current site passphrase to remove the signature from these files.

   `twadmin --remove-encryption tw.cfg ./policy/*.*`
3. Generate a new site key file named `site.key`, located in the Tripwire `key` directory. You are prompted to choose a passphrase for the new site key file.

```
twadmin --generate-keys --site-keyfile ../key/site.key
```

4. Cryptographically sign the configuration and policy files with the new site key file.

```
twadmin --encrypt --site-keyfile ../key/site.key tw.cfg ../policy/*.*
```

To change the local key file for all database and report files:

1. If you do not know what local key file signed a file, use the `twadmin` `examine encryption` command to find out.

```
twadmin --examine file1 file2 ...
```

2. Remove the cryptographic signatures from the database and report files. You must enter the current local passphrase to remove encryption from these files.

```
twadmin --remove-encryption ../db/*.* ../report/*.*
```

3. Generate a new local key file. You are prompted to choose a passphrase for the new local key file.

```
twadmin --generate-keys --local-keyfile ../key/hostname-local.key
```
4. Cryptographically sign the database and report files with the new local key file.

```
twadmin --encrypt --local-keyfile ../key/hostname-local.key ../db/*.* ../report/*.*
```

**Tripwire Agent**

Tripwire Agent is a part of Tripwire for Servers software that manages communication with the Tripwire Manager. If you are not using the Tripwire Manager, you do not need to use Tripwire Agent.

On UNIX machines, the Tripwire Agent is a daemon; on Windows machines, it is a service. On either platform, you have the option to start the Agent during the Tripwire for Servers installation process.

To change the operation of the Tripwire Agent, you need to edit the Agent configuration file. You may need to edit the Agent configuration file to change:

- the port number that the Agent uses to communicate with the Tripwire Manager
- the filename or location of any of the files that the Agent configuration file references
- the location of the authentication key file used to authenticate communication with the Tripwire Manager
- the site key file (and site passphrase) used to sign the Agent configuration file

See page 19 of the Tripwire Reference Guide for more information on the parameters in the Agent configuration file.
Using Tripwire Agent on Windows Systems

To start or stop Tripwire Agent:

1. For Windows NT, select Start > Settings > Control Panel.
2. Double-click Services.
3. For Windows NT, select Tripwire Agent and click Start or Stop.
   For Windows 2000, right-click Tripwire Agent and click Start or Stop.
4. Click Close to close the Services window.

To edit the Agent configuration file:

1. Stop the Tripwire Agent (see previous procedures).
2. Print a plain text version of the Agent configuration file.
   
tagent --print-cfgfile > agentcfg.txt

4. Encode and sign the text file and install it as the new Agent configuration file.
   
tagent --create-cfgfile --site-keyfile ..\key\site.key agentcfg.txt
The new Agent configuration file is named \textit{agent.cfg} and is saved to the same directory as the twagent executable, unless you specify another name or destination on the command line with the --cfgfile option.

\begin{verbatim}
tagent --create-cfgfile --site-keyfile ..\key\site.key
        --cfgfile C:\agents\agent.cfg agentcfg.txt
\end{verbatim}

5. Start the Agent (see procedure above).

See page 81 for more information on the \texttt{twagent} command.

\textbf{To add Tripwire Agent to the Windows Services List:}

\begin{verbatim}
tagent --install
\end{verbatim}

If you omit options to the --install command, the Agent runs as user SYSTEM and launches at startup.

You can run the Agent as another user, and specify different startup options with the --install command (page 83).

\textbf{To remove the Tripwire Agent from the Windows Services list:}

\begin{verbatim}
tagent --remove
\end{verbatim}

\textbf{To change startup options for Tripwire Agent:}

1. For Windows NT, select Start > Settings > Control Panel.

2. Double-click Services.

3. For Windows NT, select Tripwire Agent and click Startup.
   For Windows 2000, right-click Tripwire Agent and select Properties.
4. For Startup type, select Automatic to start the Agent automatically at system reboot, or select Manual to start the Agent manually.

5. Click OK.

**To change the user account that Tripwire Agent uses:**

1. For Windows NT, select Start > Settings > Control Panel.
   

2. Double-click Services.

3. For Windows NT, select Tripwire Agent and click Startup.
   
   For Windows 2000, right-click Tripwire Agent, then select Properties and the Log On tab.

4. Select This Account in the Log On As section, then click the browse button to select from a list of users.

5. For Windows NT, select a user, then click Add and OK.
   
   For Windows 2000, select a user, then click OK.

6. Enter and confirm the login password for the user, then click OK.

Only the user who created a Tripwire database file can access that file to run an integrity check. If you change the user permissions for the Agent, you may not be able to access existing database files.

### Using Tripwire Agent on UNIX Systems

**To start the Tripwire Agent:**

```
twagent --start
```
To stop the Tripwire Agent:

1. Find the process number for the Agent.
   
   ps -e | grep twagent

2. Kill the process.
   
   kill -9 process#

To edit the Agent configuration file:

1. Stop the Tripwire Agent daemon (see the procedure above).
2. Print a plain text version of the Agent configuration file.
   
   twagent --print-cfgfile > agentcfg.txt

4. Encode and sign the text file and install it as the new Agent configuration file.
   
   twagent --create-cfgfile --site-keyfile ../key/site.key agentcfg.txt
The new Agent configuration file is named `agent.cfg` and is located in the same directory as the `twagent` executable, unless you specify another name or destination on the command line with the `--cfgfile` option.

```
twagent --create-cfgfile --site-keyfile ../key/site.key
    --cfgfile /agents/agent.cfg agentcfg.txt
```

5. Start the Agent daemon.

```
twagent --start
```

See page 81 for more information on the `twagent` command.
Command Reference
Introduction

This section describes the commands that Tripwire for Servers uses, and lists the command-line options for each of those commands. The following five executables are used for all Tripwire operations:

- `tripwire` creates the baseline database, checks integrity, updates the database and policy files, and tests e-mail configuration parameters.
- `twadmin` creates configuration and policy files and performs cryptographic operations with Tripwire files.
- `twprint` opens Tripwire database and report files as plain text.
- `siggen` generates and prints hashes for specified files.
- `twagent` controls the operation of the Tripwire Agent, which Tripwire software uses to communicate with the Tripwire Manager.

All commands take their default values from the configuration file, unless you specify a value on the command line.

Command Conventions

You must construct all Tripwire commands as follows:

```
command mode option1 argument1 option2 argument2 objects
```

Specify the command followed by the mode, followed by one or more options and arguments. Objects associated with the command mode are last. Commands specified in any other order generate a syntax error.

