

Oracle® Communications Session Border Controller

Administrative Security Guide



Release S-Cz8.4.0 - for Service Provider and Enterprise
F31113-13
May 2023

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

F31113-13

Copyright © 2007, 2022, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

About This Guide

My Oracle Support vi

Revision History

1 Access

Administrative Security Feature Set	1-1
Enabling the Admin Security Feature	1-2
Supported Platforms	1-2
JITC Support	1-2
Supported Platforms	1-3
Login Banner	1-3
Local User Accounts	1-3
Password Policy	1-4
Configuring Password Policy Properties	1-5
Configuring the Administrative Security with ACP Password Rules	1-7
Changing a Password	1-8
Changing Password Process	1-9
Changing the user Password	1-9
Changing the admin Password	1-10
Changing a Local Account Password	1-10
RADIUS and TACACS+ Passwords	1-12
Login Policy	1-12
Authentication and Authorization	1-15
Local Authentication and Authorization	1-15
Console Login	1-15
Serial Port Control	1-16
Initial Login	1-16
Remote SSH Login with Password	1-17
Remote SSH Login with Public Key	1-17
Two-Factor Authentication	1-19

Enable Two-Factor Authentication	1-21
RADIUS Authentication and Authorization	1-21
RADIUS Authorization Classes	1-22
RADIUS and SSH	1-23
RADIUS and Password Policies	1-23
TACACS+ Support	1-23
SSH and SFTP	1-24
SSH Operations	1-25
Configuring SSH Properties	1-25
Manage SSH Keys	1-26
SFTP Operations	1-37
Secure Radius Connection	1-38
Factory Reset for the Oracle Communications Session Border Controller	1-39
Using the Oracle Rescue Account for PNF Zeroization	1-40
Reinstalling the VM for VNF Installation	1-41

About This Guide

The Administrative Security Essentials Guide explains the concepts and procedures that support the Admin Security feature set. The feature provides a suite of applications and tools that enhance secure access, monitoring, and management of the Oracle Communications Session Border Controller (SBC).

This guide covers:

- Access authentication and authorization
- Hardware Factory Reset
- Audit logs
- JITC compliance

This publication is used with Oracle Communications Session Border Controller and Oracle Enterprise Session Border Controller.

Documentation Set

The following table describes the documentation set for this release:

Document Name	Document Description
Acme Packet 3900 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 3900.
Acme Packet 4600 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 4600.
Acme Packet 6100 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6100.
Acme Packet 6300 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6300.
Acme Packet 6350 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6350.
Release Notes	Contains information about the current documentation set release, including new features and management changes.
ACLI Configuration Guide	Contains information about the administration and software configuration of the Service Provider Session Border Controller (SBC).
ACLI Reference Guide	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.
Maintenance and Troubleshooting Guide	Contains information about SBC logs, performance announcements, system management, inventory management, upgrades, working with configurations, and managing backups and archives.

Document Name	Document Description
MIB Reference Guide	Contains information about Management Information Base (MIBs), Oracle Communication's enterprise MIBs, general trap information, including specific details about standard traps and enterprise traps, Simple Network Management Protocol (SNMP) GET query information (including standard and enterprise SNMP GET query names, object identifier names and numbers, and descriptions), examples of scalar and table objects.
Accounting Guide	Contains information about the SBC's accounting support, including details about RADIUS and Diameter accounting.
HDR Resource Guide	Contains information about the SBC's Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Administrative Security Essentials	Contains information about the SBC's support for its Administrative Security license.
SBC Family Security Guide	Contains information about security considerations and best practices from a network and application security perspective for the SBC family of products.
Installation and Platform Preparation Guide	Contains information about upgrading system images and any pre-boot system provisioning.
Call Traffic Monitoring Guide	Contains information about traffic monitoring and packet traces as collected on the system. This guide also includes WebGUI configuration used for the SIP Monitor and Trace application.
FIPS Compliance Guide	Contains conceptual and procedural information for configuration using the tools and protocols required to manage call traffic on the SBC.
HMR Resource Guide	Contains information about configuring and using Header Manipulation Rules to manage service traffic.
TSCF SDK Guide	Contains information about the client-side SDK that facilitates the creation of secure tunnels between a client application and the TSCF of the SBC.
REST API Guide	Contains information about the supported REST APIs and how to use the REST API interface.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

My Oracle Support

My Oracle Support (<https://support.oracle.com>) is your initial point of contact for all product support and training needs. A representative at Customer Access Support (CAS) can assist you with My Oracle Support registration.

Call the CAS main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <http://www.oracle.com/us/support/contact/index.html>. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select 2 for New Service Request.
2. Select 3 for Hardware, Networking, and Solaris Operating System Support.
3. Select one of the following options:
 - For technical issues such as creating a new Service Request (SR), select 1.
 - For non-technical issues such as registration or assistance with My Oracle Support, select 2.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

Emergency Response

In the event of a critical service situation, emergency response is offered by the Customer Access Support (CAS) main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at <http://www.oracle.com/us/support/contact/index.html>. The emergency response provides immediate coverage, automatic escalation, and other features to ensure that the critical situation is resolved as rapidly as possible.

A critical situation is defined as a problem with the installed equipment that severely affects service, traffic, or maintenance capabilities, and requires immediate corrective action. Critical situations affect service and/or system operation resulting in one or several of these situations:

- A total system failure that results in loss of all transaction processing capability
- Significant reduction in system capacity or traffic handling capability
- Loss of the system's ability to perform automatic system reconfiguration
- Inability to restart a processor or the system
- Corruption of system databases that requires service affecting corrective actions
- Loss of access for maintenance or recovery operations
- Loss of the system ability to provide any required critical or major trouble notification

Any other problem severely affecting service, capacity/traffic, billing, and maintenance capabilities may be defined as critical by prior discussion and agreement with Oracle.

Locate Product Documentation on the Oracle Help Center Site

Oracle Communications customer documentation is available on the web at the Oracle Help Center (OHC) site, <http://docs.oracle.com>. You do not have to register to access these documents. Viewing these files requires Adobe Acrobat Reader, which can be downloaded at <http://www.adobe.com>.

