Oracle Solaris 11.4 Compliance Guide

E61020-01
August 2021

Oracle Solaris 11.4 Compliance Guide,

E61020-01

Primary Author: Cathleen Reiher, Sharon Veach

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, la documentation du logiciel, les données (telles que définies dans la réglementation "Federal Acquisition Regulation") ou la documentation qui l'accompagne sont livrés sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

UTILISATEURS DE FIN DU GOUVERNEMENT É.-U. : programmes Oracle (y compris tout système d'exploitation, logiciel intégré, tout programme intégré, installé ou activé sur le matériel livré et les modifications de tels programmes) et documentation sur l'ordinateur d'Oracle ou autres logiciels OracleLes données fournies aux utilisateurs finaux du gouvernement des États-Unis ou auxquelles ils ont accès sont des "logiciels informatiques commerciaux", des "documents sur les logiciels informatiques commerciaux" ou des "données relatives aux droits limités" conformément au règlement fédéral sur l'acquisition applicable et aux règlements supplémentaires propres à l'organisme. À ce titre, l'utilisation, la reproduction, la duplication, la publication, l'affichage, la divulgation, la modification, la préparation des œuvres dérivées et/ou l'adaptation des i) programmes Oracle (y compris tout système d'exploitation, logiciel intégré, tout programme intégré, installé, ou activé sur le matériel livré et les modifications de ces programmes), ii) la documentation informatique d'Oracle et/ou iii) d'autres données d'Oracle, sont assujetties aux droits et aux limitations spécifiés dans la licence contenue dans le contrat applicable. Les conditions régissant l'utilisation par le gouvernement des États-Unis des services en nuage d'Oracle sont définies par le contrat applicable à ces services. Aucun autre droit n'est accordé au gouvernement américain.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Oracle®, Java, et MySQL sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut être une marque appartenant à un autre propriétaire qu'Oracle.

Intel et Intel Inside sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Epyc, et le logo AMD sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité et excluent toute garantie expresse ou implicite quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

Contents

Using This Documentation

1 Reporting Compliance to Security Standards

2 Centrally Managing Compliance Assessments

3 Administering Compliance to Critical Security Updates

A Compliance Reference

Glossary

Using This Documentation

Product Documentation Library

Documentation and resources for this product and related products are available at http://www.oracle.com/pls/topic/lookup?ctx=E37838-01.

Feedback

Provide feedback about this documentation at http://www.oracle.com/goto/docfeedback.

1

Reporting Compliance to Security Standards

What's New in Compliance in Oracle Solaris 11.4

This section highlights information for existing customers about important new compliance features in this release.

- You can run and administer compliance reports from a central system and store them on a common server. See Centrally Managing Compliance Assessments and the `compliance-roster`(8) man page.
- You can tag compliance assessments for identification and filtering. See Using Metadata to Manage Assessments and the *Match Parameters* section of the `compliance`(8) man page.
- When Oracle Solaris Cluster is configured and running, administrators can use the Solaris `Recommended` profile to verify the configuration's compliance to internal and external security requirements. For a customized list of only Oracle Solaris Cluster-related checks, use the `compliance tailor` command, as described in Creating Tailorings From Compliance Benchmarks.
- The `compliance explain` command describes in detail the compliance policy that is in effect. For more information, see the `compliance`(8) man page.

About Compliance in Oracle Solaris

Systems that comply with security standards provide more secure computing environments, and are easier to test, maintain, and protect. Oracle Solaris provides scripts that assess and report the compliance of your Oracle Solaris system to two security benchmarks: Solaris Security Benchmark and Payment Card Industry-Data Security Standard (PCI DSS).

Compliance assessment is critical for validating system compliance to external and internal security policies. The handling of security compliance and auditing requirements accounts for a large percent of IT security spending, including documentation and reports, and the validation itself. Organizations such as banks, hospitals, and governments have specialized compliance requirements. Auditors who are unfamiliar with an operating system can struggle to match security controls with requirements. Therefore, tools that map security controls to requirements can reduce time and costs by assisting auditors.

Compliance assessment is based on scripts. The scripts follow the Security Content Automation Protocol (SCAP), written in Open Vulnerability and Assessment Language (OVAL). The SCAP implementation in Oracle Solaris also supports scripts that conform to the Script Check Engine (SCE). These scripts add security checks that the current OVAL schemas and probes do not provide.

For information about the SCAP set of tools that support the `compliance` command, see the `oscap`(8) man page. To display the version of the SCAP set of tools, issue the `oscap -V` command.

Note:

The SCAP set of tools cannot localize the reports that the `oscap` command produces, nor can it localize the test descriptions.

Additional scripts can be used to meet other regulatory environment standards, such as the Gramm-Leach-Bliley Act (GLBA), the Health Insurance Portability and Accountability Act (HIPAA), Sarbanes Oxley (SOX), and the Federal Information Security Management Act (FISMA). For links to these standards, see Compliance Reference.

Security Benchmarks and Oracle Solaris

Oracle Solaris supplies compliance scripts for two standards: Solaris and PCI DSS. Independently, Center for Internet Security (CIS) provides a third-party benchmark for Oracle Solaris. You can also create customized assessments based on security benchmarks and profiles, called tailorings.

Solaris Security Policy Benchmark

The Solaris security policy benchmark is a standard based on the “secure by default” (SBD) default installation of Oracle Solaris.

The benchmark provides two profiles: `Baseline` and `Recommended`.

- The `Baseline` profile of the Solaris benchmark closely matches the default SBD installation of Oracle Solaris.
- The `Recommended` profile satisfies organizations with stricter security requirements than the `Baseline` profile. Systems that comply with the `Recommended` profile also comply with the `Baseline` profile.

The features which comprise SBD are described in Oracle Solaris Configurable Security in *Oracle Solaris 11.4 Security and Hardening Guidelines*.

The Solaris benchmark does not satisfy the requirements of the PCI DSS, CIS, or Defense Information Systems Agency-Security Technical Information Guides (DISA-STIG) benchmarks for Oracle Solaris.

PCI DSS Security Policy Benchmark

The PCI DSS security policy benchmark is a proprietary information security standard for organizations that handle cardholder information for major debit and credit cards. The standard is defined by the Payment Card Industry Security Standards Council. The intent is to reduce credit card fraud.

An Oracle Solaris system requires configuration to comply with the PCI DSS standard. The compliance report indicates which tests failed and which tests passed, and provides remediation steps. Because some PCI DSS requirements do not correspond directly to software, you must examine the compliance report and then perform additional tasks to comply with the standard.

CIS Benchmark for Oracle Solaris

The CIS standards organization provides automated compliance checking tools for its Oracle Solaris benchmark. Contact CIS to determine the cost of using CIS tools. You can use them on a Microsoft Windows system for checking Oracle Solaris compliance.

Tailorings From Benchmarks

You can create tailorings from security policy benchmarks and profiles. Tailorings customize assessments to verify the security policy of particular systems at your site. To create these customized assessments, you include or exclude rules from an existing benchmark, profile, or tailoring. To use a tailoring to assess systems, you must install the source benchmark as well as the tailoring. For more information, see Security Benchmarks and Oracle Solaris and Creating Tailorings From Compliance Benchmarks.

Enterprise Health Check Policy Benchmark

The `compliance` command and framework is a security audit capability and best-practices check that runs benchmarks and generates HTML reports for one or more systems. See the `compliance`(8) man page.

The Enterprise Health Check benchmark determines how compliant your system is with Oracle Solaris 11.4 best practices. Best practices include highlighting legacy software that is targeted for removal in an upcoming Oracle Solaris 11.4 Support Repository Update (SRU).

The following command runs the `ehc` benchmark on the local system:

```
$ pfexec compliance assess -b ehc
```

The generated assessments and reports are stored in the same location, see Listing Compliance Information and Locating Assessments and Reports. You can retrieve the any HTML reports by using RAD/REST. Each HTML report shows the state of the system and shows rule output that provides corrective actions. See Chapter 3, REST APIs for RAD Clients in *Remote Administration Daemon Client User's Guide*. Also see examples about how to access this RAD/REST API in the Managing Oracle Solaris Through REST article on GitHub.

Install the `ehc-solaris-policy` package to obtain the `ehc` benchmark by running the following command:

```
# pkg install ehc-solaris-policy
```

The `ehc-solaris-policy` package is part of the `solaris-desktop`, `solaris-small-server`, and `solaris-large-server` package groups. See Installing the Oracle Solaris OS in *Oracle Solaris 11.4 Security and Hardening Guidelines*.

Update the `ehc-solaris-policy` package periodically so that you can run the latest version of the benchmark without having to change the version of Oracle Solaris 11.4. You can update the `pkg:/security/compliance` package periodically, as well. Run the following command:

```
# pkg update ehc-solaris-policy@latest
```

compliance Command and Package

To measure security compliance, hereafter called *compliance*, requires a set of rules that define a security benchmark or profile; a measurement of compliance to that benchmark, called an *assessment*; and then a report of the findings. The report can also be printed in guide form for training or archiving purposes.

> **Note:**
>
> Consider updating the pkg:/security/compliance package periodically so that you can run the latest version of the benchmarks without having to change the version of Oracle Solaris 11.4. Run the following command:
>
> ```
> # pkg update compliance@latest
> ```

Oracle Solaris provides the compliance command to measure security compliance. The command can generate, list, and delete assessments and reports. While any user can view compliance reports, you must have rights to manage and generate assessments. For more information, see Rights to Run Compliance Assessments and Reports and the compliance(8) man page.

Many compliance commands can check remote systems as well as local systems. When you have completed Configuring Administrators to Run Remote Compliance Commands, the following compliance subcommands can run either remotely or locally:

assess
Runs a compliance assessment. See Running Assessments and Reports.

delete
Deletes the specified assessment. For examples, see Using Metadata to Manage Assessments.

explain
Lists the details of the rules in a specified benchmark or profile. See the compliance(8) man page.

get-options
Gets the default parameters of the compliance assess command.

get-policy
Shows the compliance policy that is in effect on the specified system. See Setting Policy and Assessment Options.

list
Lists the benchmarks, profiles, and rosters on a specified system. See Listing Compliance Information and Locating Assessments and Reports.

set-options
Sets the default parameters for the compliance assess command.

set-policy
Sets the default compliance policy for a specified system. See Setting Policy and Assessment Options.

The following `compliance` subcommands run on the local system only:

guide
Creates a guide of the compliance rules that are available on the system. See New Guides for New Benchmarks.

report
Shows the location of assessment reports. See Compliance Reports and Guides.

roster
Creates, modifies, and lists rosters, which are scripts that specify a set of systems to be assessed and the options of each assessment. See Running Multiple Remote Assessments.

store
Copies specified assessments, including all associated reports, to a remote assessment store. See Using a Common Store for Compliance Assessments.

tailor
Creates, modifies, and lists tailorings, which are customized sets of compliance rules. See Creating Tailorings From Compliance Benchmarks.

For mounted file systems, best practice is to separately test the compliance of the clients and the servers. For example, if you mount user home directories from central servers, run the `compliance assess` command on the user systems and on every home directory server. For how to run assessments on remote systems from a terminal window on your local system, see Running Remote Assessments on One or More Systems.

> **Note:**
>
> The `compliance` command automates compliance assessment, not remediation.

Rights to Run Compliance Assessments and Reports

Oracle Solaris provides two rights profiles to handle compliance assessment and report generation.

- The Compliance Assessor rights profile enables users to perform assessments, place them in the assessment store in report format, and delete assessments from the store.
- The Compliance Reporter rights profile enables users to locate and display existing assessments.

Compliance subcommands require the following rights:

- `compliance assess` command – Requires all privileges and the `solaris.compliance.assess` authorization. The Compliance Assessor rights profile provides these rights.
- `compliance delete` command – Requires write access to the assessment store and the `solaris.compliance.assess` authorization. The Compliance Assessor rights profile provides these rights.

- `compliance list` command – Requires read access to the assessment store to list and read assessments, reports, and any tailorings that are not yet packaged. Users with basic rights can list benchmarks, profiles, and packaged tailorings.
- `compliance report` command – Requires read access to the assessment store to generate new reports. Users who are assigned the Compliance Assessor rights profile can generate reports and read the reports. Users with the Compliance Reporter rights profile can read the reports.
- `compliance tailor` command – Requires write access to the assessment store and the `solaris.compliance.assess` authorization. The Compliance Assessor rights profile provides these rights.