All Tripwire modes and options have both a short form and a long form. The short form is a single letter preceded by a single `-` character. The equivalent long form is a more descriptive word or phrase preceded by double `-` characters. You can combine short and long forms of options in any command.
Command Reference

For example, the following commands are functionally identical:

```
tripwire --update-policy --site-keyfile A:\site.key --silent
   --secure-mode high newpolicy.txt
```

```
tripwire -mp -S A:\site.key -s -Z high newpolicy.txt
```

You must use full path names (explicit or relative) when specifying file names as command-line arguments. Universal Naming Convention (UNC) names are supported for all files on Windows systems, using the format `\machinename\share`.

```
\twadmin --create-cfgfile --site-keyfile ..\key\site.key
   \GARNET\config\config.txt
```

# Command-Line Help

All Tripwire commands support the following arguments for obtaining usage, version, and copyright information. If a help argument is on the command line, the help message is displayed and all other command-line arguments are ignored.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`-?</td>
<td>Display command modes and version information.</td>
</tr>
<tr>
<td><code>--help</code></td>
<td>Display command modes and version information.</td>
</tr>
<tr>
<td>`-? all</td>
<td>Display help for all command modes.</td>
</tr>
<tr>
<td><code>--help all</code></td>
<td>Display help for all command modes.</td>
</tr>
<tr>
<td><code>-? mode</code></td>
<td>Display help for the current command mode.</td>
</tr>
<tr>
<td><code>--help mode</code></td>
<td>Display help for the current command mode.</td>
</tr>
<tr>
<td>`-m V</td>
<td>Display version information.</td>
</tr>
<tr>
<td><code>--version</code></td>
<td>Display version information.</td>
</tr>
</tbody>
</table>
Wildcards

You can use wildcards to specify directories or files for Tripwire commands that accept multiple command-line arguments.

Warning: Using wildcards on the command line creates a small but potentially significant security risk. By inserting a file that mimics a command-line option, an intruder could adversely affect the operation of Tripwire software.

tripwire

You can run the tripwire command in one of five modes.

In database initialization mode, the tripwire executable builds a database of information about your system, based on the rules in the policy file. This database file serves as the baseline for later integrity checks. You should only perform this step when Tripwire software is first installed.

The integrity checking mode compares the actual state of your system to the information stored in the database file. A report of any violations discovered is sent to stdout, and a binary copy of the report is stored in the Tripwire report directory.

After you run an integrity check, use database update mode to update the database file, using the changes from a report file. This is the only way you can securely update the information in the database file.

Use policy update mode to change the rules in the policy file, to change the way that Tripwire software monitors your system. Tripwire software automatically updates the information in the database file to reflect the changes in the policy file.

Use test mode to test the Tripwire e-mail notification parameters in the configuration file.
tripwire Database Initialization Mode

In database initialization mode, the tripwire executable generates a database file based on the policy file rules and then signs it. Because this database becomes the baseline for later integrity checks, it is essential that you create the database file from a system that has not been compromised.

The short and long forms of the command are as follows:

```
tripwire -m i [options]
tripwire --init [options]
```

where `options` includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-v</code>    --verbose</td>
<td>Display additional status information. Mutually exclusive with --silent.</td>
</tr>
<tr>
<td><code>-s</code>    --silent</td>
<td>Suppress additional status information. Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td><code>-p</code>    --polfile</td>
<td>Use this policy file instead of the policy file specified in the configuration file.</td>
</tr>
<tr>
<td><code>-c</code>    --cfgfile</td>
<td>Use this configuration file, instead of <code>tw.cfg</code>.</td>
</tr>
<tr>
<td><code>-S</code>    --site-keyfile</td>
<td>Use this site key file to read the configuration and policy files.</td>
</tr>
<tr>
<td><code>-L</code>    --local-keyfile</td>
<td>Use this local key file to sign the database file. Mutually exclusive with --no-encryption.</td>
</tr>
<tr>
<td><code>-d</code>    --dbfile</td>
<td>Write the new database file here, instead of the default location specified by the <code>DBFILE</code> parameter in the configuration file.</td>
</tr>
<tr>
<td><code>-P</code>    --local-passphrase</td>
<td>Use this passphrase with the local key to sign the database file. Mutually exclusive with --no-encryption.</td>
</tr>
<tr>
<td><code>-e</code>    --no-encryption</td>
<td>Do not sign the database file. The file is still binary-encoded. Mutually exclusive with --local-passphrase and --local-keyfile.</td>
</tr>
</tbody>
</table>
tripwire Integrity Check Mode

An integrity check compares the current state of system objects with the values stored in the database file, then reports violations based on rules in the policy file.

By default, a plain text copy of this report is sent to the screen, and a binary copy of the report is saved to the Tripwire report directory. You can also direct Tripwire software to send e-mail reports at different levels of detail.

The short and long forms of the command are as follows:

```
tripwire -m c [options] [objects]
tripwire --check [options] [objects]
```

where `options` includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-I</td>
<td>--interactive</td>
</tr>
<tr>
<td>-v</td>
<td>--verbose</td>
</tr>
<tr>
<td>-s</td>
<td>--silent</td>
</tr>
<tr>
<td>-p</td>
<td>--polfile</td>
</tr>
<tr>
<td>-d</td>
<td>--dbfile</td>
</tr>
<tr>
<td>-c</td>
<td>--cfgfile</td>
</tr>
<tr>
<td>-S</td>
<td>--site-keyfile</td>
</tr>
<tr>
<td>-L</td>
<td>--local-keyfile</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>-P</code></td>
<td><code>--local-passphrase</code> &lt;i&gt;local passphrase&lt;/i&gt; Use this passphrase with the local key. Also used to write the database file in <code>--interactive</code> mode. Valid only with <code>--signed-report</code> or <code>--interactive</code>.</td>
</tr>
<tr>
<td><code>-n</code></td>
<td><code>--no-tty-output</code> Do not display the report at the console.</td>
</tr>
<tr>
<td><code>-r</code></td>
<td><code>--report-file</code> &lt;i&gt;report file&lt;/i&gt; Write the output report file here, instead of the location specified by the REPORTFILE parameter in the configuration file.</td>
</tr>
<tr>
<td><code>--email-report</code></td>
<td>E-mail reports to the recipients designated in the policy file, using the e-mail notification settings in the configuration file.</td>
</tr>
<tr>
<td><code>-E</code></td>
<td><code>--signed-report</code> Sign the report file. If you do not specify a passphrase on the command line, you are prompted for the local passphrase.</td>
</tr>
<tr>
<td><code>-t</code></td>
<td><code>--email-report-level</code> &lt;i&gt;{0</td>
</tr>
<tr>
<td><code>-l</code></td>
<td><code>--severity</code> &lt;i&gt;{level</td>
</tr>
<tr>
<td><code>-R</code></td>
<td><code>--rule-name</code> &lt;i&gt;rule name&lt;/i&gt; Only run this policy rule. See page 51 of the Tripwire Reference Guide for more information on the naming rules in the policy file. Mutually exclusive with <code>--severity</code>.</td>
</tr>
<tr>
<td><code>-V</code></td>
<td><code>--visual</code> &lt;i&gt;editor&lt;/i&gt; Use this editor to update the database file after an integrity check. Only applies with the <code>--interactive</code> option.</td>
</tr>
<tr>
<td><code>-x</code></td>
<td><code>--section</code> &lt;i&gt;section&lt;/i&gt; Only check policy rules in this section (NTFS or NTREG) of the policy file.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>-i</code></td>
<td><strong>Do not compute or compare the properties specified in list.</strong> The format for list is:</td>
</tr>
<tr>
<td><code>--ignore</code></td>
<td><strong>&quot;property,property,property...&quot;</strong></td>
</tr>
<tr>
<td><code>list</code></td>
<td><strong>&quot;C,M,S,H&quot;</strong></td>
</tr>
<tr>
<td></td>
<td><strong>For Windows systems, the format for list is:</strong></td>
</tr>
<tr>
<td></td>
<td><strong>&quot;section:property,property,section:property&quot;</strong></td>
</tr>
<tr>
<td></td>
<td><strong>&quot;NTFS:haval,md5,NTREG:group&quot;</strong></td>
</tr>
<tr>
<td></td>
<td><strong>where section (NTFS or NTREG) specifies a section of the policy file and property specifies properties to ignore for that section. You can specify any section one or more times in the list.</strong></td>
</tr>
<tr>
<td><code>object1 object2...</code></td>
<td><strong>Only check these file system and registry objects. If objects are not specified, every object in the policy file is checked. This option overrides the --severity, --rule-name, --section, and --email-report options.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>For Windows systems, the format for a list of objects is:</strong></td>
</tr>
<tr>
<td></td>
<td><strong>section: object object section: object...</strong></td>
</tr>
<tr>
<td></td>
<td><strong>ntfs: C:\vital.doc NTREG: HKEY_USERS</strong></td>
</tr>
<tr>
<td></td>
<td><strong>where section (NTFS or NTREG, case insensitive) specifies a section of the policy file, and objects are specified using the policy file syntax.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>You may specify the NTFS or NTREG section one or more times; if you do not specify a section, NTFS is assumed.</strong></td>
</tr>
</tbody>
</table>
Tripwire Database Update Mode