1. Access the Oracle Help Center site at <http://docs.oracle.com>.
2. Click **Industries**.
3. Under the Oracle Communications sub-header, click the **Oracle Communications documentation** link.

The Communications Documentation page appears. Most products covered by these documentation sets appear under the headings "Network Session Delivery and Control Infrastructure" or "Platforms."

4. Click on your Product and then Release Number.
A list of the entire documentation set for the selected product and release appears.
5. To download a file to your location, right-click the **PDF** link, select **Save target as** (or similar command based on your browser), and save to a local folder.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Revision History

This section provides a revision history for this document.

Date	Description
June 2020	<ul style="list-style-type: none">Initial releaseUpdates to 2FA explanation
September 2020	<ul style="list-style-type: none">Updates for public-key auth with SFTP serversRemoved old ssh-pub-key commands and updated section on SSH key management.
September 2020	<ul style="list-style-type: none">Removes IKEv2 appendicesUpdates "About This Guide" to state that this document is used with both Oracle Communications Session Border Controller and Oracle Enterprise Session Border Controller.
November 2020	<ul style="list-style-type: none">Clarifies the admin class and user classRemoves SystemAdmin RADIUS class; not supported in 8.4.0Adds process to change password for local accounts
March 2021	<ul style="list-style-type: none">Adds 'hostname' to audit file filename
June 2021	<ul style="list-style-type: none">Updates JITC descriptionACP over TCP is always blockedRemove unsupported AdminSec ACP license
June 2022	<ul style="list-style-type: none">Removes 'Manage Local Accounts'. See Configuration Guide.Adds factory reset is not a secured erase process in Factory Reset for the Oracle Communications Session Border Controller section.Adds 6350 to supported platform list.Moved Audit Log chapter to Maintenance and Troubleshooting Guide
July 2022	<ul style="list-style-type: none">Added KVM to Supported Platforms.
May 2023	<ul style="list-style-type: none">Adds TACACS+ over IPsec configuration steps.

1

Access

Administrative Security Feature Set

This section describes implications of adding and removing the Admin Security feature set on an Oracle Communications Session Border Controller (SBC).

This feature enables various security enhancements described in this document. In the absence of an Admin Security feature set, these enhancements are not available.



Note:

The Admin Security feature set is not intended for all customer use. Consult your Oracle representative to understand the ramifications of enabling these features.

If the Admin Security feature is removed, protected areas of the system remain unavailable. This ensures that a system cannot be compromised by removing features. Once the Admin Security feature is provisioned, it cannot be removed, and the SBC may retain sensitive information. To remove all sensitive data, you must perform a complete factory reset (zeroization). On supported Acme Packet platforms, zeroization is done using the Oracle Rescue Account. To perform zeroization on a virtual SBC, you must perform a complete image reinstallation. For more information on the performing a factory reset, see "Factory Reset for the Oracle Communications Session Border Controller" in this guide.



Note:

The Government Security Certification SKU is equivalent to the Admin Security feature.

When enabling the Admin Security via the **setup entitlements** command, the SBC warns the user with the following message:

```
*****
***
CAUTION: Enabling this feature activates enhanced security functions.
Once saved, security cannot be reverted without resetting the system
back to factory default state.
*****
***
Note: The 'factory default' process via the 'oracle rescue account' menu can
be used for support to guide the
removal of these features in the field by resetting the system back to the
as-shipped state.
```

When the Admin Security feature set is present and enabled, the following security policies and restrictions are implemented:

- shell access is denied
- history log access is denied
- password policy features are enabled in addition to some additional Admin Security specific password requirements

When the Admin Security feature set is disabled and deleted, the following security policies and restrictions are implemented:

- shell access is denied
- SSH keys are denied
- password policy features are disabled

Enabling the Admin Security Feature

Provision the Admin Security feature by enabling Admin Security via the **setup entitlements** command. For more information on installing the Admin Security feature set, see the *Oracle Enterprise Session Border Controller Release Notes*. For instructions on provisioning this feature set, see the *Oracle Enterprise Session Border Controller CLI Configuration Guide*.

Supported Platforms

The following platforms support Admin Security:

- Acme Packet 1100
- Acme Packet 3900
- Acme Packet 3950
- Acme Packet 4600
- Acme Packet 4900
- Acme Packet 6300
- Acme Packet 6350
- VMWare
- KVM

JITC Support

The SBC supports Joint Interoperability Testing Command (JITC). To enable JITC, use the **setup entitlements** command to enable both the Admin Security entitlement and the FIPS entitlement.



Note:

The JITC feature set is supported only on Enterprise releases.

For more information on installing the Admin Security feature set, see the *Oracle Enterprise Session Border Controller Release Notes*. For instructions on provisioning this feature set, see the *Oracle Enterprise Session Border Controller Configuration Guide*.



Note:

JITC features supersede Admin Security features.

Supported Platforms

The following platforms support JITC mode:

- Acme Packet 1100
- Acme Packet 3900
- Acme Packet 4600
- Acme Packet 6300
- VME

Login Banner

Upon successful user authentication and authorization, the Oracle SBC displays the login banner.

- Last login: displays the date and time that the current user last successfully logged-in
- System last accessed: displays the date and time and user name of the last user who successfully logged-in
- Unsuccessful login attempts: displays the date and time of the last five unsuccessful login attempts by the current user
- Confirm reading: requires user acknowledgement of the display banner. A positive response (y) successfully completes login, and starts audit-log activity for this user session. A negative response (n) generates an audit-log entry and logs the user out of the SBC.

The login banner also provides notification of impending password or SSH public key expiration as described in Password Policy Configuration.

Local User Accounts

The SBC comes with two local, factory accounts for access. System administrators may create additional local accounts for each user or administrator who needs to access the SBC. Local accounts ensure your ability to audit an individual's activity on the SBC.