`compliance` Package

The compliance rules, benchmarks, profiles, and commands are available in the `pkg:/security/compliance` package. The `solaris-small-server` and `solaris-large-server` package groups install this package.

- For information about package groups, see Installing the Oracle Solaris OS in *Oracle Solaris 11.4 Security and Hardening Guidelines*.
- For a description of the compliance package, issue the `pkg info compliance` command.

Compliance Reports and Guides

Oracle Solaris provides three report formats for compliance assessments and a guide for each benchmark and profile.

Assessment Report Formats

After you have run an assessment, the assessment directory contains a log file, a report, and a guide of your system's compliance to that specific assessment. The assessment directory files contain the following information:

- `log` – In text form, contains the results for every test that was performed for the assessment and its rule ID. The following example shows a sample entry:

  ```
  Title   The OS version is current
  Rule    OSC-53005
  Result  pass
  ```

- `report.html` – In browser-ready form, contains the results for every test that was performed for the assessment and its rule ID, time the test was run, compliance severity (high, medium, or low), description, and remediation assistance. The following example shows a sample entry:

  ```
      Result for Package integrity is verified
      Result: fail
      Rule ID: OSC-54005
      Time: 2016-09-07
        07:07
      Severity: high
        Run 'pkg verify' to check that all installed Oracle Solaris software
  matches
        the packaging database and that ownership, permissions and content are
  correct.
  ```

```
     Remediation instructions
      'pkg verify' has produced errors. Rerun the command and evaluate the errors.
       As appropriate, based on errors found,you should run 'pkg fix <package-
fmri>'
       See the pkg(1) man page.

    Remediation script
    # pkg verify
    followed by
    # pkg fix <package-fmri>

The following packages showed errors
    pkg:/library/perl-5@0.nn-11.4.n.n.n.n.n          ERROR
```

- `results.xccdf.xml` – Contains the results of every test in the benchmark. In addition to the information that is covered in `report.html`, the guide contains introductions to the areas that are assessed and references to Oracle Solaris system administration guides.

Oracle Solaris Compliance Guides

At installation, the `compliance` package provides guides to the compliance benchmarks and profiles. A guide contains the rationale for each security check and the steps to fix a failed check. Guides can be useful for training and as guidelines for future testing.

The guides that are installed with the `compliance` package are:

- *benchmark* guide – Contains every test in *benchmark*. Examples are `pci-dss`, `solaris_pci-dss`, `solaris`, and `ehc`.
- *benchmark.profile* guide – Contains every test in *benchmark*, plus a table at the end of the guide that lists which tests are selected or not selected for *profile*. Examples are `solaris.baseline` and `solaris.recommended`.

New Guides for New Benchmarks

The `compliance:generate-guide` service creates guides for each security benchmark and profile at installation. If you add a new benchmark or profile, you can create a guide for it. The following command creates guides for all installed benchmarks and profiles:

```
$ pfexec compliance guide -a
```

Administering Compliance Assessments and Reports

Compliance assessments of a benchmark or profile are comprehensive. With the `compliance` command, you can administer assessments and reports in the following ways:

- You can list compliance information on the local system and locate assessments and their reports. For more information, see Listing Compliance Information and Locating Assessments and Reports.
- You can assign a default compliance policy for each system, then regularly assess the compliance of each system by using the `scheduled` method of the `compliance:default` service. For more information, see Running Assessments at Regular Intervals.
- You can run assessments remotely. For more information, see Running Remote Assessments on One or More Systems.

- You can tag assessments with system-defined keys. You can also tag assessments with keys and values that you create. You can then list for viewing or deletion similarly tagged assessments. For more information, see the `Match Parameters` section of the `compliance`(8) man page and About Remote Assessment Features.

Listing Compliance Information and Locating Assessments and Reports

The `compliance list` command is available to all users. With the verbose `-v` and profile `-p` options, this command lists the benchmarks, their short descriptions. and their profiles. With the `-a` option, the command lists all the assessments that are stored on this system.

The `-m` *matches* parameter enables you to administer assessments and reports according to metadata. System tags are on all assessments. You can use the `-m`() *matches* option to add your own tags to selected assessments, then match those tags when listing and deleting assessments. To tag assessments with metadata, see Tagging Assessments With Metadata. For managing assessments based on their metadata, see Using Metadata to Manage Assessments.

Example 1-1   Listing All Benchmarks, Profiles, Assessments, and Reports

In this example, the administrator has specified `pci` and `recommended` on the command line as assessment names. The other assessment names were generated by the `compliance assess` command without specified assessment names.

```
$ compliance list -vp
Benchmarks:
pci-dss:        Solaris_PCI-DSS
                PCI-DSS Security/Compliance benchmark for Oracle Solaris
solaris:        Baseline, Recommended
                Oracle Solaris Security Policy

$ compliance list -a
        pci
            UUID: b83e25ee-3eec-11e6-854a-12345678abc
            UUID: 38f99dbe-3ef0-11e6-854c-12345678abc
        recommended
            UUID: 439b344a-3eef-11e6-854b-12345678abc
            UUID: 8e0b65de-3ef1-11e6-854d-12345678abc
pci-dss.2016-06-30,11:40
solaris.Recommended.2016-06-30,11:41
```

> **Note:**
>
> The `compliance list -vp` lists only those benchmarks that have been installed. So, the previous `compliance list -vp` output shows that only the `compliance` package is installed. If you also have the `ehc-solaris-policy` installed, information about the `ehc` benchmark is also shown in the `compliance list -vp` or `compliance list -b` output.

Example 1-2 Locating Files in the Compliance Repository

The reports of assessments are stored in the `/var/share/compliance/assessments` directory, also known as the repository. In this example, an administrator with the Compliance Reporter rights profile lists the locations of the latest reports for the `recommended` assessment name.

```
$ pfexec compliance report -a recommended
/var/share/compliance/assessments/8e0b65de-3ef1-11e6-854d-12345678abcd/report.html

$ pfexec compliance report -f log -a recommended
/var/share/compliance/assessments/8e0b65de-3ef1-11e6-854d-12345678abcd/log

$ pfexec compliance report -f xccdf -a recommended
/var/share/compliance/assessments/8e0b65de-3ef1-11e6-854d-12345678abcd/
results.xccdf.xml
```

Running Assessments and Reports

The `compliance` package is required to run assessments and reports. By default, the `solaris-small-server` and `solaris-large-server` packages include the `compliance` package. The `solaris-desktop` and `solaris-minimal` packages do not include the `compliance` package. To manage the assessment directories and reports in the repository requires privilege.

You can create assessment reports for benchmarks, profiles, and tailorings. For information about tailorings, see Creating Tailorings From Compliance Benchmarks. You can run a specified assessment on a system at regular intervals, as described in Running Assessments at Regular Intervals.

You can run assessments locally or remotely.

- By default, the `compliance assess` command assesses the local system.
- You can assess a remote system from your system by using the `-N` *node-URI* option, where *node-URI* specifies the remote system in remote administration daemon (RAD) URI format.
- You can assess several systems from your system by using a **roster** of host names in RAD URI format. Roster assessments run asynchronously. You use the `compliance roster` command to create and manage rosters.

By adding the `-s` *store-URI* option to the `compliance assess` command, you can transfer assessments to a common store (*store-URI*). For running remote assessments and for storing assessments in a common store, see Centrally Managing Compliance Assessments.

How to Run Assessments and Reports Locally

You must be assigned the Software Installation rights profile to add packages to the system. You must be assigned administrative rights for most compliance commands, as described in Rights to Run Compliance Assessments and Reports. For more information, see Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*.

In this procedure, you create assessment reports locally.

1. **Install the compliance package in every zone where you plan to run compliance tests.**

   ```
   $ pkg install compliance
   ```

The following message indicates that the package is installed:

```
No updates necessary for this image.
```

For more information, see the `pkg`(1) man page.

2. **Install the `pkg:/solaris/compliance/benchmark/ehc-solaris-policy` package in every zone where you plan to run the ehc benchmark.**

   ```
   $ pkg install benchmark/ehc-solaris-policy
   ```

3. **S11.4 commands, output, and explanations different from S11.3List the benchmarks and profiles that are available.**

   ```
   $ compliance list -vp
       ehc:    Standard
               Oracle Enterprise Health Check (EHC) tests
       pci-dss:        Solaris_PCI-DSS
               PCI-DSS Security/Compliance benchmark for Oracle Solaris
       solaris:        Baseline, Recommended
               Oracle Solaris Security Policy
   ```

4. **Create an assessment.**

   ```
   $ pfexec compliance assess -p profile -b benchmark -a assessment-name
   ```

   `-p` *profile*
   Indicates the name of the profile. The profile name is case sensitive.

   `-b` *benchmark*
   Indicates the name of the benchmark. The benchmark name is case sensitive.

   `-a` *assessment-name*
   Optional. Indicates the name of the assessment. The default name includes a time stamp.

   For example, the following command assesses the system using the `Recommended` profile and creates an assessment directory in the compliance repository for the assessment named `recommended`.

   ```
   $ pfexec compliance assess -p Recommended -b solaris -a recommended
   ```

   After the command completes, the reports are stored in a plain text log file named `log`, an XML file named `results.xccdf.xml`, and an HTML file named `report.html`.

   ```
   $ pfexec compliance report -a recommended
   /var/share/compliance/assessments/12345678-1111-1111-1111-12345678abcd/
   report.html
   ```

   If you run the same `compliance assess` command again, the files are not replaced. The system differentiates the assessments by `UUID`. For example:

   ```
   $ compliance list -a recommended
   recommended
       UUID: 12345678-1111-1111-1111-12345678abcd
       UUID: ab345678-1111-1111-1111-12345678abcd
   ```

5. **View the full report.**

   You can view the log file in a text editor, view the HTML file in a browser, or view the XML file in an XML viewer.

For example, to view the latest `report.html`, type the following browser entry:

```
file:///var/share/compliance/assessments/ab345678-1111-1111-1111-12345678abcd/
report.html
```

To display an earlier assessment, use its `UUID` in the browser entry, as in:

```
file:///var/share/compliance/assessments/12345678-1111-1111-1111-12345678abcd/
report.html
```

6. **Fix any failures that must pass.**
   a. **Complete the fix for the entry that failed.**
   b. **If the fix includes rebooting the system, reboot the system before running the assessment again.**

Creating Tailorings From Compliance Benchmarks

The benchmarks that Oracle Solaris provides might report failures or false positives that do not accurately reflect the compliance of particular systems. For these systems, you can create tailorings, which are inclusions or exclusions of rules from these benchmarks, or modifications of rules with variable values. Rules with variable values are explicitly marked in the interface. By modifying a variable used in a rule, you can include the rule while providing a more fine-grained expression of your security policy. You can then use these tailorings to assess the security posture of particular systems.

You create a tailoring by including or excluding rules from a benchmark, profile, or tailoring, then saving the new rule set under a different name. You can create multiple tailorings from a source benchmark and the tailorings are independent of each other. Every tailoring has a unique name.

You can use tailorings to assess the compliance of systems to a few rules or to many rules. You can save a tailoring in a form to be incorporated in an IPS package for installation on many systems. See How to Create a Package Manifest for a Tailoring.

Users who are assigned the Compliance Assessor rights profile can create tailorings and run assessments. To create a tailoring, you modify the selection of rules or modify the variable values of a rule in a benchmark or profile. You do not and cannot modify the rules themselves. Because the source of a tailoring is a particular benchmark, that benchmark must be installed on a system where you run the tailoring.

Tip:

Before creating a tailoring, print the table of contents for your source benchmark or profile. The table of contents contains the titles and numbers of the rules that you might want to exclude or include in your tailoring. Good sources for a table of contents are the guide for a benchmark or a compliance report in HTML format.

- If the tailoring will modify only a few rules from the source, start the tailoring by excluding all rules (`exclude -a`), then include the few rules you want or change the variable values of certain rules.
- Otherwise, exclude or include a few rules from the source benchmark or profile, or change the variable values of certain rules.

Editing Compliance Tailorings

The `compliance tailor` command provides two editing options, an interactive command-line editor and a curses editor, called the *pick screen*.

The pick screen is implemented in the curses programming language. It provides GUI-like functionality on a text-only device, such as a console or a hardware ANSI terminal.

How to Create a Tailoring From a Compliance Benchmark

You must be assigned the Compliance Assessor rights profile to create a tailoring that can be added to the system store. For more information, see Rights to Run Compliance Assessments and Reports and Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*.