The tripwire database update mode displays a report file, allowing you to view violations and approve changes to the database file. The editor that you use to update the database file is specified by the --visual option, the EDITOR value in the configuration file, or the $VISUAL or $EDITOR environment variables, in that order.

You may specify a report on the command line, or Tripwire software uses the file specified by the REPORTFILE parameter in the configuration file. By default, the REPORTFILE parameter includes a time-based variable, $(DATE). If you use this default value, you must specify a report file on the command line. Tripwire software cannot find a file with the $(DATE) variable because the current date and time do not match the date when the file was saved.

The short and long forms of the command are as follows:

```
tripwire -m u [options][objects]
tripwire --update [options][objects]
```

where options includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v</td>
<td>Display additional status information.</td>
</tr>
<tr>
<td>--verbose</td>
<td>Mutually exclusive with --silent.</td>
</tr>
<tr>
<td>-s</td>
<td>Suppress additional status information.</td>
</tr>
<tr>
<td>--silent</td>
<td>Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td>--quiet</td>
<td></td>
</tr>
<tr>
<td>-d</td>
<td>Update this database file, instead of the database file specified in the configuration file.</td>
</tr>
<tr>
<td>--dbfile</td>
<td></td>
</tr>
<tr>
<td>-c</td>
<td>Use this configuration file, instead of tw.cfg.</td>
</tr>
<tr>
<td>--cfgfile</td>
<td></td>
</tr>
<tr>
<td>-S</td>
<td>Use this site key file to read the configuration and policy files.</td>
</tr>
<tr>
<td>--site-keyfile</td>
<td></td>
</tr>
<tr>
<td>-L</td>
<td>Use this local key file to read and write the database file and to read the report file.</td>
</tr>
<tr>
<td>--local-keyfile</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>-r</code></td>
<td><code>--report-file</code> report file</td>
</tr>
<tr>
<td><code>-P</code></td>
<td><code>--local-passphrase</code> passphrase</td>
</tr>
<tr>
<td><code>-V</code></td>
<td><code>--visual</code> editor</td>
</tr>
<tr>
<td><code>-a</code></td>
<td><code>--accept-all</code></td>
</tr>
<tr>
<td><code>-Z</code></td>
<td><code>--secure-mode</code> { high</td>
</tr>
<tr>
<td><code>-R</code></td>
<td><code>--rule-name</code> rule name</td>
</tr>
<tr>
<td><code>-x</code></td>
<td><code>--section</code> section</td>
</tr>
</tbody>
</table>

Tripwire for Servers User Guide 65
tripwire Policy Update Mode

You can use the tripwire policy update mode to change the contents of the policy file and to synchronize the database file with this new policy file information.

The short and long forms of the command are as follows:

```
tripwire -m p [options] policy_file.txt
tripwire --update-policy [options] policy_file.txt
```

where `options` includes the following options and arguments:

```
-v --verbose        Display additional status information. Mutually exclusive with --quiet.
-s --silent         Suppress additional status information. Mutually exclusive with --verbose.
-p --polfile policy_file Update this policy file, instead of the policy file specified in the configuration file.
```

For Windows systems, the format for a list of objects is:

```
section: object object section: object...
ntfs: C:\vital.doc NTREG: HKEY_USERS
```

where `section` (NTFS or NTREG, case insensitive) specifies a section of the policy file, and objects are specified using the policy file syntax.
You can use the `tripwire` test mode to check the parameters in the configuration file by sending a test e-mail message.

Test mode only tests e-mail notification for the address specified on the command line; it does not check for syntax errors with the `emailto` attribute in the policy file.
The short and long forms of the command are as follows:

```
tripwire -m t -e email_address
tripwire --test --email email_address
```

where `email address` is expressed as `user@domain.com` or “MAPI Name” for Windows systems with MAPI e-mail addresses.

**twprint**

Use the `twprint` command to view and print Tripwire database and report files in plain text form. Tripwire database files are binary encoded and signed. Tripwire report files are encoded and may optionally be signed. Printing Tripwire policy or configuration files requires the `twadmin` command, as described on page 71.

**To redirect the output from the `twprint` command to a text file:**

```
twprint --print-dbfile > file.txt
```

**twprint Print Report Mode**

Use the `twprint` print report mode to view Tripwire report files. On the command line, use the `--twrfile` option to specify the report that you want to view.
If you do not specify a report file, `twprint` attempts to display the report specified by the REPORTFILE parameter in the configuration file. The default value for REPORTFILE is `$HOSTNAME-$DATE.twr`, where `$DATE` represents the time that the report was generated, to the nearest second. Since the value for `$DATE` when you attempt to print the file is different from the value when the report was generated, `twprint` cannot find a valid report file, and you must specify a report file on the command line.