When creating local accounts, you must specify the username and the user class. Usernames must be unique, and neither `user` nor `admin` may be used.

There are two user classes: `user` and `admin`. Local accounts in the `user` class have the same access level as the factory user account, and local accounts in the `admin` class have the same access level as the factory admin account.

After a second administrator account has been created, you may disable the factory user and admin accounts. The SBC requires at least one administrator account. Only administrators may delete accounts, and administrators may not delete their own account. Use the command `factory-accounts` to disable or re-enable the factory accounts.

The file `cli.audit.log` records the timestamp, the local account name, the connecting IP address, and the command run by any user or administrator.

```
2020-10-01 15:35:06.530 TaskID: 0xab7c8710, admin@10.2.2.7 : 'show
users'
2020-10-01 15:36:14.112 TaskID: 0xab7c8710, alice@10.2.2.8 : 'show
users'
```

Local Accounts and TACACS+

When the `tacacs-authentication-only` attribute is enabled in the `security` configuration element or when the Admin Security entitlement is enabled, authentication to a local account changes when TACACS+ is configured. If a TACACS+ server is configured and available, then authentication uses TACACS+ and the SBC rejects attempts to authenticate to local accounts. If a TACACS+ server is configured but unavailable, the SBC allows authentication to local accounts. This ensures that, when TACACS+ is configured, authentication to local accounts is only possible when the TACACS+ server is down. If no TACACS+ server is configured, local accounts are accessible.

Local Accounts and SSH Keys

SSH authorized keys take precedence over local accounts. For example, if an administrator imported Alice's SSH key into the admin class, then Alice can authenticate with `ssh alice@10.0.0.1` whether or not a local account exists. Moreover, if a local account named `alice` exists in the user class but an SSH authorized-key exists in the admin class, Alice can still authenticate as an administrator because SSH keys take precedence over local accounts. Conversely, if Alice's SSH key were imported into the user class but a local account in the admin class were created for Alice, she would by default log in as an ordinary user and not as an administrator. This happens because SSH clients usually try public key authentication before attempting password-based authentication. To authenticate using password-based authentication when public key authentication is an option, use the `-o` option: `ssh -o PubkeyAuthentication=no alice@10.0.0.1`.

When deleting an account, it is important to remember to delete any unused SSH keys for that user or administrator.

Password Policy

The Admin Security feature set supports the creation of password policies that enhance the authentication process by imposing requirements for:

- password length
- password strength
- password history and re-use
- password expiration and grace period

The Admin Security feature set mandates the following password length/strength requirements.

- user class passwords must contain at least 9 characters (Admin Security only)
- admin class passwords must contain at least 15 characters
- passwords must contain at least 2 lower case alphabetic characters
- passwords must contain at least 2 upper case alphabetic characters
- passwords must contain at least 2 numeric characters
- passwords must contain at least 2 special characters (such as !, ", #, \$, %, &, ', (,), *, +, ,, -, ., /, :, ;, <, =, >, ?, @, [, \,], ^, _ ` , {, }, ~)
- passwords must differ from the prior password by at least 4 characters
- characters in password must differ from the prior password in at least 8 positions
- passwords cannot contain, repeat, or reverse the entire username
- passwords cannot contain three consecutive identical characters

Some specific password policy properties, specifically those regarding password lifetime and expiration procedures, are also applicable to SSH public keys used to authenticate clients.

Configuring Password Policy Properties

The single instance **password-policy** configuration element defines the password policy.

1. From superuser mode, use the following command path to access password-policy configuration mode.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# password-policy
ORACLE(password-policy)#
```

The **password-policy** configuration element properties (with the introduction of the Admin Security or JITC feature) are shown below with their default values.

```
min-secure-pwd-length      8
expiry-interval            90
expiry-notify-period      30
grace-period               30
grace-logins               3
password-history-count     3
password-change-interval  24
```

2. Use the **expiry-interval** command to specify the password lifetime in days. Password lifetime tracking begins when a password is changed.

Allowable values are integers within the range 0 through 65535, with a default value of 90 (days).

 **Note:**

The minimum **expiry-interval** is 0 with a provisioned JITC feature only and remains 1 when only an Admin Security feature is provisioned.

```
ORACLE(password-policy)# expiry-interval 60
ORACLE(password-policy)#
```

3. Use the **password-change-interval** command to specify the minimum password lifetime (the minimum time that must elapse between password changes.)

Allowable values are integers within the range 1 through 24, with a default value of 24 (hours).

```
ORACLE(password-policy)# password-change-interval 18
ORACLE(password-policy)#
```

4. Use the **expiry-notify-period** to specify the number of days prior to expiration that users begin to receive password expiration notifications.

Allowable values are integers within the range 1 through 90, with a default value of 30 (days).

During the notification period, users are reminded of impending password expiration at both Session Director login and logout.

```
ORACLE(password-policy)# expiry-notify-period 10
ORACLE(password-policy)#
```

5. Use the **grace-period** command in conjunction with the **grace-logins** command, to police user access after password expiration.

After password expiration, users are granted some number of logins (specified by the **grace-logins** command) for some number of days (specified by the **grace-period** command). Once the number of logins has been exceeded, or once the grace period has expired, the user is forced to change his or her password.

Allowable values for **grace-period** are integers within the range 1 through 90, with a default value of 30 (days).

Allowable values for **grace-logins** are integers within the range 1 through 10, with a default value of 3 (logins).

```
ORACLE(password-policy)# grace-period 1
ORACLE(password-policy)# grace-logins 1
ORACLE(password-policy)#
```

6. Use the **password-history-count** command to specify the number of previously used passwords retained in encrypted format in the password history cache.

Allowable values are integers within the range 1 through 24, with a default value of 3 (retained passwords).

 **Note:**

The maximum **password-history-count** is 24 with a provisioned JITC feature only and remains 10 when only an Admin Security feature is provisioned.

By default, a user's three most recently expired passwords are retained in the password history. As the user's current password is changed, that password is added to the history, replacing the oldest password entry.

New, proposed passwords are evaluated against the contents of the password cache, to prevent password re-use, and guard against minimal password changes.