1. **Open the compliance editor.**

   The following command sets options on the command line and opens the pick screen.

   ```
   $ pfexec compliance tailor -t basic
   *** compliance tailor: Can't get existing tailor "basic", initializing
   tailoring:basic> set benchmark=solaris
   tailoring:basic> exclude -a
   tailoring:basic> pick
   ```

   where:

   - `basic` is the name of the tailoring
   - `solaris` is the source benchmark
   - `exclude -a` loads the `solaris` benchmark with none of the rules included
   - `pick` opens the pick screen

   The pick screen displays all of the rules in the `solaris` benchmark. None of them are included.

2. **On the pick screen, use the keyboard to include particular rules, exclude rules, and navigate.**

   - The spacebar toggles between including and excluding an entry.
   - An `x` indicates an excluded rule.
   - A greater-than symbol (>) in reverse video indicates an included rule. No `x` is a second indication that the rule is included.
   - An `exit` or ESC returns you to the `compliance tailor` command line in interactive mode.

3. **Include a few basic rules.**

   For example, you might include the rules OSC-53005, OSC-16005, OSC-35000, OSC-46000, OSC-01511, OSC-04511, and OSC-75511.

4. **Display the contents of the tailoring.**

   ```
   tailoring:basic> export
   set tailoring=basic
   # version=2016-09-07T22:07:02.000+00:00
   ```

```
set benchmark=solaris
exclude -a
# OSC-53005: The OS version is current
include OSC-53005
# OSC-16005: All local filesystems are ZFS
include OSC-16005
# OSC-35000: /etc/motd and /etc/issue contain appropriate policy text
include OSC-35000
# OSC-46000: Passwords must be at least 8 characters long
include OSC-46000
# OSC-01511: Address Space Layout Randomization (ASLR) is enabled
include OSC-01511
# OSC-04511: Booting the system should require a password
include OSC-04511
# OSC-75511: Stacks are non-executable
include OSC-75511
tailoring:basic>
```

Tailorings that you create with the `compliance tailor` declare the benchmark and profile inside them.

5. **Commit your changes then exit the command-line interface.**

```
tailoring:basic> commit
tailoring:basic> exit
#
```

6. **Test the tailoring and evaluate the output.**

```
$ pfexec compliance assess -t basic
Assessment will be named 'basic.2016-09-07,07:07'
Title   The OS version is correct
Rule    OSC-53005
Result  pass
...
Title   Stacks are non-executable
Rule    OSC-75511
Result  pass
```

7. **Display the assessment report in a browser.**

   a. **Locate the assessment.**

```
# compliance report
/var/share/compliance/assessments/12345678-1111-1111-1111-12345678abcd/
report.html
```

   b. **Load the assessment into the browser.**

   The following example shows a sample browser entry:

```
file:///var/share/compliance/assessments/12345678-1111-1111-1111-12345678abcd/
report.html
```

Example 1-3 Loading a Different Tailoring

In this example, the administrator loads tailorings that are stored but not in current use.

```
$ pfexec compliance tailor
tailoring>list
basic
firsttest
testg
tailoring>load firsttest
tailoring:firsttest>info
```

```
    tailoring=firsttest
    benchmark=solaris
    profile: not set
tailoring:firsttest>load testg
tailoring:testg>
```

How to Update Tailorings Based on Previous Benchmark Versions

You must be assigned the Compliance Assessor rights profile.

You can discover that you have an outdated tailoring after updating Oracle Solaris. This procedure shows how to update or delete the tailoring.

1. **Run an assessment by using the tailoring.**

   ```
   $ pfbash ; compliance assess -t tailoring
   WARNING: version mismatch between tailoring 'tailoring'(1.nnnn) and
   benchmark 'solaris'(1.higher-nnnn), assessment test selections may not be as
   expected
   ```

2. **Review the report.**

   If the results are what you expect, then you can update the tailoring to the current benchmark or profile version. If the results do not accurately report the compliance of the system, you can modify the tailoring or create a new one.

3. **View the contents of the tailoring.**

   ```
   $ compliance tailor -t tailoring
   *** compliance tailor: WARNING: version mismatch between tailoring
   'tailoring'(1.nnnn) and
   benchmark 'solaris'(1.higher-nnnn), assessment test selections may not be as
   expected
   tailoring:basic> export
   ...
   ```

4. **To delete the tailoring, type delete in the interactive interface and confirm the deletion.**

   ```
   tailoring:basic> delete
   OK to delete tailoring 'basic' (y/N)? y
   $
   ```

5. **If needed, modify the tailoring to create the correct assessment.**

   ```
   tailoring:basic> exclude OSC-
   nnnnn
   tailoring:basic> include OSC-
   nnnnn
   ...
   tailoring:basic> export
   tailoring:basic> commit
   tailoring:basic> exit
   ```

   For a different update method, see Updating a Tailoring From an Export File.

6. **Verify that your assessment runs without error.**

   ```
   $ compliance assess -t
   tailoring
   Assessment will be named 'tailoring.YYYY-MM-DD,HH:MM
   ```

Example 1-4 Updating a Tailoring From an Export File

The administrator imports the outdated tailoring to verify that its output is accurate.

1. Using the `pfexec compliance tailor` command, the administrator opens the tailoring and exports it to a file.

   ```
   $ pfbash ;  compliance tailor -t myTailoring
   *** compliance tailor: WARNING: version mismatch between tailoring
   'myTailoring'(1.1234) and
   benchmark 'solaris'(1.2345), assessment test selections may not be as expected
   tailoring:myTailoring> export -o myTailoring1.txt
   tailoring:myTailoring> exit
   ```

2. The administrator edits the export file to rename the tailoring.

   ```
   $ pfedit myTailoring1.txt
   set tailoring=myTailoring1
   ```

3. The administrator imports the modified exported rule set.

   ```
   $ compliance tailor -f myTailoring1.txt
   tailoring:myTailoring1> commit
   tailoring:myTailoring1> exit
   ```

4. The administrator verifies that the new tailoring performs the same job as the original tailoring.

   ```
   $ compliance assess -t myTailoring1
   ...
   ```

5. The administrator deletes the outdated tailoring.

   ```
   $ compliance tailor -t myTailoring
   tailoring:myTailoring> delete
   OK to delete tailoring 'myTailoring' (y/N)? y
   $
   ```

Troubleshooting

If you are denied permission to update or delete the tailoring, either assume the `root` role, or if you have the Compliance Assessor rights profile, precede the `compliance tailor` command with `pfexec`.

How to Export a Tailoring

Exporting a tailoring lets you examine it for completeness. The export file contains comments that describe the rules that are included and excluded. You can use this file to import the tailoring on a different system for further testing. The directory to which you export the tailoring must be writable by you.

You can also use the `export` command to create a file for an IPS package of your tailoring. See How to Create a Package Manifest for a Tailoring.

1. **Load and export the tailoring.**

   The `-o` option specifies the file name. In this example, the administrator uses the `txt` file extension to indicate that the file is in plain text.

   ```
   $ pfexec compliance tailor
   tailoring>list
   basic
   testg
   ```

```
tailoring>load basic
tailoring:basic> export -o /home/jdoe/basic.tailor.txt
```

2. **When the new tailoring is ready for production, export it in XML format by using the -x option.**

   In this example, the administrator uses the `xccdf.xml` file extension to indicate that the file is in the required format for an IPS package.

   ```
   $ pfexec compliance tailor -t basic
   tailoring:basic> export -x -o /home/jdoe/basic.xccdf.xml
   tailoring:basic> exit
   ```

Example 1-5 Creating a Kerberos Tailoring From the `Recommended` Profile

In this example, the administrator creates a tailoring that includes Kerberos compliance rules. The administrator sets the source benchmark and profile and creates a tailoring from the profile plus rules that apply to Kerberos. The `export` command shows the effects of the rule inclusions and exclusions.

```
$ pfexec compliance tailor -t RKerberos
tailoring:RKerberos>set benchmark=solaris
tailoring:RKerberos>set profile=Recommended
tailoring:RKerberos>exclude OSC-28010
tailoring:RKerberos>exclude OSC-30510
tailoring:RKerberos>exclude OSC-31010
tailoring:RKerberos>exclude OSC-31510
tailoring:RKerberos>exclude OSC-63005
tailoring:RKerberos>include OSC-02511
tailoring:RKerberos>commit
tailoring:RKerberos>export
set tailoring=RKerberos
# version=2016-06-14T21:29:32.000+00:00
set benchmark=solaris
set profile=Recommended
# OSC-28010: Service svc:/network/security/kadmin:default is in disabled state
exclude OSC-28010
# OSC-30510: Service svc:/network/security/krb5_prop:default is in disabled state
exclude OSC-30510
# OSC-31010: Service svc:/network/security/krb5kdc:default is in disabled state
exclude OSC-31010
# OSC-31510: Service svc:/network/shell:kshell is disabled or not installed
exclude OSC-31510
# OSC-63005: Service svc:/network/rpc/gss is enabled if and only if Kerberos is
configured
exclude OSC-63005
# OSC-02511: The auditd(8) daemon is enabled
include OSC-02511
```

How to Create a Package Manifest for a Tailoring

After testing your new tailoring thoroughly, you can create an IPS package to install the new rules file. The **package manifest** is an early step in package creation. For the steps in creating a package, see *Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.4*.

1. **Export a thoroughly tested tailoring.**

   ```
   $ pfexec compliance tailor -t basic
   tailoring:basic> export -x -o basic.xccdf.xml
   tailoring:basic> exit
   ```

The package that you create installs this file.

2. **Create a manifest with the package name and the suffix .p5m.**

> **Tip:**
>
> Create your manifest in a working directory that will not be overwritten during updates, such as your home directory.

The following output shows a sample template for a package manifest for a tailoring. This tailoring is based on the solaris benchmark, so the tailoring package is dependent on the solaris-policy package, which installs the solaris benchmark. The items in bold in the manifest are invariant. Long lines are continued on an indented second line for ease of reading. In the manifest, the lines are not broken.

```
$ pfedit solaris-basic.p5m
set name=pkg.fmri value=pkg://
publisher-name/hierarchical-name
pkg-name@mainVersion.revision
set name=pkg.summary value="
summary
"
set name=pkg.description value="
description
"
file ./
exported-rules-file
group=
group
mode=
permissions
owner=
owner
path=usr/lib/compliance/benchmarks/solaris/tailorings/
installed-rules-file.xml
depend fmri=pkg:/security/compliance/benchmark/solaris-policy type=require
```

where:

- pkg.fmri value= specifies the full name of the package. You provide this name. The publisher name is optional. You can provide it here or when you publish the package.
- pkg.summary value= specifies the information that displays in the Summary field of the pkg info *mainVersion* command. You write the summary.
- pkg.description value= specifies the information that displays in the Description field of the pkg info *mainVersion* command. You write the description.
- file specifies where the tailoring is installed. The specification includes the source name and the installed name of the rules file for the tailoring, the directory location of the installed file without the initial slash (usr/lib/compliance/benchmarks/solaris/tailorings), and DAC permissions. The DAC permissions and location are fixed. You provide the name of the rules file that the package installs on the system. The name of the source rules file can be different from its installed version.
- depend specifies that the package that delivers the source benchmark for your tailoring will be installed on your system if it is not already installed. This entry is required.

Because `basic` tailoring is based on the `solaris` benchmark, the `solaris-policy` package will be installed on your system if it is not already installed. The `solaris-policy` package installs the directory `/usr/lib/compliance/benchmarks/solaris/tailorings` where your tailoring file is placed. To view the specification of this package, type the `pkg contents -m solaris-policy` command.

> **Caution:**
>
> In your package manifest, do not duplicate a path that has already been specified by a package that your package depends on.

3. **Create a manifest file from an existing file.**

   a. **Use the following example text as your manifest file.**

   In this sample manifest, the `solaris-basic.exportx.xml` file from the `example-IT` repository is installed as the file `basic.xccdf.xml`.

```
set name=pkg.fmri value=pkg://example-IT/security/compliance/tailorings/
solaris-basic@1.0
set name=pkg.summary value="Tailors a basic Solaris compliance
assessment for all
    systems"
set name=pkg.description value="This Solaris basic tailoring is
applicable to all
    systems, development and production. All Oracle Solaris systems are
expected
    to pass the rules in this tailoring."
file ./solaris-basic.exportx.xml group=sys mode=0555 owner=root
    path=usr/lib/compliance/benchmarks/solaris/tailorings/basic.xccdf.xml
depend fmri=pkg:/security/compliance/benchmark/solaris-policy
type=require
```

   b. **Modify the file, then save it.**

> **Note:**
>
> Be careful when typing the content of a package manifest. Make sure you join the lines that were too long to display on a single line.