The short and long forms of the command are as follows:

```
twprint -m r -r report_file [options]
twprint --print-report --twrfile report_file [options]
```

where `options` includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v --verbose</td>
<td>Display additional status information. Mutually exclusive with --silent.</td>
<td></td>
</tr>
<tr>
<td>-s --silent</td>
<td>Suppress additional status information. Mutually exclusive with --verbose.</td>
<td></td>
</tr>
<tr>
<td>-c --cfgfile</td>
<td>Use this configuration file, instead of <code>tw.cfg</code>.</td>
<td></td>
</tr>
<tr>
<td>-r --report-file</td>
<td>Print this report file, instead of the report file specified in the configuration file.</td>
<td></td>
</tr>
<tr>
<td>-L --local-keyfile</td>
<td>Use this local key file to verify the report file, if it was signed.</td>
<td></td>
</tr>
<tr>
<td>-t --report-level</td>
<td>Print the report at this level of detail. Report level 0 is the least detailed, and 4 is the most detailed.</td>
<td></td>
</tr>
</tbody>
</table>
### twprint Print Database Mode

Use the `twprint` print database mode to print the contents of a Tripwire database file to the screen, or to a text file. You can specify a database file with the `--dbfile` option, or use the database file specified by the `DBFILE` parameter in the configuration file.

The short and long forms of the command are as follows:

```
twprint -m d [options] [objects]
twprint --print-dbfile [options] [objects]
```

where `options` includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-v</code></td>
<td><code>--verbose</code> Display additional status information. Mutually exclusive with <code>--silent</code>.</td>
</tr>
<tr>
<td><code>-s</code></td>
<td><code>--silent</code> Suppress additional status information. Mutually exclusive with <code>--verbose</code>.</td>
</tr>
<tr>
<td><code>-c</code></td>
<td><code>--cfgfile</code> Use this configuration file, instead of <code>tw.cfg</code>.</td>
</tr>
<tr>
<td><code>-d</code></td>
<td><code>--dbfile</code> Print this database file, instead of the file specified in the configuration file.</td>
</tr>
<tr>
<td><code>-L</code></td>
<td><code>--local-keyfile</code> Use this local key file to read the database file.</td>
</tr>
<tr>
<td><code>object1 object2...</code></td>
<td>Only print these database objects. Wildcards may be used to specify a group of objects, but wildcard use is discouraged for security reasons.</td>
</tr>
</tbody>
</table>

For Windows systems, the format for a list of objects is:

```
section: object object section: object...
ntfs: C:\vital.doc NTREG: HKEY_USERS
```

where `section` (NTFS or NTREG, case insensitive) specifies a section of the policy file, and objects are specified using the policy file syntax. You may specify the NTFS or NTREG sections one or more times; if you do not specify a section, NTFS is assumed.
**twadmin**

The `twadmin` command has eight command modes. Four of these modes are used for editing and printing the configuration and policy files:

**Create configuration file** mode designates an existing plain text file as the new configuration file.

**Print configuration file** mode prints the configuration file in plain text.

**Create policy file** mode designates an existing plain text file as the new Tripwire policy file. This mode should **not** be used to update an existing policy file. See page 43 for information on updating the policy file.

**Print policy file** mode prints the policy file in plain text.

The other four modes of the `twadmin` command are used for managing keys and encryption:

**Remove encryption** mode removes cryptographic signatures from configuration, policy, database, or report files.

**Encryption** mode signs configuration, policy, database, or report files cryptographically.

**Examine encryption** mode reports the encryption status of Tripwire files.

**Generate keys** mode creates site or local keys for Tripwire files.

**Warning:** You may want to move the `twadmin` executable to a floppy disk after installation to prevent unauthorized reading of configuration and policy files.
twadmin Create Configuration File Mode

Use this mode to designate an existing plain text file as the new configuration file. After you specify the plain text file on the command line, Tripwire software encodes, signs, and saves the new configuration file.

The short and long forms of the command are as follows:

```
twadmin -m F [options] configuration_file.txt
twadmin --create-cfgfile [options] configuration_file.txt
```

where `options` includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v</td>
<td>Display additional status information. Mutually exclusive with --silent.</td>
</tr>
<tr>
<td>--verbose</td>
<td></td>
</tr>
<tr>
<td>-s</td>
<td>Suppress additional status information. Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td>--silent</td>
<td></td>
</tr>
<tr>
<td>--quiet</td>
<td></td>
</tr>
<tr>
<td>-S</td>
<td>Use this site key file to sign the new configuration file. Mutually exclusive with the --no-encryption option. You must specify either --no-encryption or --site-keyfile.</td>
</tr>
<tr>
<td>--site-keyfile</td>
<td><code>site key file</code></td>
</tr>
<tr>
<td>-c</td>
<td>Create the new configuration file here.</td>
</tr>
<tr>
<td>--cfgfile</td>
<td></td>
</tr>
<tr>
<td>-e</td>
<td>Do not sign the configuration file. Mutually exclusive with the --site-keyfile and --site-passphrase options. You must specify either --no-encryption or --site-keyfile.</td>
</tr>
<tr>
<td>--no-encryption</td>
<td></td>
</tr>
<tr>
<td>-Q</td>
<td>Use this site key to sign the configuration file. Valid only with --site-keyfile.</td>
</tr>
<tr>
<td>--site-passphrase</td>
<td><code>passphrase</code></td>
</tr>
<tr>
<td><code>configuration file.txt</code></td>
<td>The plain text version of the configuration file.</td>
</tr>
</tbody>
</table>
twadmin Print Configuration File Mode

Use this mode to print the current configuration file in plain text. The file is stored in a binary-encoded form.

The short and long forms of the command are as follows:

```
twadmin -m f [options]
twadmin --print-cfgfile [options]
```

where `options` includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v</td>
<td>Display additional status information. Mutually exclusive with --silent.</td>
</tr>
<tr>
<td>--verbose</td>
<td>Display additional status information. Mutually exclusive with --silent.</td>
</tr>
<tr>
<td>-s</td>
<td>Suppress additional status information. Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td>--silent</td>
<td>Suppress additional status information. Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td>--quiet</td>
<td>Suppress additional status information. Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td>-c</td>
<td>Print this configuration file, not the current configuration file.</td>
</tr>
<tr>
<td>--cfgfile</td>
<td>Print this configuration file, not the current configuration file.</td>
</tr>
</tbody>
</table>

**twadmin Create Policy File Mode**

Use this mode to designate an existing plain text file as the new policy file. After you specify the plain text file on the command line, the `twadmin` executable encodes, signs, and saves the new policy file.

Although you can use this mode to edit and save an existing policy file, you should use the `tripwire` policy update mode instead. When you use the `twadmin` create policy file mode, you must re-initialize the database file, because the records in the old database file no longer match the rules in the policy file. This gives tacit (and possibly incorrect) approval that the current state of the file system is an appropriate baseline for future integrity checks.
The **tripwire** update policy mode updates the policy file and the database file simultaneously, checking for possible policy violations as it goes. See page 44 for more information on updating the policy file.