```
ORACLE(password-policy)# password-history-count 10
ORACLE(password-policy)#
```

7. Use **done**, **exit** and **verify-config** to complete password policy.

Configuring the Administrative Security with ACP Password Rules

To enforce the stronger password rules and restrictions that the Administrative Security ACP license it provides, you must enable the `password-policy-strength` parameter.

- Confirm that the Administrative Security ACP license is installed on the system.
- You must have Superuser permissions.

From the command line, go to the **password-policy** configuration element and set the **password-policy-strength** parameter to **enabled**.

 **Note:**

The **password-policy** configuration element displays the **min-secure-pwd-len** command. You do not need to configure the **min-secure-pwd-len** command because the Administrative Security ACP license overrides this command with a stronger rule.

You can configure any of the other password policy settings without a system override, according to the ranges specified in this procedure. For more information about the ranges, see "Administrative Security ACP License Configuration."

1. Access the **password-policy** configuration element.

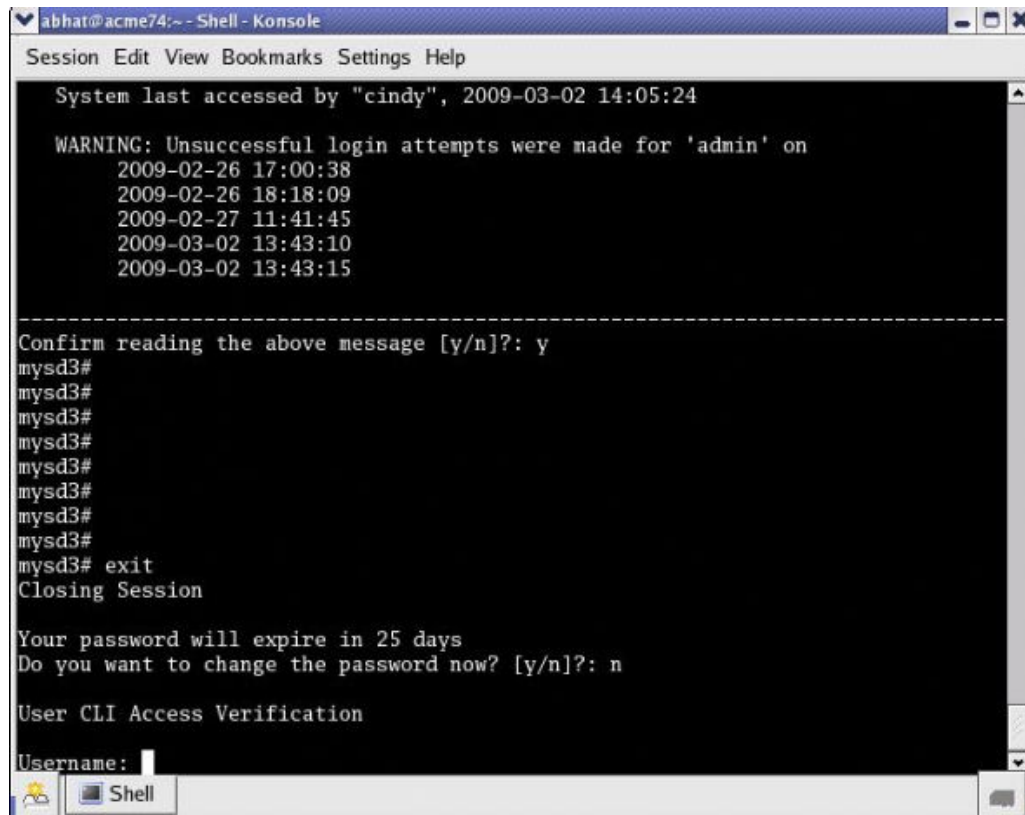
```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# password-policy
ORACLE(password-policy)#
```

2. Type **select**, and press ENTER.
3. Type **show**, and press ENTER.
4. Configure the following password policy settings, as needed:
 - **expiry-interval**. 1-65535 days.

- **expiry-notify-period.** 1-90 days.
 - **grace-period.** 1-90 days.
 - **grace-logins.** 1-10 attempts.
 - **password-history-count.** 1-10 passwords.
 - **password-change-interval.** 1-24 hours.
 - **password-policy-strength.** Type **enabled**, and press ENTER.
5. Do the following:
 - a. Type **done**, and press ENTER.
 - b. Type **exit**, and press ENTER.
 - c. Type **done**, and press ENTER.

Changing a Password

As shown in the following figures, the **password-policy** configuration element provides prior notice of impending password expiration via the login banner display, and with additional notices when ending a login session.



```
abhat@acme74:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
System last accessed by "cindy", 2009-03-02 14:05:24

WARNING: Unsuccessful login attempts were made for 'admin' on
2009-02-26 17:00:38
2009-02-26 18:18:09
2009-02-27 11:41:45
2009-03-02 13:43:10
2009-03-02 13:43:15