Example 1-6 Creating a Package Manifest for a Compliance Package for Oracle Solaris NFS Clients

This example shows how to create a package manifest for a tailoring for NFS clients. The source name of the rules selection file is `solaris-Baseline-nfs-client.exportx.xml`. Its installed version is `nfs-client.xccdf.xml`. The tailoring is based on the `Baseline` profile of the `solaris` benchmark, so the package is dependent on the `solaris-policy` package.

1. Export the tailoring and quit the editor.

```
$ pfexec compliance tailor -t solaris-Baseline-nfs-client
tailoring:solaris-Baseline-nfs-client> export -x -o sB-nfs-client.exportx.xml
tailoring:solaris-Baseline-nfs-client> exit
```

2. Create a manifest with the package name and fill out the manifest.

```
$ pfedit /home/ooyl/packages/tailorings/solaris-Baseline-nfs-client.p5m

set name=pkg.fmri value=pkg://corporate-IT/security/compliance/tailorings/
    solaris-Baseline-nfs-client@1.0
set name=pkg.summary value="An NFS client tailoring for Solaris Baseline systems."
set name=pkg.description value="This NFS tailoring is an adjunct to the
solaris.Baseline
    profile. Assess all NFS client systems with this nfs-client tailoring."
file ./sB-nfs-client.exportx.xml group=sys mode=0555 owner=root
    path=usr/lib/compliance/benchmarks/solaris/tailorings/nfs-client.xccdf.xml
depend fmri=pkg:/security/compliance/benchmark/solaris-policy type=require
```

> **Note:**
>
> A tailoring that is installed as a package is stored in the `/usr/lib/compliance/benchmarks/`*name*`/tailorings` directory.

Example 1-7 Creating Assessments and Reports From Tailorings

In this example, an administrator has installed two tailoring packages and has a tailoring testing file. `solaris/` indicates that the installed tailoring packages are based on the `solaris` benchmark.

```
$ compliance tailor list
solaris/basic
solaris/RKerberos
testBaselinePlus
```

The Compliance Assessor administrator runs the installed tailorings assessments and views the results in a browser.

1. The administrator runs assessments for both tailorings.

```
$ pfexec compliance assess -t solaris/basic
Assessment will be named "basic.2015-11-11,10:10"
Title   The OS version is correct
Rule    OSC-53005
Result  pass
...
% compliance report
/var/share/compliance/assessments/12341111-1111-1111-1111-12345678abcd/report.html

$ pfexec compliance assess -t solaris/RKerberos
Assessment will be named "RKerberos.2015-11-11,10:20"
...
Title   Service svc:/network/rpc/gss is enabled
Rule    OSC-62511
Result  pass
...
$ compliance report
/var/share/compliance/assessments/abcd1111-1111-1111-1111-12345678abcd/report.html
```

2. The administrator views the reports by typing the following entries in a browser.

```
file:///var/share/compliance/assessments/12341111-1111-1111-1111-12345678abcd/
report.html
file:///var/share/compliance/assessments/abcd1111-1111-1111-1111-12345678abcd/
report.html
```

Next Steps

To complete the testing and delivery of this package, see *Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.4*. You should sign your tailoring packages. The packaging utility includes other attributes, such as facets, that you might want to use in the package manifest.

Selecting Alternate Values for Variables in Compliance Rules

Selecting an alternate value for a compliance rule value is called **value tailoring**. Value tailoring enables you to more finely tune your tailoring. Instead of removing a rule that does not reflect your site security requirements, you can set the rule's variable value to a value that reflects site policy. For example, you might modify the rule for password length to check for a password length of 13 characters rather than 8, or for a service that is enabled rather than disabled as in the default configuration.

> **Note:**
>
> You can modify the value of a rule only if that value is declared as a variable. Not all rules are coded with variable values.

You can display variable values and change them from the command line as shown in Creating a Tailoring That Checks for a Password Length of 13, or from the pick screen of the curses editor, as described in How to Create a Tailoring From a Compliance Benchmark.

How to Select a Non-Default Value for a Rule in a Tailoring

You must be assigned the Compliance Assessor rights profile to create a tailoring that can be added to the assessment store. For more information, see Rights to Run Compliance Assessments and Reports and Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*.

1. **Create a tailoring that modifies the value of a rule.**

```
$ pfbash ; compliance tailor -t tailoring
*** compliance tailor: Can't get existing tailor "tailoring", initializing
tailoring:tailoring> set benchmark=benchmark
```

2. **List the rules in the benchmark or profile that contain variables.**

```
tailoring:tailoring> values
OSCV-nnnnn (summary):  value
OSCV-nnnnn (summary):  value
...
#
```

3. **Change the value of a rule that contains a variable.**

```
tailoring:tailoring> include OSC-nnnnn
tailoring:tailoring> value OSCV-nnnnn=value
```

4. **Commit your changes and test.**

```
tailoring:tailoring> commit
tailoring:tailoring> exit
$ compliance assess -t tailoring
```

```
Assessment will be named 'tailoring.date'
Title   Rule title
Rule    OSC-nnnnn
Result  pass
```

Example 1-8   Creating a Tailoring That Checks for a Password Length of 13

1. Change the default password length according to site requirements.

   Change the `PASSLENGTH` value in the `/etc/default/passwd` file.

   ```
   ## /etc/default/passwd file
   ##PASSLENGTH=8
   PASSLENGTH=13
   ```

2. Create a tailoring from the `solaris` benchmark.

   ```
   $ pfbash ;  compliance tailor -t passwdLength13Test
   *** compliance tailor: Can't get existing tailor "passwdLength13Test", initializing
   tailoring:passwdLength13Test> set benchmark=solaris
   ```

3. Display the rules in the `solaris` benchmark that contain variables and their possible values.

   ```
   tailoring:passwdLength13Test> values -v
   OSCV-19500 (gdm service):  _disabled_ /disabled|enabled/
   OSCV-37500 (NFS client service):  _disabled_ /disabled|enabled/
   OSCV-46000 (Minimum Password Length):  6 <= _8_ <= 255 /6|8|14/
   OSCV-47000 (Minimum Password Character Difference):  1 <= _3_ /3/
   OSCV-48000 (Minimum Password Lower-Case Character Count):  0 <= _0_ /0|1/
   OSCV-49000 (Minimum Password Special Character Count):  0 <= _0_ /0|1/
   ```

   The output shows that the minimum password length that rule OSC-46000 can check for is `6` and the maximum is `255`. The current value is `8`.

4. Set the rule to check for a minimum password length of `13`.

   ```
   tailoring:passwdLength13Test> include OSC-46000
   tailoring:passwdLength13Test> value OSCV-46000=13
   tailoring:passwdLength13Test> commit
   tailoring:passwdLength13Test> exit
   $
   ```

5. Test the tailoring.

   ```
   $ compliance assess -t passwdLength13Test
   Assessment will be named 'passwdLength13Test.2015-10-10,10:10'
   Title   Passwords must be at least 13 characters long
   Rule    OSC-46000
   Result  pass
   ```

Running Assessments at Regular Intervals

The `compliance:default` SMF service enables you to run assessments at regular intervals. When you enable the service, it regularly runs an assessment against the system's default policy. You can modify the default policy and set a different schedule.

The `compliance:default` service is delivered disabled. To run regular assessments, you enable the service. To schedule a policy assessment that is not the default, you perform two operations:

1. Change the system's policy to a different benchmark, profile, or tailoring by running the `compliance set-policy` command.
2. Modify the assessment schedule by using the `svccfg` command to add or modify one or more `scheduled` service properties of the `compliance:default` service.

For more information about scheduled services, review the following:

- Chapter 4, Configuring Services in *Managing System Services in Oracle Solaris 11.4*
- Scheduling Executions of a Scheduled Service Start Method in *Developing System Services in Oracle Solaris 11.4*
- `svc.periodicd`(8) man page

How to Schedule a Regular Assessment of a System Using Its Default Policy

You must be assigned the Compliance Assessor rights profile to schedule assessments that can be added to the assessment store. To run the `svccfg`, you must be assigned the Service Configuration rights profile. For more information, see Rights to Run Compliance Assessments and Reports and Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*.

1. **Change the default policy to the correct policy if needed.**

   a. **List the default policy.**

   ```
   $ compliance get-policy
   Benchmark:      solaris
   Profile:        Baseline
   Tailoring:
   ```

   b. **List the available benchmarks, profiles, and tailorings.**

   ```
   $ compliance list -p
   pci-dss:        Solaris_PCI-DSS
   solaris:        Baseline, Recommended
   $ compliance list -t
           basic
           RKerberos
   ```

   You can also use the `compliance tailor list` command to list the available tailorings.

   c. **Set the correct default policy for this system.**

   In this example, you assign an existing tailoring as the default policy.

   ```
   $ pfbash ; compliance set-policy -t RKerberos
   $ compliance get-policy
   Benchmark:
   Profile:
   Tailoring:    RKerberos
   ```

2. **Before changing to a new schedule, return the schedule to the default schedule.**

   ```
   $ svccfg -s compliance:default delcust
   $ svccfg -s compliance:default listprop scheduled
    scheduled           schedule
   ```

```
 scheduled/frequency integer     1
 scheduled/interval  astring     week
```

3. **Set the new schedule and list it.**

   ```
   $ svccfg -s compliance:default setprop scheduled/property  =  type:   value
   $ svccfg -s compliance:default listprop scheduled
   scheduled           schedule
   scheduled/frequency integer     1
   scheduled/interval  astring     week
   scheduled/property
   type
   value
   $ svcadm refresh compliance:default
   ```

   Several properties are defined for scheduled services, such as `scheduled/hour` and `scheduled/day_of_week`. For examples of these properties, see Scheduling the Weekday and Hour of an Assessment and Running a Policy Assessment Daily. For more information, see How to Schedule a Periodic or Scheduled Service in *Managing System Services in Oracle Solaris 11.4* and the `svc.periodicd`(8) man page.

4. **Refresh the service.**

   ```
   $ svcadm refresh compliance:default
   ```

5. **Enable the service if it is not enabled.**

   ```
   $ svcs -x compliance:default
   svc:/application/security/compliance:default (Scheduled compliance assessment)
    State: disabled since Fri Jan  8 10:10:10 2016
   Reason: Disabled by an administrator.
      See: http://support.oracle.com/msg/SMF-8000-05
      See: compliance(8)
      See: /var/svc/log/application-security-compliance:default
   Impact: This service is not running.
   $ svcadm enable compliance:default
   $ svcs compliance:default
   STATE          STIME    FMRI
   online         10:21:22 svc:/application/security/compliance:default
   ```

6. **Verify that the initial run is scheduled.**

   ```
   $ svcs -o lrun,nrun compliance:default
   LRUN     NRUN
            Jan_08
   ```

7. **After the initial run, verify that the assessment ran.**

   ```
   $ svcs -o lrun,nrun compliance:default
   LRUN     NRUN
   02:10:10 Jan_08
   ```

8. **View the assessment in a browser.**

   a. **Locate the report.**

      ```
      $ pfexec compliance report
      /var/share/compliance/assessments/1111abcd-1111-1111-1111-12345678abcd/
      report.html
      ```

   b. **To view the report, type the file location into the browser.**

      ```
      file:///var/share/compliance/assessments/1111abcd-1111-1111-1111-12345678abcd/
      report.html
      ```

Example 1-9 Setting the Default Policy to a Benchmark or Profile

This example sets the default policy to the `Recommended` profile of the `solaris` benchmark.

```
$ compliance list -p
pci-dss:        Solaris_PCI-DSS
solaris:        Baseline, Recommended
$ compliance set-policy -b solaris -p Recommended
$ compliance get-policy
Benchmark:      solaris
Profile:        Recommended
Tailoring:
```

Example 1-10 Running the `ehc` Benchmark

This example shows you how to install the `ehc-solaris-policy` package, list the installed benchmarks, and run the `ehc` benchmark.

```
$ pkg install ehc-solaris-policy
...
$ compliance list -b
    ehc
    pci-dss
    solaris
$ compliance assess -b ehc
...
```

Example 1-11 Setting the Default Policy to an Installed Tailoring

This example sets the default policy to the `RKerberos` tailoring, which the administrator installed as a package.

```
$ compliance set-policy -b solaris -t RKerberos
$ compliance get-policy
Benchmark:    solaris
Profile:      Recommended
Tailoring:    RKerberos
```

For the contents of the `RKerberos` tailoring, see Creating a Kerberos Tailoring From the Recommended Profile.