The short and long forms of the command are as follows:

```
twadmin -m P [options] policy_file.txt
twadmin --create-polfile [options] policy_file.txt
```

where *options* includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v/--verbose</td>
<td>Display additional status information. Mutually exclusive with --silent.</td>
</tr>
<tr>
<td>-s/--silent</td>
<td>Suppress additional status information. Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td>-c/--cfgfile</td>
<td>Use this configuration file, instead of tw.cfg.</td>
</tr>
<tr>
<td>-S/--site-keyfile</td>
<td>Use this site key file to sign the new policy file. Mutually exclusive with --no-encryption.</td>
</tr>
<tr>
<td>-p/--polfile</td>
<td>Create the new policy file here, instead of the location specified in the configuration file.</td>
</tr>
<tr>
<td>-e/--no-encryption</td>
<td>Do not sign the policy file. The policy file is still stored in a binary-encoded form and is not human-readable. Mutually exclusive with --site-keyfile and --site-passphrase.</td>
</tr>
<tr>
<td>-Q/--site-passphrase</td>
<td>Use this passphrase with the site key to sign the policy file. Mutually exclusive with --no-encryption.</td>
</tr>
<tr>
<td><em>policy file.txt</em></td>
<td>The text policy file that becomes the new binary-encoded and signed policy file.</td>
</tr>
</tbody>
</table>
**twadmin Print Policy File Mode**

Use this mode to print the current policy file in plain text. The policy file is stored in a binary-encoded form.

The short and long forms of the command are as follows:

```
twadmin -m p [options]
twadmin --print-polfile [options]
```

where *options* includes the following options and arguments:

- `-v` *--verbose*  
  Display additional status information. Mutually exclusive with `--silent`.

- `-s` *--silent*  
  *--quiet*  
  Suppress additional status information. Mutually exclusive with `--verbose`.

- `-c` *--cfgfile*  
  *configuration file*  
  Use this configuration file, instead of `tw.cfg`.

- `-p` *--polfile*  
  *policy file*  
  Print this policy file, instead of the policy file specified in the configuration file.

- `-S` *--site-keyfile*  
  *site key file*  
  Use this site key file.

**twadmin Remove Encryption Mode**

Use this mode to remove cryptographic signatures from configuration, policy, database, or report files. Removing the signature from a file is the first step in changing the key file and passphrase for a file (page 46). You can specify multiple files on the command line and use wildcards to specify files.

To remove encryption from Tripwire files, you need to enter the appropriate local or site passphrase, or both, to remove encryption from a combination of files. Even with cryptographic signatures removed, these files are in a binary-encoded form that is unreadable.
Command Reference

The short and long forms of the command are as follows:

```
twadmin -m R [options] file1 file2 ...
twadmin --remove-encryption [options] file1 file2 ...
```

where `options` includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-v</code>/---verbose</td>
<td>Display additional status information. Mutually exclusive with <code>-s</code>/--silent.</td>
</tr>
<tr>
<td><code>-s</code>/---silent/---quiet</td>
<td>Suppress additional status information. Mutually exclusive with <code>-v</code>/---verbose.</td>
</tr>
<tr>
<td><code>-c</code>/---cfgfile</td>
<td>Use this configuration file, instead of <code>tw.cfg</code>.</td>
</tr>
<tr>
<td><code>-L</code>/---local-keyfile</td>
<td>Specify the local key file to use when removing signatures for database files and reports.</td>
</tr>
<tr>
<td><code>-S</code>/---site-keyfile</td>
<td>Specify the site key file to use when removing signatures for configuration and policy files.</td>
</tr>
<tr>
<td><code>-P</code>/---local-passphrase</td>
<td>Specify the passphrase to use with the local key file when removing signatures from database files and reports.</td>
</tr>
<tr>
<td><code>-Q</code>/---site-passphrase</td>
<td>Specify the passphrase to use with the site key file when removing signatures from configuration and policy files.</td>
</tr>
</tbody>
</table>

`file1 file2...` Remove signatures from these files. Specify at least one file, and separate multiple files with spaces. You can use wildcards to specify files, but wildcard use is discouraged for security reasons.
## twadmin Encrypt a File Mode

Use this mode to sign Tripwire files cryptographically. This is the last step in changing the key file and passphrase for a file (page 46). You can only sign files that are not currently signed. You can specify multiple files on the command line and use wildcards to specify files. The `twadmin` command uses either the site or local key to sign the files, as appropriate for the file type. To automate the process, you can include the passphrase for the key files on the command line.

The short and long forms of the command are as follows:

```
twadmin -m E [options] file1 file2 ...
twadmin --encrypt [options] file1 file2 ...
```

where `options` includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-v</code></td>
<td>Display additional status information. Mutually exclusive with <code>--silent</code>.</td>
</tr>
<tr>
<td><code>--verbose</code></td>
<td>Display additional status information. Mutually exclusive with <code>--silent</code>.</td>
</tr>
<tr>
<td><code>-s</code></td>
<td>Suppress additional status information. Mutually exclusive with <code>--verbose</code>.</td>
</tr>
<tr>
<td><code>--silent</code></td>
<td>Suppress additional status information. Mutually exclusive with <code>--verbose</code>.</td>
</tr>
<tr>
<td><code>-c</code></td>
<td>Use this configuration file, instead of <code>tw.cfg</code>.</td>
</tr>
<tr>
<td><code>--cfgfile</code></td>
<td>Use this configuration file, instead of <code>tw.cfg</code>.</td>
</tr>
<tr>
<td><code>-L</code></td>
<td>Specify the local key file to use when signing database files and reports.</td>
</tr>
<tr>
<td><code>--local-keyfile</code></td>
<td>Specify the local key file to use when signing database files and reports.</td>
</tr>
<tr>
<td><code>-S</code></td>
<td>Specify the site key file to use when signing configuration and policy files.</td>
</tr>
<tr>
<td><code>--site-keyfile</code></td>
<td>Specify the site key file to use when signing configuration and policy files.</td>
</tr>
<tr>
<td><code>-P</code></td>
<td>Specify the passphrase to be used with the local key file when signing database files and reports.</td>
</tr>
<tr>
<td><code>--local-passphrase</code></td>
<td>Specify the passphrase to be used with the local key file when signing database files and reports.</td>
</tr>
<tr>
<td><code>-Q</code></td>
<td>Specify the passphrase to be used with the site key file when signing configuration and policy files.</td>
</tr>
<tr>
<td><code>--site-passphrase</code></td>
<td>Specify the passphrase to be used with the site key file when signing configuration and policy files.</td>
</tr>
<tr>
<td><code>file1 file2...</code></td>
<td>Sign these files using the site or local key, depending on the file type. Specify at least one file, and separate multiple files with spaces. You can use wildcards to specify files, but wildcard use is discouraged for security reasons.</td>
</tr>
</tbody>
</table>
twadmin Examine Encryption Mode