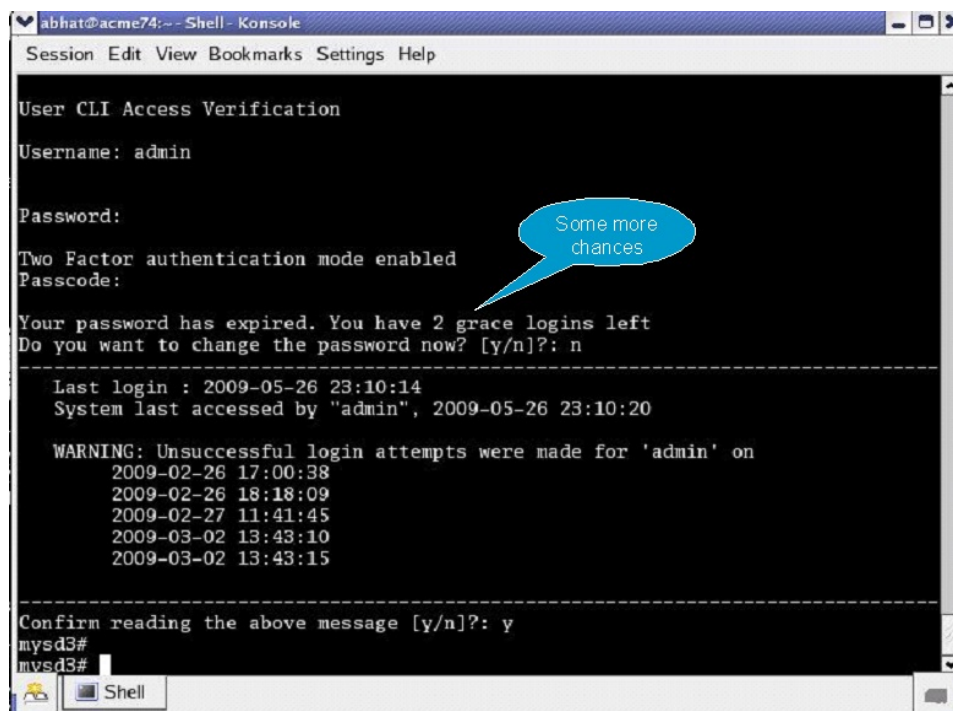
-----
Confirm reading the above message [y/n]?: y
mysd3#
mysd3#
mysd3#
mysd3#
mysd3#
mysd3#
mysd3#
mysd3#
mysd3#
mysd3# exit
Closing Session

Your password will expire in 25 days
Do you want to change the password now? [y/n]?: n

User CLI Access Verification
Username: |
```

Password Expiration Notices at Login and Logout

After password expiration, additional notices are displayed with each grace login. If all notices are ignored, the password-policy enforces a password change when grace logins have been exhausted, or when the grace period has elapsed.



```
abhat@acme74:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

User CLI Access Verification
Username: admin
Password:
Two Factor authentication mode enabled
Passcode:
Your password has expired. You have 2 grace logins left
Do you want to change the password now? [y/n]?: n

-----
Last login : 2009-05-26 23:10:14
System last accessed by "admin", 2009-05-26 23:10:20

WARNING: Unsuccessful login attempts were made for 'admin' on
2009-02-26 17:00:38
2009-02-26 18:18:09
2009-02-27 11:41:45
2009-03-02 13:43:10
2009-03-02 13:43:15

-----
Confirm reading the above message [y/n]?: y
mysd3#
mysd3#
```

A blue speech bubble points to the text "You have 2 grace logins left" with the text "Some more chances".

Changing Password Process

To change your password in response to (1) an impending expiration notice displayed within the login banner or at system logout, (2) a grace login notice, or (3) an expiration notice:

1. If responding to an impending expiration notice, or a grace login notice, type `y` at the Do you want to change the password ... prompt.
2. Provide a new, valid password in response to the Enter New Password: prompt.
3. Re-enter the password in response to the Confirm New Password: prompt.
4. If performing a login, enter `y` to acknowledge reading the login banner to complete login with the new password.

A user class account can change its password only in response to one of the three notifications described above.

Similarly, an admin class account can change the password in response to the same notifications. Additionally, these accounts can change passwords using the ACLI as described in the following sections.

Changing the user Password

Change the password of the default factory user account from an admin class account.

1. Enter **secret login** at the prompt and provide the current password when challenged.

```
ORACLE# secret login
Enter current password :
```

2. Type the new password.

```
ORACLE# secret login
Enter current password :
Enter new password :
```

3. Confirm the password.

```
ORACLE# secret login
Enter current password :
Enter new password :
Enter password again :
ORACLE#
```

Changing the admin Password

Change the password of the default factory admin account from an admin class account.

1. Enter **secret enable** at the prompt and provide the current password when challenged.

```
ORACLE# secret enable
Enter current password :
```

2. Type the new password.

```
ORACLE# secret enable
Enter current password :
Enter new password :
```

3. Confirm the password.

```
ORACLE# secret enable
Enter current password :
Enter new password :
Enter password again :
ORACLE#
```

Changing a Local Account Password

To change the password of a local account, you must be in admin mode.

1. Log in to an admin-class local account or the factory admin account.
2. Use the **local-accounts** command to change the password.

The syntax:

```
local-accounts change-password <username>
```

Enter the user's existing password first and then the new password. For example:

```
ORACLE# local-accounts change-password bob
Enter Existing Password:
Enter New Password:

Password is acceptable.

Enter Password Again:
Request to 'SAVE-CONFIG' has Finished,
Save complete
Currently active and saved configurations do not match!
To sync & activate, run 'activate-config' or 'reboot activate'.

Please wait...

Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete

-- Password updated. -----
ORACLE#
```

After the password is accepted, the system automatically runs activate-config to update the password.

3. If you do not know the current password for that user, use the **local-accounts reset** command to create a temporary, one-time password for that user.

```
ORACLE# local-accounts reset bob
This command will reset the current password for account {bob}
Then prompt for a new single-use temporary password
Are you sure you want to proceed [y/n]?: y
Enter Temporary Password:

Password is acceptable.

Enter Password Again:
Request to 'SAVE-CONFIG' has Finished,
Save complete
Currently active and saved configurations do not match!
To sync & activate, run 'activate-config' or 'reboot activate'.

Please wait...

Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete

-- Password updated. -----
ORACLE#
```

A user whose password has been reset must create a new password after logging in.

```
$ ssh bob@10.0.0.1
Password:

Your password has expired
You must change your password to continue
Enter New Password:

Password is acceptable.

Enter Password Again:
Request to 'SAVE-CONFIG' has Finished,
Save complete
Currently active and saved configurations do not match!
To sync & activate, run 'activate-config' or 'reboot activate'.

Please wait...

Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete

-- Password updated. -----
ORACLE>
```

For more information about managing local accounts, see "[Manage Local Accounts](#)" in the *Configuration Guide*.

RADIUS and TACACS+ Passwords

With RADIUS or TACACS+ enabled, passwords are stored and controlled on the remote server or servers. Consequently, none of the length/strength, re-use, history, or expiration requirements mandated by the password policy are applicable to these passwords.

Login Policy

The Login Policy controls concurrent system access to a specified number of users, sets the maximum number of unsuccessful login attempts, specifies the response to login failure, and specifies the login mode (single-factor or two-factor).



Note:

If user authentication fails or a user is locked out of the system, the SBC will not display the reason why the login failed.

The single instance **login-config** configuration element defines login policy.

1. From an admin class account, access the **login-config** configuration element:

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# admin-security
ORACLE(admin-security)# login-config
ORACLE(login-config)#
```

login-config configuration element properties are shown below with their default values

concurrent-session-limit	2
max-login-attempts	3
login-attempt-interval	4
lockout-interval	60
send-alarm	enabled
login-auth-mode	single-factor
enable-login-banner	enabled

2. **concurrent-session-limit**—specifies the maximum number of simultaneous connections allowed per user name

Allowable values are integers within the range 1 through 10, with a default of 2 (simultaneous connections).

Retain the default value, or specify a new connection limit.

```
ORACLE(login-config)# concurrent-session limit 4
ORACLE(login-config)#
```

3. **max-login-attempts**—specifies the number of consecutive unsuccessful login attempts that trigger disconnection of a console, SSH, or SFTP session.

Allowable values are integers within the range 2 through 100, with a default of 3 (sessions).

Retain the default value, or specify a new threshold value.

```
ORACLE(login-config)# max-login-attempts 5
ORACLE(login-config)#
```

4. **login-attempt-interval**—specifies an idle interval in seconds imposed after an unsuccessful login attempt.

Allowable values are integers within the range 4 through 60, with a default value of 4 seconds.

Retain the default value, or specify a new login interval.

```
ORACLE(login-config)# login-attempt-interval 6
ORACLE(login-config)#
```

5. **lockout-interval**—specifies the number of seconds that logins from an interface are not allowed after the **max-login-attempts** threshold has been reached

Allowable values are integers within the range of 15 through 300. The default value is 60 seconds.

 **Note:**

The minimum **lockout-interval** is 15 when the JITC feature is enabled, but remains 30 when only the Admin Security feature is provisioned.

Retain the default value, or specify a new lockout interval.

```
ORACLE(login-config)# lockout-interval 30
ORACLE(login-config)#
```

- 6. send-alarm**—enables the generation and transmission of alarms in the event of an interface lockout

Allowable values are **enabled** (the default) or **disabled**.

Retain the default value, or select **disabled** to squelch alarm generation.

```
ORACLE(login-config)# send-alarm disabled
ORACLE(login-config)#
```

- 7. login-auth-mode**—specifies the local login authentication mode

Allowable values are **single-factor** (the default) or **two-factor**.

single-factor authentication requires the service requester to present a single authentication credential, a password.

two-factor authentication requires the service requester to present two authentication credentials, a password and a passcode.

Retain the default value, or specify two-factor authentication.

```
ORACLE(login-config)# login-auth-mode two-factor
ORACLE(login-config)#
```

- 8. enable-login-banner**—enables or disables display of the login banner

Allowable values are **enable** (the default) or **disable**.

Retain the default value, or disable login banner display.

```
ORACLE(login-config)# enable-login-banner disable
ORACLE(login-config)#
```

A sample login policy configuration appears below:

```
ORACLE(login-config)# concurrent-session limit 4
ORACLE(login-config)# max-login-attempts 5
ORACLE(login-config)# login-attempt-interval 6
ORACLE(login-config)# lockout-interval 30
ORACLE(login-config)# done
ORACLE(login-config)# exit
ORACLE(admin-security)#
```

Defines a login-config configuration element that allows four simultaneous connections per username. An idle interval of 6 seconds is imposed after an unsuccessful login attempt. Five consecutive unsuccessful login attempts trigger a 30-second lockout of the interface over which the unsuccessful logins were received. By default, single-factor authentication, alarm generation, and login banner display are enable.

Authentication and Authorization

Authentication is the process of confirming the alleged identity of a service requester. Authorization, a process performed after authentication, determines the access or privilege level granted to an authenticated requester.

Local Authentication and Authorization

This section describes user authentication and authorization performed locally by a Oracle SBC that has either the Admin Security or JITC feature set provisioned.

The SBC provides two default factory accounts:

- user
- admin

Each of the two factory accounts is associated with an eponymous authorization class which defines the access level for all local accounts within that class.

user (authorization class)

- provides read-only access to non-security configurations
- provides read access to visible files
- login to user mode
- cannot switch to admin mode

admin (authorization class)

- provides read-write access to all configuration
- provides read/write access to a sub-set of file system elements
- login to admin mode
- cannot switch to user mode

When local accounts are created, you specify the authorization class using the syntax:

```
local-accounts add <username> [ user | admin ]
```

Console Login

The following table summarizes the authorization classes for local accounts.

User Name	Login prompt	Authorization class
user	user mode >	user

User Name	Login prompt	Authorization class
admin	admin mode #	admin

Serial Port Control

With the addition of the Admin Security feature, you may enable or disable access to the serial (console) port. In the absence of this feature, access to the serial is generally available. The ACLI command **console-io** functions as a switch that you set to **enabled** to allow serial port access and to **disabled** to keep the serial port from being used.

If you remove the administrative management feature after disabling the serial port, the SBC reverts to its default behavior by providing serial port access.

To turn off access to the serial port:

- At the system prompt, type **console-io** followed by a Space. Then type **disabled** and press Enter.

```
ORACLE# console-io disabled
```

If you want to re-enable the serial port, use the same command with the **enabled** argument.

Initial Login

Upon initial login, user and admin are required to change the respective password. Initial login is completed only after password change and acknowledgment of the login banner.