Example 1-12 Scheduling the Weekday and Hour of an Assessment

In this example, the `root` role adds to the default schedule by specifying the day of the week and the hour that the assessment should run. After refreshing the service, `root` checks that the new schedule is valid.

```
$ pfbash ; svccfg -s compliance:default setprop scheduled/day  =  astring:
Sunday
$ svccfg -s compliance:default setprop scheduled/hour  =  integer:  2
$ svccfg -s compliance:default listprop scheduled
scheduled           schedule
scheduled/frequency integer     1
scheduled/interval  astring     week
scheduled/day       astring     Sunday
scheduled/hour      integer     2
$ svcadm refresh compliance:default
$ svcs -x compliance:default
svc:/application/security/compliance:default (Scheduled compliance assessment)
```

```
 State: online since Fri Jan 08 11:11:11 2016
 ...
```

Example 1-13 Running a Policy Assessment Daily

In this example, the `root` role changes the assessment to run daily after 2 a.m. After refreshing the service, `root` checks that the new schedule is in effect.

```
$ pfbash ; svccfg -s compliance:default setprop scheduled/interval  =  astring:  day
$ svccfg -s compliance:default setprop scheduled/hour  =  integer:  2
$ svcadm refresh compliance:default
$ svccfg -s compliance:default listprop scheduled
scheduled           schedule
scheduled/frequency integer     1
scheduled/interval  astring     day
scheduled/hour      integer     2
$ svcs compliance:default
STATE          STIME    FMRI
online          11:11:11 svc:/application/security/compliance:default
$ svcs -o lrun,nrun compliance:default
LRUN     NRUN
-        Jan_08
```

How to Return to the Default Schedule for Running Assessments

You must be assigned the Service Configuration rights profile to run the `svccfg` command. For more information, Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*.

1. **List your customizations.**

   ```
   $ pfbash ; svccfg -s compliance:default listcust
   general/enabled                    boolean     admin                 true
   scheduled/day                      astring     admin                 Sunday
   scheduled/hour                     integer     admin                 2
   periodic_restarter                 framework   admin
   periodic_restarter/scheduled       time        admin                 1111224444
   periodic_restarter/next_run        time        admin                 1111224444
   ```

2. **Return the service to the default.**

   ```
   $ svccfg -s compliance:default delcust
   $ svccfg -s compliance:default listprop scheduled
   scheduled           schedule
   scheduled/frequency integer     1
   scheduled/interval  astring     week
   ```

See Also

How to Schedule a Periodic or Scheduled Service in *Managing System Services in Oracle Solaris 11.4*

Troubleshooting

If a scheduled compliance run cannot run at its scheduled start time, future scheduled runs may also fail. To restart the schedule, clear the maintenance state, then disable and enable the `compliance:default` service instance.

```
$ svcadm clear compliance:default; svcadm disable
compliance:default; svcadm enable compliance:default
```

2

Centrally Managing Compliance Assessments

About Running and Storing Compliance Assessments Remotely

The compliance utility enables you to assess multiple systems for compliance to security requirements. Remote assessments and reports use the remote administration daemon (RAD) to securely connect to remote systems. Therefore, to remotely run and store assessments, you must configure RAD on the communicating hosts. RAD requires a URI in a particular format to identify and authenticate you to a remote system.

For a description of valid RAD URIs, see Connecting in Python to a RAD Instance by Using a URI in *Remote Administration Daemon Client User's Guide* and the rad(8) man page. To configure RAD for running and storing compliance assessments remotely, see Configuring Administrators to Run Remote Compliance Commands.

You can remotely set all the parameters that you can set for a local compliance assessment. For more information, see About Remote Assessment Features.

About Remote Assessment Features

The following parameters can be set for local and remote assessments.

- Default policy – The policy that is assessed when no benchmark, profile, or tailoring is passed to the compliance assess command. For examples of listing and setting policy, see How to Schedule a Regular Assessment of a System Using Its Default Policy.
- Metadata – Identifiers that are available or that you add to an assessment.

  Metadata or **matches** are system-defined and user-defined keys and values for tagging assessments. You can use the system keys to select assessments, and you can create your own keys and values with which to tag and select assessments. The selection mechanism supports general boolean expressions on the keys and supports selections based on time ranges. For examples, see Tagging Assessments With Metadata and Using Metadata to Manage Assessments.
- Assessment name – The following list illustrates the naming conventions.

```
basic.2016-12-29,11:11
       basic
           UUID: d0deea3e-3e56-11e6-978b-9f0b610d6a70
           UUID: c9d9c748-3e58-11e6-978d-9f0b610d6a70
       default.2016-12-29,11:15
       roster1.2016-12-29,11:10
```

  - The first name is from running a tailoring named basic.
  - The UUID names are from the basic tailoring when the name of two assessments would be identical. Typically, names are identical when the time stamps are identical.
  - The third name is from running an assessment when no argument is passed to the compliance assess command.

  - The final name is from running a roster named `roster1`.

  The default naming convention is usually sufficient. See Listing All Benchmarks, Profiles, Assessments, and Reports.

- Where the assessment is stored – The storage can be local or remote. The default is local storage, but with the `-s` *RAD-URI* option, specifies the *RAD-URI* storage location. For examples, see Using a Common Store for Compliance Assessments.
- Where policy is listed or set, features are listed or set, or assessment is run – The operation can be local or remote. The default is a local operation, but with the `-N` *RAD-URI* option, the specified *RAD-URI* can be a remote node. For examples of listing and setting features remotely, see How to Schedule a Regular Assessment of a System Using Its Default Policy.

Assessments can also be run asynchronously from a **roster**. A roster is a list of remote systems and their assessment parameters that you create to pass to the `compliance assess` command. For examples, see Running Multiple Remote Assessments and the `compliance-roster`(8) man page.

About Running Assessments Remotely

To remotely get or set policy, get or set the features of a remote assessment, and run or store an assessment or roster, you must configure authentication between the local node and the remote node. The authentication must be with a non-interactive `ssh` version of the RAD URI. Specifically, the `ssh` connection cannot prompt for a password. For instructions, see Configuring Administrators to Run Remote Compliance Commands.

The `compliance_mgr` RAD module provides programmatic access to the assessment store to create and delete, store and retrieve assessment contents, and to retrieve the assessment metadata. For syntax information, see the `compliance`(8) man page.

About Sending Assessments to a Common Store

After completing Configuring Administrators to Run Remote Compliance Commands, you can use the `-s` *RAD-URI* option to send assessments immediately or copy assessments later to a common store.

- The `compliance assess` command sends the assessment immediately to a common store.
- The `compliance store` command copies existing compliance assessments to a common store.

If the `-s` *RAD-URI* option is omitted, the assessment is stored on the same node where the assessment runs.

Configuring Administrators to Run Remote Compliance Commands

In this release, you must use Secure Shell as the RAD URI for remote compliance assessments and storage. RAD recognizes the following formats for Secure Shell:

- `ssh://`*`username`*`@`*`hostname`*`|`*`IP-address`*`:`*`port`*

- *username*@*hostname*|*IP-address*:*port*
- *hostname*|*IP-address*:*port*

username and *port* are optional. *username* defaults to the login user, and *port* defaults to the RAD IANA port, `12302`.*hostname* can be expanded to the fully qualified domain name, as in *hostname.domain.suffix*.

> **Note:**
>
> Secure Shell treats a simple host name as a separate host from its host name in FQDN format. Therefore, the RAD URIs `jdoe@host1` and `jdoe@host1.example.org` require separate authentication.

User-to-user authentication is the safest method. Host-based authentication is less secure. You can configure user-to-user in two ways: by having an LDAP user authenticate to all the clients and servers, or by creating an identical local user on all hosts, then copying each local user's public key to all hosts where the local user will run or store assessments. The LDAP method works well for a large enterprise. The local user method is useful for smaller networks and testing.

After you assign the Compliance Assessor rights profile to trusted users, the users configure a no-password-prompt `ssh` connection over RAD to every system where they will run or store assessments.

The trusted user's main steps are the following:

1. Generate a Secure Shell key pair with no passphrase.
2. In the `ldap` scope, add the user's key to the `ssh-agent` daemon on all hosts where the user will run remote commands.
3. In the `files` scope, copy the public key to the user's `.ssh/authorized_keys` file on every system.
4. On every system where the user will run the `compliance` command, verify that the `ssh` command does not prompt for a password.
5. Run remote compliance commands in a profile shell.

For detailed steps, go to the following procedures:

- How to Configure an LDAP User to Administer Remote Compliance
- How to Configure a Local User to Administer Remote Compliance

How to Configure an LDAP User to Administer Remote Compliance

You must have the rights to assign the Compliance Assessor rights profile. The `root` role or an administrator with the Compliance Assessor rights profile can assign the profile. For more information, see Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*.

1. **Assign the Compliance Assessor rights profile to the administrator who is creating and sending the assessments.**

   Because this user should be running with only the Compliance Assessor rights profile, this command replaces the user's rights profiles with the Compliance Assessor profile.

```
# usermod -K profiles="Compliance Assessor" -S ldap username
```

2. **Configure Secure Shell to authenticate username non-interactively.**

   - **As *username*, create an RSA key pair for Secure Shell authentication.**

     ```
     username@host1 $ ssh-keygen -t rsa -P ""
     Generating public/private rsa key pair.
     Enter file in which to save the key (/home/username/.ssh/id_rsa):
     Your identification has been saved in $HOME/.ssh/id_rsa
     Your public key has been saved in $HOME/.ssh/id_rsa.pub
     The key fingerprint is:
     SHA256:BLNj0v9...izsQ username@host1
     The key's randomart image is:
     +---[RSA 2048]----+
     |      o       .=B|
     ...
     ```

     The `-P ""` option indicates no passphrase for the private key. RAD authentication cannot be interactive, so do not use a passphrase.

3. **On all hosts where the user will run remote commands, add the user's key to the `ssh-agent` daemon.**

   ```
   username@host $ eval `ssh-agent`
   Agent pid nnnn
   username@host $ ssh-add
   Identity added: /home/username/.ssh/id_rsa (/home/username/.ssh/id_rsa)
   username@host $ ssh-add -l
   2048 SHA256:MZck..... /home/username/.ssh/id_rsa (RSA)
   ```

4. **Log in to the remote system by using the `ssh` command.**

   The first time you log in to a system, type `yes` when prompted to establish the authenticity of the host.

   ```
   The authenticity of host 'hostname (192.0.2.38)' can't be established.
   AB98765 key fingerprint is SHA256:tAXFy.....................
   Are you sure you want to continue connecting (yes/no)? yes
   ```

   After each remote system is known, you should not be prompted for a password.

5. **Log out and log in again to verify that you are not prompted for a password.**

   If you continue to be prompted, see the following Troubleshooting section.

> **Tip:**
>
> Add the IP address, the hostname, and the FQDN to the `/etc/ssh/ssh_known_hosts` entry for each remote system.
>
> ```
> 192.0.2.38,myx86,myx86.example.org ssh-rsa AAAAB3NzaC1yc2...
> ```

Troubleshooting

If the user continues to be prompted for a password, debug the client-server connection.

As `root`, debug the Secure Shell server. Run a command similar to the following:

```
SSHserver# /usr/lib/ssh/sshd -p 2222 -d
```

In a different terminal window and using the same port, connect as the user from the client and search for the ultimate cause of the failure.

```
SSHclient$ ssh -p 2222 SSHserver
```

Messages similar to the following can indicate the ultimate cause of the failure:

```
Authentication refused: bad ownership or modes for directory /home/ username
```

In this instance, the user has a writable group directory above the directory where the private key is stored. Change the directory permissions to 755.

How to Configure a Local User to Administer Remote Compliance

You must have the rights to create a user, assign a password, and assign the Compliance Assessor rights profile. The root role has all of these rights. For more information, see Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*.

1. **As the administrator on the local system and all remote systems, assign the Compliance Assessor rights profile to the user who is creating and sending the assessments.**

   For example, create the cpltester user on the local host.