Use this mode to determine the encryption status of Tripwire files. This mode displays the following information for each file:

- the filename
- the file type and version number
- the key, if any, used to sign the file

The short and long forms of the command are as follows:

```
twadmin -m e [options] file1 file2 ...
twadmin --examine [options] file1 file2 ...
```

where `options` includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v</td>
<td>Display additional status information. Mutually exclusive with --silent.</td>
</tr>
<tr>
<td>-s</td>
<td>Suppress additional status information. Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td>-c</td>
<td>Use this configuration file, instead of <code>tw.cfg</code>.</td>
</tr>
<tr>
<td>-L</td>
<td>Specify the key to use as a local key when examining database or report files.</td>
</tr>
<tr>
<td>-S</td>
<td>Specify the key to use as a site key when examining policy or configuration files.</td>
</tr>
<tr>
<td>file1 file2...</td>
<td>List of Tripwire files to examine. Specify at least one file, and separate multiple files with spaces. You can use wildcards to specify files, but wildcard use is discouraged for security reasons.</td>
</tr>
</tbody>
</table>
twadmin Generate Keys Mode

Use this mode to create new site or local key files for Tripwire files. You can use these key files to change the keys and passphrases that sign Tripwire files (page 46).

You can generate site and local keys simultaneously, or generate them individually with two separate invocations of twadmin.

**Warning:** Always make backup copies of key files before generating new keys. Whenever you overwrite a site or local key file, any files signed with that key become permanently unusable. Tripwire, Inc. cannot help you recover such files.

The short and long forms of the command are as follows:

```
twadmin -m G [options]
twadmin --generate-keys [options]
```

where *options* includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v</td>
<td>Display additional status information. Mutually exclusive with --silent.</td>
</tr>
<tr>
<td>--verbose</td>
<td>Display additional status information. Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td>-s</td>
<td>Suppress additional status information. Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td>--silent</td>
<td>Suppress additional status information. Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td>--quiet</td>
<td>Suppress additional status information. Mutually exclusive with --verbose.</td>
</tr>
<tr>
<td>-L</td>
<td>Create a new local key file here. You must specify either --local-keyfile, --site-keyfile, or both.</td>
</tr>
<tr>
<td>--local-keyfile</td>
<td><em>local key file</em></td>
</tr>
<tr>
<td>-S</td>
<td>Create a new site key file here. You must specify either --local-keyfile, --site-keyfile, or both.</td>
</tr>
<tr>
<td>--site-keyfile</td>
<td><em>site key file</em></td>
</tr>
<tr>
<td>-P</td>
<td>Use this passphrase when generating a local key.</td>
</tr>
<tr>
<td>--local-passphrase</td>
<td><em>local passphrase</em></td>
</tr>
<tr>
<td>-Q</td>
<td>Use this passphrase when generating a site key.</td>
</tr>
<tr>
<td>--site-passphrase</td>
<td><em>site passphrase</em></td>
</tr>
</tbody>
</table>
siggen

Use the `siggen` utility to display hashes for any specified file. Look for more detailed information about each of the signature functions supported by Tripwire software in the glossary.

`siggen` displays one or more hash values for files in base 64 notation. Commercial versions of Tripwire use a different base 64 notation than Tripwire academic source releases, so signature values for the same file may be different for different versions.

The format for `siggen` is as follows:

```
siggen [options] file1 file2 ...
```

where `options` includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-t --terse</td>
<td>Print requested signatures for a given file on one line, delimited by spaces, one file per line.</td>
</tr>
<tr>
<td>-h --hexadecimal</td>
<td>Display results in hexadecimal rather than base 64 notation.</td>
</tr>
<tr>
<td>-a --all</td>
<td>Display all signature function values (default).</td>
</tr>
<tr>
<td>-C --CRC32</td>
<td>Display CRC-32, POSIX 1003.2 compliant 32-bit Cyclic Redundancy Check.</td>
</tr>
<tr>
<td>-M --MD5</td>
<td>Display MD5, the RSA Data Security, Inc.® Message Digest Algorithm.</td>
</tr>
<tr>
<td>-S --SHA</td>
<td>Display SHA value.</td>
</tr>
<tr>
<td>-H --HAVAL</td>
<td>Display HAVAL value, a 128-bit signature code.</td>
</tr>
</tbody>
</table>

Generate signatures for these files. You can use wildcards to specify files, but wildcard use is discouraged for security reasons.
**twagent**

Use the `twagent` command to control the operation of the Tripwire Agent, which Tripwire software uses to communicate with the Tripwire Manager. If you are not using Tripwire Manager, you do not need to use the `twagent` command.

With the `twagent` command, you can edit or print the Agent configuration file, start the Tripwire Agent daemon (for UNIX systems), or add or remove the Tripwire Agent service from the Services list (for Windows systems).

**twagent Create Agent Configuration File Mode**

Use this mode to designate an existing plain text file as the new Agent configuration file. After you specify the plain text file on the command line, the `twagent` executable encodes, signs, and saves the new configuration file.

The short and long forms of the command are as follows:

```
twagent -m F [options]
twagent --create-cfgfile [options]
```

where `options` includes the following options and arguments:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-S, --site-keyfile</td>
<td><code>site key file</code> Use this site key file to sign the new Agent configuration file. Mutually exclusive with the <code>--no-encryption</code> option. You must specify either <code>--no-encryption</code> or <code>--site-keyfile</code>.</td>
</tr>
<tr>
<td>-Q, --site-passphrase</td>
<td><code>site passphrase</code> Use this passphrase with the site key to sign the Agent configuration file. Valid only with <code>--site-keyfile</code>.</td>
</tr>
<tr>
<td>-c, --cfgfile</td>
<td><code>configuration file</code> Create the new Agent configuration file here.</td>
</tr>
</tbody>
</table>

Twagent -m F [options]
Twagent --create-cfgfile [options]
**twagent Print Agent Configuration File Mode**

Use this mode to print the current Agent configuration file in plain text. The Agent configuration file is stored in a binary-encoded form.

The short and long forms of the command are as follows:

- `twagent -m f [-c Agent_configuration_file]`
- `twagent --print-cfgfile [--cfgfile Agent_configuration_file]`

**twagent Start mode**

Use this command to start the Tripwire Agent daemon on a UNIX machine. The short and long forms of the command are as follows:

- `twagent -m S`
- `twagent --start`
**twagent Install Mode**

Use this command to install the Tripwire Agent as a Windows service. The short and long forms of the command are as follows:

```
twagent -m i [options]
twagent --install [options]
```

where `options` includes the following options and arguments:

- `-e` `--executable-path` `path` Full path to the Tripwire Agent executable. If no path is specified, the executable path is set to the path of the directory of the calling Tripwire Agent.