With each release, the default ciphers of the SBC may be upgraded to their latest secure versions. If a verbose connection log of an SSH or SFTP client shows that it cannot agree on a cipher with the OCSBC, upgrade your client.

The following figure shows the initial login screen for the default admin account.

To complete initial login:

1. SSH to the SBC using one of the two factory default accounts (user or admin).

```
ssh admin@<IP address>
```

2. Enter the initial password.

The initial password for the default user account is `acme`; the initial password for the default admin account is `packet`.

3. Enter a new password in response to the Enter New Password: prompt.

Passwords must meet the following length/strength requirements.

- user password must contain at least 9 characters

- admin password must contain at least 15 characters
 - passwords must contain at least 2 lower case alphabetic characters
 - passwords must contain at least 2 upper case alphabetic characters
 - passwords must contain at least 2 numeric characters
 - passwords must contain at least 2 special characters
 - passwords must differ from the prior password by at least 4 characters
 - characters in password must differ from the prior password in at least 8 positions
 - passwords cannot contain, repeat, or reverse the user name
 - passwords cannot contain three consecutive identical characters
4. Re-enter the new password in response to the Confirm New Password: prompt.
 5. Enter **y** to acknowledge reading the login banner to complete initial login.

Remote SSH Login with Password

The following shows logging in as the factory default admin account.

```
[bob@linuxbox ~]$ ssh admin@10.1.1.2
Password:
-----
---
Last login : 2020-11-09 14:10:03
System last accessed by "admin", 2020-11-09 14:10:03

WARNING: Unsuccessful login attempts were made for 'admin' on
2020-09-14 15:14:08
2020-11-03 16:53:57
2020-11-06 14:15:53
2020-11-06 14:16:09
2020-11-06 14:16:21
-----
---
Confirm reading the above message [y/n]?: y
ADMINSEC#
```

Remote SSH Login with Public Key

As an alternative to password-based authentication, you can authenticate using SSH public keys. To set up public-key authentication, import a copy of the public key of each user who will authenticate using this method. The public key identifies the user as a trusted entity when the Oracle SBC performs authentication.

During the SSH login, the user presents its public key to the SBC, which validates the offered public key against the previously obtained trusted copy of the key to identify and authenticate the user.

1. On the SSH client, export your public key to RFC 4716 format.

```
[user@host ~]$ ssh-keygen -ef .ssh/id_rsa.pub
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "1024-bit RSA, converted from OpenSSH by user@host"
AAAAB3NzaC1yc2EAAAABIwAAV595VmIZB1pAIEAxcYTeOZx9sX/
mSqdHy12P+AFihDJYdL
qJIWdiZuSmIC6na62iDny8HZIxT25mlyBCU7UsLwEdyLhlYOuknkmxDyHTbrQ4dLHz3b
1q
3Tb8auz97/J39PT42Colp4pwRODzPBrXJV0SSJ8BjC9+OglNE/83ClYLEwE=
---- END SSH2 PUBLIC KEY ----
[user@host ~]$
```

2. Use the **ssh-key** command to import the public key to the SBC.

Syntax:

```
ssh-key authorized-key import <name> <authorizationClass>
```

- where name is an alias or handle assigned to the imported public key, often the user's name.
- where authorizationClass is either user (the default) or admin.

To import a public key for Dwight in the user class:

```
ORACLE# ssh-key authorized-key import Dwight
ORACLE#
```

To import a public key for Matilda in the admin class:

```
ORACLE# ssh-key authorized-key import Matilda admin
ORACLE#
```

3. Paste the public key with the bracketing Begin and End markers at the cursor point.
4. Enter a semi-colon (;) to signal the end of the imported host key.

The entire import sequence is shown below.

```
ORACLE# ssh-key authorized-key import Matilda admin
```

IMPORTANT:

Please paste ssh public key in the format defined in RFC 4716.
Terminate the key with ";" to exit.....

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "1024-bit RSA, converted from OpenSSH by user@host"
AAAAB3NzaC1yc2EAAAABIwAAV595VmIZB1pAIEAxcYTeOZx9sX/
mSqdHy12P+AFihDJYdL
qJIWdiZuSmIC6na62iDny8HZIxT25mlyBCU7UsLwEdyLhlYOuknkmxDyHTbrQ4dLHz3b
1q
3Tb8auz97/J39PT42Colp4pwRODzPBrXJV0SSJ8BjC9+OglNE/83ClYLEwE=
---- END SSH2 PUBLIC KEY ----;
SSH public key imported successfully....
```

```
WARNING: Configuration changed, run "save-config" command to save it
and run "activate-config" to activate the changes
ORACLE#
```

5. Save and activate the configuration.
6. If necessary, repeat the above procedure to import additional user-specific public keys.

 **Note:**

Imported SSH public keys are subject to the same expiration policies and procedures as passwords. An SSH public key's lifetime is the same as a password, and it is subject to the same notifications and grace intervals. If an SSH public key expires, the admin user must import a new SSH public key for the user. To ensure continuity of access, the admin should import a new SSH public key prior to the key expiration.

The following figure shows the successful public-key authentication of Matilda, who has logged in with admin privileges.

```
[user@host ~]$ ssh Matilda@10.1.1.2
-----
-----
Last login : 2020-11-09 14:13:59
System last accessed by "Matilda", 2020-11-09 14:13:59