   ```
   Local # useradd -c "Assessment Admin" -u 1111 -m -s /usr/bin/pfbash \
   -K profiles="Compliance Assessor" -S files cpltester
   Local # passwd cpltester
   New Password:
   Re-enter new Password:
   passwd: password successfully changed for cpltester
   ```

   Repeat the useradd and passwd commands exactly on the remote systems. For more information about user account options, see the useradd(8) man page.

   > **Note:**
   >
   > Local users are going to be identical except for their private keys. Each local user that is going to run or store remote assessments must generate a unique private key and send its public key to the identical user on each remote system.

2. **As the administrator, add the remote systems to each system's /etc/hosts file.**

   For example, in the /etc/hosts file on 192.0.2.111 add entries for 192.0.2.122 and all other hosts.

   ```
   ## /etc/hosts on 192.0.2.111 x86
   ::1 myx86 localhost
   127.0.0.1 myx86 localhost loghost
   192.0.2.122 mysparc
   192.0.2.113 host3
   192.0.2.114 host4
   ...
   ```

   In the file on 192.0.2.122, add entries for 192.0.2.111 and all other hosts.

   ```
   ## /etc/hosts on 192.0.2.122 sparc
   ::1 mysparc localhost
   127.0.0.1 mysparc localhost loghost
   ```

```
192.0.2.111 myx86
...
```

3. **As the `cpltester` user, configure Secure Shell to authenticate `cpltester` non-interactively.**

   These instructions are similar to the instructions for remotely administering ZFS in How to Remotely Administer ZFS With Secure Shell in *Managing Secure Shell Access in Oracle Solaris 11.4*.

   a. **On each host, create a key pair for Secure Shell authentication.**

      ```
      cpltester $ ssh-keygen -t rsa -P ""
      Generating public/private rsa key pair.
      Your identification has been saved in $HOME/.ssh/id_rsa
      Your public key has been saved in $HOME/.ssh/id_rsa.pub
      The key fingerprint is:
      SHA256:BLNj0v9...izsQ cpltester@Local
      The key's randomart image is:
      +---[RSA 2048]----+
      |       o       .=B|
      ...
      ```

      The `-P ""` option indicates no passphrase for the private key. RAD authentication cannot be interactive, so do not use a passphrase.

   b. **Copy the contents of the local `$HOME/.ssh/id_rsa.pub` to every remote system where you plan to run or store an assessment.**

      The remote system's file name for the originating system's `.ssh/id_rsa.pub` file is `.ssh/authorized_keys`.

      i. **Create your .ssh directory on every system.**

         ```
         Hostn $ cd; mkdir -m 700 .ssh
         ```

      ii. **Append the contents of id_rsa.pub to the authorized_keys file on every system.**

         Type your password when prompted. You can use the `cat >>` or the `scp` command.

         ```
         Local $ cd; cat .ssh/id_rsa.pub | ssh Remote-IP-address \
         'cat >> /export/home/cpltester/.ssh/authorized_keys && echo "Key
         copied"'
         ```

         The following command uses the `scp` command:

         ```
         Local $ cd; scp /export/home/cpltester/.ssh/id_rsa.pub \
         cpltester@Remote-IP-address:/export/home/cpltester/.ssh/
         authorized_keys
         ```

4. **Verify that the remote system does not prompt username for a password.**

   As the user, `ssh` to each remote system. You should not be prompted for a password.

   > **Tip:**
   >
   > Add the IP address, the hostname, and the FQDN to the `/etc/ssh/known_hosts` entry for each remote host.

As the user, `ssh` to each remote system using the IP address, the hostname, and the FQDN. All `ssh` connections should authenticate without a password.

Tagging Assessments With Metadata

Assessments have several tags, such as `UUID`, `Node`, and `Username`, that you can use to identify groups of assessments. You can also apply your own *keyword=value* pairs to assessments. For more examples of finding assessments by metadata, see Using Metadata to Manage Assessments.

Example 2-1 Listing Available Metadata

System tags are attached to all assessments. The `compliance list -av` *assessment* command output lists all the provided tags.

```
$ compliance list -av pci-dss.2016-09-07,07:07
            UUID: 2b333333-1111-1111-1111-9aaaabbbbb99
                Benchmark=pci-dss
                Profile=Solaris_PCI-DSS
                Status=Complete
                Node=mysparc
                Platform=cpe:/o:oracle:solaris:11.4
                Architecture=sun4v
                Timestamp=2016-09-07T11:07:07
                Username=jdoe
                UserID=j987654321d
```

Example 2-2 Adding Tags to Assessments

You can add system tags or your own keyword-value pairs as tags to assessments, then locate similarly-tagged assessments. The `compliance assess –m` *matches* adds the tags to the assessment.

```
$ compliance assess -m "trialrun=1"
Assessment will be named 'default.2016-009-07,07:07'
```

The following command finds the assessments with this tag.

```
$ compliance list -am "trialrun=1"
        1b333333-1111-1111-1111-8aaaabbbbb88
                Name=default.2016-09-07,09:09
```

The following command shows the details of the assessment that has this tag.

```
$ compliance list -av -m "trialrun=1"
        1b333333-1111-1111-1111-8aaaabbbbb88
                Name=default.2016-09-07,09:09
                Benchmark=solaris
                Profile=Baseline
                Status=Complete
                Node=mysparc
                Platform=cpe:/o:oracle:solaris:11.4
                Architecture=sun4v
                Timestamp=2016-09-07T16:09:09
                Username=jdoe
                UserID=j987654321d
                trialrun=1
```

Using a Common Store for Compliance Assessments

After assessments are stored in a common location, you can create summary reports to compare and contrast the compliance results. For security, you should be running as a user with the Compliance Assessor rights profile only when you send assessments to a common store. For the requirements and the steps, see Configuring Administrators to Run Remote Compliance Commands.

> **Note:**
>
> Some `compliance` subcommands are performed by agent processes of the Remote Administration Daemon (RAD). These processes run with the full rights available to the user. However, not all subcommands run with full rights. Prefix `compliance` commands that operate on a remote node with `pfexec` to assure that the operation is performed with the full rights available to the user.

How to Send Assessments Immediately to a Common Store

You must become an administrator who is assigned the Compliance Assessor rights profile on both hosts. For more information, see Configuring Administrators to Run Remote Compliance Commands. See also Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*.

1. **Run the assessment with the `-s` *store-URI* option.**

   The format of the command for a benchmark or a profile is the following:

   ```
   $ compliance assess [-b benchmark][-p profile][-m matches][-a assessment][-s
   store-URI]
   ```

   The format of the command for a tailoring is the following:

   ```
   $ compliance assess [-b benchmark][-t tailoring][-m matches][-a assessment][-
   s store-URI]
   ```

   where:

   matches
   Are the system or user-defined keys that add metadata to the assessment.

   store-URI
   Specifies the remote system and how to authenticate with its RAD server. The authentication must be non-interactive.

2. **On the remote system, verify that the transfer completed.**

   ```
   Remote $ compliance list -a
   ```

Example 2-3 Sending Remote Assessments to a Remote Common Store

In this example, the administrator assesses the `hostA` remote system for compliance and sends its report to `hostB` for storage. All three hosts, the invoking host, `hostA`, and `hostB` are accessible over Secure Shell without an interactive prompt. `hostA` is configured with a default compliance profile. The user who is running the `compliance assess` command can be authenticated by RAD on `hostA` and `hostB` by using the FQDN format for *host*.

```
$ pfexec compliance assess -N hostA.dev.example.com  -s hostB.archive.example.com
```

How to Copy Assessments to a Common Store

You must become an administrator who is assigned the Compliance Assessor rights profile on both systems. For more information, see Configuring Administrators to Run Remote Compliance Commands. See also Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*.

1. **Run the assessment.**

   Follow one of the procedures in Administering Compliance Assessments and Reports.

2. **Copy the assessment to a remote storage location.**

   ```
   $ pfexec compliance store [-n] [-v] [-s ssh://username@host ] {-m matches |
   assessment ...}
   ```

   where:

   `-n`
   Shows the results of running the command without running the command.

   `-v`
   Provides verbose output. For sample output, see `Example 4` in the `compliance`(8) man page.

   `-m` ***matches* | *assessment***
   Copies the assessments according the metadata of the assessments or the name of an assessment.

   If the `-s` *store-URI* option is not specified, the default *current-user@remote-host*:*RAD-port* URI is used. The original assessments remain in the local assessment store.

3. **Verify that the transfer completed.**

   - From the local system, list the compliance assessments on the remote system.

     ```
     Local $ compliance list -a -N Remote
     ```

   - On the remote system, list the compliance assessments.

     ```
     Remote $ compliance list -a
     ```

Running Remote Assessments on One or More Systems

The remote administration daemon (RAD) handles remote assessments. The `compliance` package must be installed on every host where you plan to run compliance assessments. You must complete Configuring Administrators to Run Remote Compliance Commands before running remote assessments.

You must be assigned administrative rights to run and store compliance assessments, as described in Rights to Run Compliance Assessments and Reports. For more information, see the `compliance-roster`(8) man page and the examples in this section.

Running One Remote Assessment

The `compliance assess -N` *node-URI* command enables you, as a correctly configured user, to run an assessment on a remote system. With the `-s` *store-URI* option, you can store the assessment remotely.

Example 2-4 Running a Remote Assessment and Storing It Locally

The administrator has set up a `test1` account with the Compliance Assessor rights profile to test running a remote assessment on a SPARC system and on an x86 system.

- `192.0.2.111` is the x86 system.
- `192.0.2.122` is the SPARC system.

The administrator runs the commands with various versions of *node-URI*.

The following commands run the assessments on the remote x86 system, but store them on the local SPARC system.

```
SPARC $ pfexec compliance assess -t basic -N ssh://test1@192.0.2.111
SPARC $ pfexec compliance assess -t basic -N test1@192.0.2.111
SPARC $ pfexec compliance assess -t basic -N 192.0.2.111
SPARC $ pfexec compliance assess -t basic -N myx86
```

Example 2-5 Running a Remote Assessment and Storing It Remotely

The administrator runs the commands with various versions of *node-URI*. The following commands run the assessments on the remote x86 system and store them there.

```
SPARC $ pfexec compliance assess -t basic -N ssh://test1@192.0.2.111 \
       -s ssh://test1@192.0.2.111
SPARC $ pfexec compliance assess -t basic -N test1@192.0.2.111 -s
test1@192.0.2.111
SPARC $ pfexec compliance assess -t basic -N 192.0.2.111 -s 192.0.2.111
SPARC $ pfexec compliance assess -t basic -N myx86 -s myx86
```

Running Multiple Remote Assessments

The `compliance assess -r` *example-roster* enables you to run assessments on all systems in the roster from your local system. The assessments run asynchronously, so you can continue to work on your system while the results come in. You create rosters with the `compliance roster -r` *rostername* command. For more information, see the `compliance-roster`(8) man page and the procedures in this section.

How to Create a Roster for Multiple Remote Assessments

You must become an administrator who is assigned the Compliance Assessor rights profile on both systems. For more information, see Configuring Administrators to Run Remote Compliance Commands. See also Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*.

In this procedure, you create a roster to run assessments on several remote systems at once. Rosters use the `node` parameter to identify systems.

1. **Name the roster.**

   ```
   $ pfexec compliance roster -r example-roster
   *** compliance roster: No existing roster: 'example-roster', initializing
   roster:example-roster>
   ```

2. **Add two systems to the roster.**

   ```
   roster:example-roster> add node
   roster:example-roster/node> node
   node1
   roster:example-roster/node/node1> end
   roster:example-roster> add node
   roster:example-roster/node> node
   node3
   ; end
   roster:example-roster> info; expand
   info:  roster:example-roster, 2 node(s)
     node:node1
     node:node3
   ```

3. **Commit your changes.**

   ```
   roster:example-roster> commit; list
   example-roster
   roster:example-roster> exit
   ```

4. **Specify more details for the nodes.**

   ```
   $ pfexec compliance roster -r example-roster
   roster:example-roster> select node=
   node1
   roster:example-roster/node:node1>  help structure
    The roster hierarchy consists of these object types
    and their associated properties:
       roster: (policy, match)
           node: (policy, match)
           group: (policy, match)
               node: (policy, match)
   ```

   At the node level, you can specify a policy different from the default policy. You can also tag the assessment with a *keyword-value* pair, and then later match assessments based on the tag.