- `-u` `--user` `username` User that Tripwire Agent will use, where `username` is `domain_name\user_name`, or `.\user_name` for an account in the built-in domain. The default value is `.\SYSTEM`.

- `-p` `--password` `password` The login password for the user specified with `--user`. The default value is no password.

- `-s` `--startup` `{auto|manual|disabled}` The startup disposition for the Tripwire Agent. The default value is `manual`.

**twagent Remove Mode**

Use this mode to remove the Tripwire Agent service from the Windows Services list. The short and long forms of the command are as follows:

```
twagent -m R
twagent --remove
```
Glossary

asymmetric cryptography

A type of cryptographic system that uses public and private keys, for encryption and decryption of information.

attribute

In the policy file, attributes modify the behavior of rules. Attributes allow you to associate a name or numeric severity level to a rule, or to send e-mail if the rule is violated.

checksum

A value computed, via some parity or hashing algorithm, for information that requires protection against error or manipulation. Checksums are stored or transmitted with data and are intended to detect data integrity problems.

configuration file

A Tripwire file that stores information and settings, including the paths to files, and default settings for integrity checks and other operations. The configuration file is encoded and signed with the site key file, and you must specify the site passphrase to change this file.

CRC-32 algorithm

A Cyclic Redundancy Check algorithm. This is a fast, robust algorithm that detects data transmission errors reliably. CRC-32 is well understood and consequently is a fast, but insecure, alternative to the slower message-digest algorithms. CRC-32 generates a 32-bit signature.

create configuration file mode

A twadmin command that signs a plain text file and saves it as the configuration file.
create policy file mode

A **twadmin** command that signs a plain text file and saves it as the Tripwire policy file.

damage assessment and recovery

The process of determining the extent and severity of damage after an intrusion. Tripwire integrity systems allow you to quickly see what has changed, and sort the changes based on importance or functional characteristics. This saves time and recovery resources.

database file

A Tripwire file representing a snapshot of a system that serves as the baseline for integrity checks. The database file is used for most Tripwire operations, and should be created from a system in a known secure state. The database file is encoded and signed with the local key file, and you must specify the local key file to update it.

database initialization mode

A **tripwire** command that uses the rules in the current policy file to generate the Tripwire database file.

database update mode

A **tripwire** command that updates the objects in the Tripwire database file with the data from a report file.

directive

In the policy file, a language element that begins with @@ and defines a section (@@section), applies policy rules conditionally (@ifhost,@@ifelse, and @@endif), or marks the logical end of the file (@end).
**encryption mode**

A `twadmin` command that signs Tripwire files using the site or local key.

**escape sequence**

A character sequence that introduces a special-case interpretation of functional characters or sequences. Escape sequences can also be used to represent nonprintable characters.

**examine encryption mode**

A `twadmin` command that examines Tripwire files and displays the filename, file type, whether the file is signed, and what key, if any, was used to sign it.

**generate keys mode**

A `twadmin` command that creates site or local keys for Tripwire files.

**global variable**

A variable you define in the `@@GLOBAL` section of the policy file and use in any section that follows. If a local variable and a global variable have the same name, the local section uses the local variable definition.

**hash**

The value that a hash algorithm calculates. A simple hash is sometimes called a checksum, and a one-way hash is sometimes called a message digest.
Glossary

**HAVAL algorithm**

A one-way hash algorithm for high security. It was written by Yuliang Zheng at the University of Wollongong and is described in the following document:


As shipped with Tripwire for Servers, HAVAL is configured with a 128-bit signature using four passes to ensure pseudo-random output.

**host-based intrusion detection**

Strategy of collecting information about changes to machines to detect intrusions or policy violations.

**integrity check**

A Tripwire for Servers operation that compares the last known properties of a system object to the current properties to see if there are changes.

**integrity check mode**

A tripwire command that compares the last known properties of an object to the current properties to see if there are any violations.

**key files**

Files that hold the public and private keys that Tripwire for Servers uses to sign files and verify signatures. Tripwire software uses two key files, the site key file and the local key file. If either of the key files are overwritten or otherwise destroyed, any files signed with those keys will be unusable. See the Appendix for more information on key files.
**local key file**

A file containing the keys that Tripwire for Servers uses to sign and verify the database file and (optionally) report files. You must specify the local passphrase to write to a file protected with the local key file.

**local variable**

In the policy file, a variable you define in the file system or registry sections, whose scope is limited to that section. If a local variable and a global variable have the same name, the local section uses the local variable.

**MD5 algorithm**

A one-way hash algorithm created by RSA Data Security Inc. and a proposed data authentication standard for high security. The Internet draft submission, Internet working draft RFC 1321, is available from [http://www.merit.edu/internet/documents](http://www.merit.edu/internet/documents). The MD5 algorithm generates a 128-bit signature that uses four passes to ensure pseudo-random output.

**message-digest algorithm**

A type of algorithm used to render files tamper-evident. A small change to an input data file will cause a large change to the message digest value for that file.

**network-based intrusion detection**

A class of intrusion detection tools that detect intrusions by looking for anomalous patterns of network traffic.

**object name**

In a policy file, the name of an object that Tripwire software monitors. The object name is the first element of a rule.
Glossary

**passphrases**

Long passwords which Tripwire for Servers uses to generate site and local keys. It then uses the keys to sign files. Once a file is signed, you must know the appropriate passphrase to update it.

**policy compliance**

Using Tripwire software to detect changes to the configuration of a system that violate corporate IT policy.

**policy file**

A file containing rules for checking system objects on a computer. Each rule in the policy file specifies a system object to be monitored, and describes which changes to the object should be reported, and which ones can safely be ignored. The policy file is encoded and signed with the site key file, and you must specify the site passphrase to change it.

**predefined variable**

A named set of properties that you can declare and use as a variable in a policy file rule.

**print configuration file mode**

A `twadmin` command that prints the current contents of the configuration file in a readable text format.

**print policy file mode**

A `twadmin` command that prints the current contents of the policy file in a readable text format.

**private key**

A component of Tripwire site and local key files that signs files.
property

A characteristic (e.g. file size, last access time, user permissions) of a system object that Tripwire software can monitor.

public key

A component of Tripwire site and local key files that verifies files that are signed.

recursion level

An optional level of subdirectory scanning for a policy file rule. You specify the level with the `recurse` attribute, choosing to scan only the starting directory or registry key, scan from the starting point through all subdirectories, or scan down to a particular level.

remove encryption mode

A `twadmin` command that removes cryptographic signatures from configuration, policy, database, and report files.

report file

A Tripwire file that presents the results of an integrity check violation.

rule

A policy file statement that specifies which system objects to scan and which object properties to include or exclude during integrity checks. A rule often specifies optional attributes as well. There is only one rule for each object and each rule ends with a semicolon.

rule attribute

An optional part of a policy file rule that specifies the rule’s name (`rulename`), the object’s recursion level (`recurse`), the rule’s severity level (`severity`), or an e-mail address for violation notices (`emailto`).
Glossary

**rule block**

A set of policy file rules that share common rule attributes.