5. **View the contents of the roster.**

   The `expand` subcommand in Step 2 displays the contents of the roster.

   You can also export the roster to view it in an editor.

   ```
   $ pfexec compliance roster -r example-roster
   roster:example-roster> export -o example-roster.txt
   ```

6. **Modify the contents of the roster.**

   You must be in the correct scope of the roster to change an item.

   For example, to change a node name that you mistyped, go to the node and change the node name.

   ```
   $ pfexec compliance roster -r example-roster
   roster:example-roster> select node node=node3
   roster:example-roster/node:node3> node node2; end
   roster:example-roster> commit; expand
     node:node1
     node:node2
   ```

Example 2-6   Setting Compliance Policy in a Roster

In this example, the administrator sets a policy on the node that is different from the default policy on the system.

```
$ pfexec compliance roster -r example-roster
roster:example-roster> select node=mysparc
roster:example-roster/node:mysparc> help policy
Syntax: policy [-b benchmark] [-p profile] [-t tailoring]
 sets the policy within the current scope
 The -p option can't be used with the -t option.
 Use no options to specify that this scope will inherit from an upper scope.
roster:example-roster> select node=mysparc
roster:example-roster/node:mysparc> policy -b solaris -p Recommended
roster:example-roster/node:mysparc>  end
roster:example-roster> commit
roster:example-roster> info ; expand
info:  roster:example-roster, 2 node(s)
  node:myx86
  node:mysparc       profile=Recommended benchmark=solaris
```

At the roster level, the `policy` subcommand would set the policy for all nodes in the roster that did not have an explicit policy setting at the node level. When you run the assessments using the roster, the default compliance policy that is set on the systems that are being assessed is not used.

Example 2-7   Canceling an Error in a Roster

In this example, the administrator notes the error as it is made and cancels it.

```
roster:example-roster/node:mysparc> policy -b solaris -p Baseline
roster:example-roster/node:mysparc> cancel
Canceling node modifications
roster:example-roster/node:mysparc> policy -b solaris -p Recommended
roster:example-roster/node:mysparc> info
info:  node:mysparc policy(-b solaris -p Recommended)
```

Example 2-8   Renaming a Group, Node, or Roster

In this example, the administrator renames existing rosters, groups, and nodes in the interactive editor and commits the changes.

```
roster:example-roster> select node=mysparc
roster:example-roster/node:mysparc> node mysparc1   << renamed node
roster:example-roster/node:mysparc1> end
roster:example-roster> roster myexample1            << renamed roster
roster:myexample1> select group=labsystems
roster:myexample1/group:labsystems> group labs      << renamed group
roster:myexample1/group:labs> end
roster:myexample1> commit
```

Example 2-9   Importing a Corrected Roster

In this example, the administrator found an error in a group name. Rather than recreate the group in the interactive editor, the administrator exported the roster, fixed the spelling, gave the roster a new name, imported the new version, and deleted the roster.

```
$ pfexec compliance roster -r trial1
roster:mysparc> export -o trial1.txt; exit
```

```
$ cp trial1.txt trial2.txt
pfedit trial2.txt
roster trial1
policy -b solaris -p Recommended
add group=sarc
  add node=mysparc1
    end
  add node=mysparc2
    end
  end

roster mysparcs
policy -b solaris -p Recommended
add group=sparc
  add node=mysparc1
    end
  add node=mysparc2
    end
  end
:wq

$ pfexec compliance roster -f trial2.txt
roster:mysparcs>  info ; expand
info:  roster:mysparcs policy(-b solaris -p Recommended), 1 group(s)
  node:mysparc1  profile=Recommended benchmark=solaris
  node:mysparc2  profile=Recommended benchmark=solaris
roster:mysparcs>  commit

roster:mysparcs>  list
        mysparcs
        trial1
roster:mysparcs> roster trial1
roster:trial1> delete
OK to delete roster 'trial1' (y/N)? y
$
```

How to Run Asynchronous Remote Assessments

You must become an administrator who is assigned the Compliance Assessor rights profile on both systems. For more information, see Configuring Administrators to Run Remote Compliance Commands. See also Using Your Assigned Administrative Rights in *Securing Users and Processes in Oracle Solaris 11.4*.

In this procedure, you use a roster to run assessments on several remote nodes at once.

1. **Use a roster to run assessments of the hosts in the roster.**

   If no policy was set in the roster, group, or node scope of the roster, the assessments will use the compliance policy that is set on the individual hosts.

   ```
   $ pfexec compliance assess -r example-roster
   Assessment will be named 'example-roster.YYYY-MM-DD,HH:mm'
   ```

2. **On the local system, view the progress of the assessments.**

   ```
   $ compliance list -av 'example-roster.YYYY-MM-DD,HH:mm'
   example-roster.YYYY-MM-DD,HH:mm
               UUID: ab345678-1111-1111-1111-12345678abcd
                   Benchmark=benchmark
                   Profile=profile
                   Status=Running
   ```

```
                Node=node1
                ...
```

You can also check the Status tag.

```
$ compliance list -am Status=Running
        ab345678-1111-1111-1111-12345678abcd
                Name=example-roster.YYYY-MM-DD,HH:mm
```

You can also run these commands on the remote systems.

For more information, see the compliance-roster(8) man page.

Setting Policy and Assessment Options

You can get and set compliance policy and assessment options locally. Then, when you run the command compliance assess with no options, the assessment runs with the policy and options that you have set. If you have configured RAD on your systems by completing Configuring Administrators to Run Remote Compliance Commands, you can set compliance policy and assessment options remotely.

- For examples of setting and getting local compliance policy and options, see Listing All Benchmarks, Profiles, Assessments, and Reports and How to Schedule a Regular Assessment of a System Using Its Default Policy.
- For examples of running compliance operations remotely, see Setting Up Assessments on a Remote System, Changing Remote Assessment Options, and Running a Remote Assessment With Set Options.
- For using a roster to run multiple compliance operations remotely, see How to Create a Roster for Multiple Remote Assessments.

Example 2-10 Setting Up Assessments on a Remote System

The administrator on a SPARC system shows the policy and assessment options on an x86 system, then sets the policy and options remotely before verifying the results. The remote system's assessments will be stored on the SPARC system.

The argument to the -N option can be the IP address, node name, or FQDN because all three representations of the x86 system are in the known_hosts file:
192.0.2.111,myx86,myx86.example.org ssh-rsa AAAAB3NzaC1yc2....

```
SPARC $ compliance get-policy -N myx86
Benchmark:      solaris
Profile:        Baseline
Tailoring:
SPARC $ compliance get-options -N myx86
Assessment Name:
Matches:
Store-URI:
SPARC $ pfbash compliance set-policy -b solaris -p Recommended -N myx86
SPARC $ compliance set-options -a recommended -m testing=initial -s SPARC -N
myx86
SPARC $ compliance get-policy -N myx86
Benchmark:      solaris
Profile:        Recommended
Tailoring:
SPARC $ compliance get-options  -N myx86
Assessment Name:        recommended
```

```
Matches:        testing=initial
Store-URI:      SPARC
```

Example 2-11   Changing Remote Assessment Options

The administrator on a SPARC system removes the tag from the assessments that are run on the x86 system, then verifies the results.

```
SPARC $ compliance get-options  -N myx86
Assessment Name:        recommended
Matches:        testing=initial
Store-URI:      SPARC
SPARC $ pfexec compliance set-options -m "" -N myx86
SPARC $ compliance get-options -N myx86
Assessment Name:        recommended
Matches:
Store-URI:      SPARC
```

Example 2-12   Running a Remote Assessment With Set Options

The administrator on a SPARC system runs an assessment on the x86 system that was remotely configured in the preceding examples. The output verifies that the assessment name and storage location are correctly configured.

```
SPARC $ pfexec compliance assess -N myx86
Assessment will be named 'recommended'
Remote assessment(s) will be stored via 'ssh://admin-name@myx86'
```

Using Metadata to Manage Assessments

The `compliance assess -m` *matches* command enables you to run an assessment with tags that you specify. You can later use these tags to select, list, and delete similarly-tagged assessments. For more information, see the `Match Parameters` section of the `compliance`(8) man page. The remote administration daemon (RAD) can retrieve metadata from assessments that are stored remotely.

Example 2-13   Managing Assessments by System Tags

System tags are attached to all assessments. In these examples, the administrator uses system metadata to identify and manage similar assessments.

- This command finds all assessments from the month of September 2016. The output shows the UUID and the name of the assessment.

  ```
  $ compliance list -a -m 'Timestamp>2016-08-31 & Timestamp<2016-10-01'
          471761d4-2c39-11e6-bb5e-39c6d85e0d3a
                  Name=passwdLength13Test.2016-09-07,17:57
          069cab5c-2c3c-11e6-bb60-39c6d85e0d3a
                  Name=passwdLength13Test.2016-09-17,17:17
  ```

- This command finds all assessments later than the month of August 2016 on the `192.0.2.122` remote system.

  ```
  $ compliance list -a -m 'Timestamp>2016-08-31' -N test1@192.0.2.122
  ```

- This command finds all assessments that share two tag values on a remote system.

  ```
  $ compliance list -a -m 'Timestamp>2016-08-31 & Architecture=sun4v' \
    -N test1@192.0.2.122
  ```

- This command finds all assessments from the `basic` tailoring.

```
$ compliance list -am Tailoring=basic
        282356e8-3986-11e6-9c1e-c3e78f99d41d
                Name=HereBASIC
        3989813e-3e39-11e6-9c22-c3e78f99d41d
                Name=basicRemote2
        d0deea3e-3e56-11e6-978b-9f0b610d6a70
                Name=basicRemote1
        5f0038da-3e58-11e6-978c-9f0b610d6a70
                Name=basic.2016-09-07,17:27
        c9d9c748-3e58-11e6-978d-9f0b610d6a70
                Name=basic.Remote1
```

- This series of commands finds all assessments that begin with `example`, copies them, lists the remote copies, then deletes the original assessments on the local system.

```
$ compliance list -a -m "Name>example & Name<f"
        33333333-4ea1-11e6-9691-fbfbfbfbfbfb
                Name=example-roster.2016-09-07,10:41
        11111111-4ea2-11e6-9692-abababababab
                Name=example-roster.2016-09-07,10:47
        22222222-4ea3-11e6-9693-dededededede
                Name=example-roster.2016-09-07,10:55'

$ pfexec compliance store -s mysparc -m "Name>example & Name<f"

$ compliance list -N mysparc -m "Name>example & Name<f"
Benchmarks:
        pci-dss
        solaris
Assessments:
        33333333-4ea1-11e6-9691-fbfbfbfbfbfb
                Name=example-roster.2016-09-07,10:41
        11111111-4ea2-11e6-9692-abababababab
                Name=example-roster.2016-09-07,10:47
        22222222-4ea3-11e6-9693-dededededede
                Name=example-roster.2016-09-07,10:55

$ pfexec compliance delete -n -m "Name>example & Name<f"
would delete assessment UUID=33333333-4ea1-11e6-9691-fbfbfbfbfbfb,
Name=example-roster.2016-09-07,10:41
would delete assessment UUID=11111111-4ea2-11e6-9692-abababababab,
Name=example-roster.2016-09-07,10:47
would delete assessment UUID=22222222-4ea3-11e6-9693-dededededede,
Name=example-roster.2016-09-07,10:55

$ pfexec compliance delete -m "Name>example & Name<f"
```

Example 2-14 Deleting Assessments by Metadata

In these examples, the administrator deletes assessments based on their metadata.

- This command finds all assessments that were run on SPARC systems, deletes them, and verifies the deletion.

```
$ compliance list -am Architecture=sun4v
        62622916-2dc8-11e6-9c14-c3e78f99d41d
                Name=basic.2016-07-07,15:28
        87846452-2e90-11e6-9c15-c3e78f99d41d
                Name=basic.2016-07-09,15:21
        abc11504-2e90-11e6-9c16-c3e78f99d41d
                Name=basic.2016-07-09,15:22
```

```
$ pfexec compliance delete -am Architecture=sun4v
$ compliance list -am Architecture=sun4v
$
```

- This command finds all assessments whose name begins with `default` and then deletes them.

```
$ compliance list -am "Name>default & Name<defaulu"
        94f67aae-3a24-11e6-bb79-39c6d85e0d3a
                Name=default.2016-09-07,08:59
        8142085e-52be-11e6-905f-753ff3457658
                Name=default.2016-09-25,16:21

$ pfexec compliance delete -am "Name>default & Name<default"
```

Example 2-15 Copying Assessments by Metadata

In this example, the administrator copies assessments that were run on SPARC systems and copies them to a common store. The assessments are now in two locations, the local system and the common store.

```
$ compliance list -am Architecture=sun4v
        62622916-2dc8-11e6-9c14-c3e78f99d41d
                Name=basic.2016-07-07,15:28
        87846452-2e90-11e6-9c15-c3e78f99d41d
                Name=basic.2016-07-09,15:21
        abc11504-2e90-11e6-9c16-c3e78f99d41d
                Name=basic.2016-07-09,15:22

$ pfexec compliance store -N mycommonstore -am Architecture=sun4v
```

3

Administering Compliance to Critical Security Updates

Administering CVE Updates in Oracle Solaris

Systems that contain the most recent security fixes provide a more secure computing environment. Oracle Solaris provides online access to the Common Vulnerabilities and Exposures (CVE) list and other security fixes. The `pkg` command has options to search for CVE updates.

Monitoring CVE Status in Oracle Solaris

You can monitor the status of critical updates to Oracle Solaris packages by following the information at the Oracle Critical Patch Updates, Security Alerts and Bulletins web site. You should apply critical patch updates without delay.

Locating the Packages That Have CVE Updates in Oracle Solaris

The Oracle Solaris Support package repository contains metadata for tracking security vulnerability fixes by the assigned CVE ID. Oracle Solaris creates a package of this metadata from the Oracle bug database. After installing the package, you can easily determine whether your system has all the known and required security vulnerability fixes. You do not need to derive this information from other sources. Using the Oracle bug database as your source is critically important because sometimes Oracle Solaris fixes a bug in an upstream Free and Open Source (FOSS) component by patching the code rather than by generating a new version of the component.

The metadata package from the Oracle bug database, `pkg:/support/critical-patch-update/solaris-11-cpu`, covers the `entire` dependency hierarchy. All packages that were changed for a particular CVE fix are dependencies of the `solaris-11-cpu` package. They are optional dependencies, therefore they are updated if they are already installed, but not installed if the software that is being fixed is not already installed.

The metadata package enables retrospective updates to the critical patch update (CPU) metadata where a shipped version already contains the fix for a given CVE ID. When Oracle Solaris publishes a new CPU, it also publishes a new version of the package to the Oracle Solaris support repository plus the new package versions that contain the fixes.

The version format for the CPU package is `@` *YYYY* `.` *MM-VV* where *VV* is usually a low number, as in the CPU package `solaris-11-cpu@2014.10-1`. This format enables Oracle Solaris to republish critical patch updates within the same month. Note that the day of the month (*DD*) is not part of the version format.

You can search the metadata by using either the Oracle Solaris Support package repository web site or the command-line interface. You can search for cases where a given CVE ID applies to multiple packages and also where a given package version contains fixes for multiple CVE IDs.

Installing the CPU Package

Your Oracle Solaris 11.4 systems might not have the `solaris-11-cpu` package installed by default, because this package is higher in the dependency hierarchy than the `entire` package. You must update to an SRU that includes the `solaris-11-cpu` package, install the package, then update it.

```
# pkg install recent-solaris-11-release
# pkg install solaris-11-cpu
# pkg install solaris-11-cpu@latest
```

After `solaris-11-cpu@latest` is installed, the system is updated to the SRU version of the CPU. The updating includes all package updates between the SRU of the system and the SRU version of the CPU. For more information and examples, see Applying Support Updates in *Updating Systems and Adding Software in Oracle Solaris 11.4* and Critical Patch Update Packages in *Updating Systems and Adding Software in Oracle Solaris 11.4*.

Managing CVE Updates From the Command Line

The examples in this section show how to use the command line to find CVE information.

Example 3-1 Several Ways of Listing the Packages That Contain Fixes to a CVE ID

When you know the CVE ID, you can use it to find the packages that contain the fix for it. The following searches find the fix for the `bash` Shellshock software bug.

- The `pkg search` command searches all configured repositories and the local system for the CVE ID. The output lists which packages and versions contain the fix and which CPU delivers it. Note the use of the trailing colon (`:`) in the search to indicate a missing field.

  ```
  $ pkg search CVE-2014-7187:
  INDEX          ACTION VALUE
  PACKAGE
  CVE-2014-7187 set    pkg://solaris/shell/bash@4.1.11,5.11-0.175.2.2.0.8.0
  pkg:/support/critical-patch-update/solaris-11-cpu@2015.8-1
  CVE-2014-7187 set    pkg://solaris/shell/bash@4.1.11,5.11-0.175.2.2.0.8.0
  pkg:/support/critical-patch-update/solaris-11-cpu@2015.7-3
  ...
  CVE-2014-7187 set    pkg://solaris/shell/bash@4.1.11,5.11-0.175.2.2.0.8.0
  pkg:/support/critical-patch-update/solaris-11-cpu@2014.10-1
  CVE-2014-7187 set    pkg://solaris/shell/bash@4.1.11,5.11-0.175.2.3.0.4.0
  pkg:/support/critical-patch-update/solaris-11-cpu@2014.10-1
  ```

- Without the trailing colon, the `pkg search` command lists all `solaris-11-cpu` package versions, but does not list the `bash` package that contains`the fix.

  ```
  $ pkg search CVE-2014-7187
    INDEX    ACTION VALUE          PACKAGE
  info.cve  set    CVE-2014-7187  pkg:/support/critical-patch-update/
  solaris-11-cpu@2015.8-1
  info.cve  set    CVE-2014-7187  pkg:/support/critical-patch-update/
  solaris-11-cpu@2014.4-1
  ...
  ```

```
info.cve  set    CVE-2014-7187  pkg:/support/critical-patch-update/solaris-11-
cpu@2014.10-1
```

- The following command displays the CVE ID, the package that contains the fix, and `solaris-11-cpu` package version:

  ```
  $ pkg search -Ho name,value,pkg.shortfmri CVE-2014-7187:
  CVE-2014-7187   pkg://solaris/shell/bash@4.1.11,5.11-0.175.2.2.0.8.0    pkg:/
  support/critical-patch-update/solaris-11-cpu@2015.8-1
  ...
  CVE-2014-7187   pkg://solaris/shell/bash@4.1.17,5.11-0.175.2.5.0.2.0    pkg:/
  support/critical-patch-update/solaris-11-cpu@2015.7-1
  ...
  CVE-2014-7187   pkg://solaris/shell/bash@4.1.11,5.11-0.175.2.2.0.8.0    pkg:/
  support/critical-patch-update/solaris-11-cpu@2014.10-1
  ```

- The `pkg contents -r` command searches the repository, not the local system, for the packages that fix the `bash` Shellshock software bug.

  ```
  $ pkg contents -Hro value -t set -a name=CVE-2014-7187 solaris-11-cpu
  pkg://solaris/shell/bash@4.1.11,5.11-0.175.2.2.0.8.0
  pkg://solaris/shell/bash@4.1.11,5.11-0.175.2.3.0.4.0
  pkg://solaris/shell/bash@4.1.17,5.11-0.175.2.5.0.2.0
  ```

Because SRUs and CPUs are cumulative, the fix is available after being installed once.

Example 3-2   Showing When a CVE Fix Was First Available

This example shows that the fix for the `bash` Shellshock software bug was first available for this system in the `solaris-11-cpu@2014.4-1` package and in every following SRU.

```
$ pkg search -po pkg.shortfmri CVE-2014-7187
PKG.SHORTFMRI
pkg:/support/critical-patch-update/solaris-11-cpu@2014.4-1
pkg:/support/critical-patch-update/solaris-11-cpu@2015.1-1
pkg:/support/critical-patch-update/solaris-11-cpu@2015.1-2
...
```

Example 3-3   Listing the CVE IDs in a Critical Patch Update

This example shows how to display every fixed CVE in the latest CPU.

```
$ pkg contents -rHo value -a name=info.cve solaris-11-cpu@latest
CVE-1999-0103
CVE-2002-2443
CVE-2003-0001
CVE-2004-0230
...
CVE-2015-5477
...
```

Example 3-4   Verifying That the Latest CPU Is Installed

To determine the status of the latest `solaris-11-cpu` package, use the `pkg list` command.

```
$ pkg list -af solaris-11-cpu@latest
NAME (PUBLISHER)                              VERSION                    IFO
support/critical-patch-update/solaris-11-cpu      2015.8-1                   ---
```

Because the `i` flag is not in the `I` column, the latest CPU is not installed.

Example 3-5 Verifying That a Fix for a CVE ID Is Installed

To verify that you installed a fix for a specific CVE ID, search your installed packages for the CVE ID. If it is not installed, no output displays. The `pkg search -l` command searches the local disk only.

```
# pkg search -l CVE-2014-7187
INDEX      ACTION VALUE         PACKAGE
info.cve   set    CVE-2014-7187 pkg:/support/critical-patch-update/solaris-11-
cpu@2014.10-1
```

For more information about options to the `pkg` command, see the `pkg`(1) man page.

A

Compliance Reference

Compliance Utilities

The following programs implement compliance assessment and reporting:

- Security Content Automation Protocol (SCAP)
- SCAP tools (OpenSCAP)
- Open Vulnerability and Assessment Language (OVAL)
- eXtensible Configuration Checklist Description Format (XCCDF)

The `compliance`(8) and `compliance-tailor`(8) man pages describe how to run compliance in Oracle Solaris. For command usage, type `compliance` in a shell. The following is sample output:

```
$ compliance
No command specified
Usage:  compliance subcommand [argument]...
        compliance help [subcommand],
  where subcommand is one of assess, delete, get-options, get-policy, guide,
    list, report, roster, set-options, set-policy, store, or tailor
```

Compliance Standards

The following bodies provide compliance guides, standards, or laws:

- Center for Internet Security (CIS)
- Defense Information Systems Agency-Security Technical Information Guides (DISA-STIG)
- Federal Information Security Management Act (FISMA)
- Gramm-Leach-Bliley Act (GLBA)
- Health Insurance Portability and Accountability Act (HIPAA)

  Health Information Technology for Economic and Clinical Health Act (HITECH) (Modifications to the HIPAA Rules)
- Payment Card Industry-Data Security Standard (PCI DSS)
- Sarbanes Oxley (SOX)

Glossary

assessment

In compliance terminology, a measurement of the compliance of a system against a benchmark or profile.

benchmark

In compliance terminology, a security standard against which to measure the compliance of a system. Also called a security benchmark. The `solaris` benchmark and PCI DSS are two benchmarks.

Common Vulnerabilities and Exposure

A reference method for publicly known security vulnerabilities in networked computer systems. This international list of vulnerabilities is kept by the Mitre Corporation.

Enterprise Health Check

A benchmark that determines how compliant your system is with Oracle Solaris 11.4 best practices. Best practices include highlighting legacy software that is targeted for removal in an upcoming Oracle Solaris 11.4 Support Repository Update (SRU).

metadata

In compliance terminology, a keyword=value pair or tag that identifies an assessment. Assessments always run with system tags, and users can add their own tags. Later, users can select, identify, copy, list, and delete assessments according to their metadata.

pick screen

An instance of a TUI, a textual user interface. The pick screen is an editor that is implemented in the curses programming language. It provides a GUI-like interface on a text-only device, such as a console or a hardware ANSI terminal.

policy

Generally, a plan or course of action that influences or determines decisions and actions. Specifically, compliance policy is the selection of a benchmark or profile or tailoring that reflects the security compliance requirements for a system. You check the compliance of your system against the policy.

profile

In compliance terminology, a refinement of a benchmark. The Recommended and Baseline profiles are refinements of the `solaris` benchmark.

report

In compliance terminology, a display of the results of an assessment. Compliance reports can be in text, html, or xml format.

roster

In compliance terminology, a script that identifies assessments to run on remote systems.

Systems can be collected into roster groups, and attributes such as policy can be assigned at the roster, group, and node scopes. When run, a roster assesses compliance on its included systems asynchronously from the local system.

tailoring

A customized version of a benchmark or profile against which to measure the compliance of a system. Customers create tailorings to verify the security policy of particular systems more accurately than a benchmark or profile can.

variable values

In compliance terminology, a set of alternate values for a compliance rule where the rule's value is coded as a variable. Customers can modify the variable's value in a rule in a tailoring to better assess the security posture of the system.

Index