**rule name**

An optional name for a policy file rule or block of rules. You specify the name with the `rule_name` attribute.

**section**

A part of a policy file defined with an `@@section` directive. A policy file for a UNIX system has an optional global section (`@@section GLOBAL`) and a file system section (`@@section FS`). A policy file for a Windows system has an optional global section (`@@section GLOBAL`), a file system section (`@@section NTFS`) and a registry section (`@@section NTREG`).

**severity level**

A numeric value (from 0 to 1000000, with 0 as the lowest) for the importance of a policy file rule. You specify the level with the `severity` attribute. If no severity level is specified, it defaults to 0.

**SHA/SHS algorithm**

An algorithm for high security. SHS is the NIST Digital Signature Standard, called the Secure Hash Standard, and is described in NIST FIPS 180. It is referred to here as the SHA, or Secure Hash Algorithm, because Tripwire for Servers uses a non-certified implementation and cannot claim standards conformance. SHS generates a 160-bit hash.

**signed file**

A Tripwire policy file, configuration file, database file, or optionally, report file, that Tripwire for Servers signs using appropriate site and local keys. You must specify the site or local passphrase to write to a signed file.
**site key file**

A file containing the keys that Tripwire for Servers uses to sign and verify the configuration and policy files. You must specify the site passphrase to write to a file protected with the local key file.

**stop point**

In a policy file, a rule that specifies objects to ignore during an integrity check. A ! symbol marks a stop point.

**system object**

A file, directory, or Windows registry key or value that Tripwire software monitors. Tripwire for Servers monitors system objects according to rules in the policy file.

**violation**

An addition, deletion, or modification to a system object that violates a rule in the Tripwire policy file.
Index

A
adding new rules 43
Agent configuration file
defined 8, 48
default location 50, 53
editing 49, 52, 81
printing 82

C
changes for this version 10
changing passphrases 46
commands
overview 57
command syntax 57
command-line help 58
siggen 80
triwire 59–68
twadmin 71–79
twagent 81–83
twprint 68–70
wildcards 59
configuration file
defined 8
editing 15, 72
printing 73
testing e-mail settings 19
configuring e-mail 17
configuring Tripwire for Servers 15–27
overview 15
creating a new policy file 20
cryptography, Tripwire implementation 9
Index

D
data and network integrity 4
database file
  defined 8
  initializing 23, 60
  printing 70
  problems when updating 42
  security modes 43
  updating 40, 64
  updating specific entries 41
default policy file 20

E
editing
  Agent configuration file 49, 52, 81
  configuration file 15, 72
  policy file 22, 43
e-mail
  configuration file options 17
  requirements 34
  specifying report level on the command line 34
  testing 19, 67
encryption
  adding to Tripwire files 77
  checking for Tripwire files 78
  removing from Tripwire files 75

F
files 8

G
generating hash values 80

I
initializing the database file 23, 60
  integrity check 31–39
overview 31
command-line options 61
for specific objects only 36
ignoring properties 37
scheduling 38
selective integrity checks 34
specifying report filename 33
specifying rules to check 35
using rule blocks 35
with the severity attribute 36
intrusion detection with Tripwire software 4

K
key files
derived 9
changing passphrases 46
generating 79
keys
private key 9
public key 9

L
local key file 9
log file reporting
configuring 17

N
new features for this version 10

P
passphrases 9
cautions about overwriting 46
changing 46
policy file
defined 8
creating 20, 73
default 20
Index

  default location 23, 44
  editing 22
  editing vs. updating 22, 73
  for HPUX and AIX 20
  obtaining from the Tripwire for Servers CD 21
  Policy Resource Center website 21
  printing 75
  resolving conflicts when updating 45
  tuning to reduce noise 24
  update process 44
  updating 43, 66
  Policy Resource Center website 21
  printing
    Agent configuration file 82
    configuration file 73
    database file 70
    policy file 75
    report files 68, 70
  private key 9
  procedures
    adding Tripwire Agent to the Windows Services list 50
    changing the local key file 47
    changing the site key file 46
    configuring e-mail reporting 17
    configuring Event Log reporting 18
    configuring SNMP logging 19
    configuring syslog reporting 18
    customizing your policy file 22
    editing the Agent configuration file 49, 52
    editing the configuration file 16
    ignoring properties during an integrity check 37
    initializing the database file 24
    obtaining a policy file from the Tripwire for Servers CD 21
  printing a report file 39, 70
  removing Tripwire Agent from the Windows Services list 50
  resolving policy update problems 45
  running a basic integrity check 33
  running a selective integrity check
for specific objects 36
  using rule severity levels 36
  with a specific rule 35
scheduling integrity checks
  on UNIX systems 39
  on Windows systems 38
sending e-mail reports 34
signing and installing the policy file 23
starting or stopping Agent on Windows systems 49
testing e-mail parameters 19
tuning the policy file 24
updating specific database file entries 41
updating the database file 40
updating the policy file 44
using the Policy Resource Center website 21
public key 9

R
read-only media, using for additional security 9
report files
  defined 8
  default filename 69
  printing 68, 70
  reducing noise 24, 43
  signing 62
  specifying filename and location 33
  specifying report level on the command line 34, 39
  viewing 39
risk management 5
rulename attribute
  using in an integrity check 35

S
scheduling integrity checks 38
security issues
  deleting plain text configuration file 16
  generating a database from a known good machine 23
  maximizing key file security 9
Index

removing twadmin executable 71
resolving policy update violations 45
using read-only media 9

security mode
when updating the database file 43
when updating the policy file 45

selective integrity checks 34
severity attribute
using in an integrity check 36

siggen command 80
using to verify file integrity 9

site key file 9
SNMP
configuring 18
syslog reporting
configuring 17
system management 5

T
testing e-mail 67
Tripwire Agent 48–53
adding to the Windows Services list 50
changing authentication key 48
changing options for Windows systems 50
changing port number used 48
changing user account on Windows systems 51
installing on a Windows system 83
removing from a Windows system 83
removing from the Windows Services list 50
starting or stopping
on UNIX systems 51
on Windows systems 49
starting or stopping on UNIX systems 51
see also Agent configuration file
tripwire command 59–68
overview of command modes 59

Tripwire Files 8
Tripwire Manager 10
see also Tripwire Manager User Guide

tuning the policy file 24

twadmin command 71–79
  overview of command modes 71

twagent command 81–83

twprint command 68–70

U

UNC names
  on the command line 58

updating
  database file 40, 64
  policy file 43, 66

using Tripwire software 6
  flowcharts 7, 32
  overview 31

V

viewing report files 39

W

wildcards on the command line 59