WARNING: Unsuccessful login attempts were made for 'admin' on
    2020-09-14 15:14:08
    2020-11-03 16:53:57
    2020-11-06 14:15:53
    2020-11-06 14:16:09
    2020-11-06 14:16:21
-----
-----
Confirm reading the above message [y/n]?: y
ORACLE#
```

Note that the login banner refers to the admin and user login by the aliases used when the trusted copies of their SSH public keys were imported. In all respects, however, Matilda is a admin instance.

Two-Factor Authentication

Two-factor authentication (2FA) adds an extra layer of security when authenticating to the SBC by requiring a key, such as an SSH public key or X.509 certificate, as well as a username and password. 2FA can be enabled on either the web interface, the SSH interface, or both.

Pre-Requisites

The Admin Security entitlement must be installed.

Two-factor Authentication for the Web GUI

To enable 2FA on the web interface, add the Common Name of the X.509 client certificate to the **common-name-list** attribute and set the value of **two-factor-auth-access-method-list** to **GUI**. When enabled, 2FA to the GUI requires the browser to send an X.509 client certificate to the SBC. The SBC then verifies:

- The client certificate key length is at least 1024 bytes (2048 bytes if the FIPS entitlement is also installed).
- The client certificate was signed by a previously installed root CA certificate.
- The client certificate is not revoked.
- The Common Name in the certificate is found in the list configured in the **common-name-list** attribute.

 **Note:**

2FA will fail if the Common Name associated with the client certificate that the web browser sends to the SBC is not found in the **common-name-list** attribute.

If all of these conditions are true, the SBC proceeds to authenticate the user based on the existing methods like username, RADIUS, or TACACS+.

To use 2FA on the web interface, the client browser must be configured to send the appropriate certificate to the SBC. Check your browser's documentation for how to add a client certificate to your browser's certificate manager.

Two-factor Authentication for the SSH Interface

2FA on the SSH interface can use either local authentication (to login as the local admin account or the local user account) or RADIUS/TACACS+ authentication. In either case, at least one of the following is required:

- Import the authorized-key of an SSH client; or
- Import the ca-key that will sign the public keys of SSH clients.

For example, if the **type** attribute in the **authentication** element is set to **local**, you can import the public key of the SSH client that will log in as the local admin account.

```
ssh-key authorized-key import admin admin
```

Alternatively, if the **type** attribute in the **authentication** element is set to **tacacs** or **radius**, you can import the public key of a RADIUS or TACACS+ user who will administer the SBC.

```
ssh-key authorized-key import alice admin
```

**Note:**

For instructions on signing certificates, see "Add a Certificate Authority Key" in the *ACLI Configuration Guide*.

After certificates are configured, set the **two-factor-auth-access-method-list** to **SSH**. When enabled, the SBC requires a user to authenticate with both a public key and a password.

2FA Lockout

To prevent getting locked out of an SBC with 2FA enabled, never delete the last authorized-key or the last ca-key.

**Caution:**

Removing the last authorized-key or the last ca-key will make it impossible to SSH to the management port if 2FA is enabled on the SSH interface.

Enable Two-Factor Authentication

1. Access the **two-factor-authentication** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# authentication
ORACLE(authentication)# two-factor-authentication
ORACLE(two-factor-authentication)#
```

2. **two-factor-auth-access-method-list**—Select the interface that will require 2FA. You may enable 2FA on either the SSH interface, the web interface, or both.
3. **common-name-list**—Enter the Common Name (CN) found in the certificate that the client sends.
4. Type **done** to save the configuration.

RADIUS Authentication and Authorization

As an alternative to local authentication, users may use a RADIUS server to authenticate.

For information on configuring between RADIUS servers and the SBC refer to RADIUS Authentication section in the *ACLI Configuration Guide*.

A RADIUS users file (shown below), stored on the RADIUS server, provides the basis for server authentication and authorization decisions.

```
user1  Cleartext-Password := "user1"
       Acme-User-Class = user,
       Acme-User-Privilege = sftpForAll

user2  Cleartext-Password := "user2"
       Acme-User-Class = user,
```

```
Acme-User-Privilege = sftpForAll

admin1 Cleartext-Password := "admin1"
      Acme-User-Class = admin,
      Acme-User-Privilege = sftpForAll

admin2 Cleartext-Password := "admin2"
      Acme-User-Class = admin,
      Acme-User-Privilege = sftpForAll
```

Upon receiving a login request, the SBC sends a RADIUS Access Request message to the RADIUS server. The request message contains, among other things, the username:password requesting access to SBC resources. Upon receiving the request, the RADIUS server checks its user file for the username:password pair. If it finds a congruent match, the requestor is authenticated.

Successful authentication generates a Access Accept message to the SBC; the message also contains the contents of two Oracle Vendor Specific Attributes (VSAs). Acme-User-Class specifies the configuration privileges accorded the authenticated user. Acme-User-Privilege specifies the log file access accorded to the authenticated user. Together these two VSAs provide the authorization function. Consequently, the RADIUS server functions as an authentication and authorization decision point, while the SBC functions as an enforcement point.

RADIUS Authorization Classes

The RADIUS authorization classes, as specified by the Acme-User-Class VSA, coincides with the two default admin and user authorization classes.



Note:

The value of Acme-User-Class must be lowercase.

user

- provides read-only for all system configuration
- The login prompt for this user is ORACLE>

```
Acme-User-Class = user
```

admin

- provides read-write access for all system configuration
- The login prompt for this user is ORACLE#

```
Acme-User-Class = admin
```

RADIUS and SSH

When the SBC uses a RADIUS server for authentication, you cannot log in over SSH using the two factory defined accounts (user and admin). Attempts to login via SSH are rejected.

```
[user@host ~]$ ssh user@10.1.1.2
Password:
% Error: Cannot login as local user when AAA Server is enabled
```

Use console to login as local user

RADIUS and Password Policies

With RADIUS enabled, passwords are stored and controlled on the remote RADIUS server or servers. Consequently, none of the length/strength, re-use, history, or expiration requirements mandated by the local password policy are applicable to RADIUS passwords. Most RADIUS servers, however, do enforce password policies of their own.

TACACS+ Support

As an alternative to the local authentication/authorization described in previous sections, the SBC supports TACACS+ in both Admin Security mode and JITC. The SBC allows HTTPS, SSH, and SFTP logins with TACACS+ credentials, honoring the privilege level returned by the TACACS+ server and, if **tacacs-authorization** is enabled, validates commands via TACACS+ when the user has privileges.



Note:

For SFTP, only TACACS+ users with admin privileges have permission to login.

When TACACS+ is configured and a Public Key Infrastructure (PKI) user logs into the SBC, the SBC performs the authentication locally against the locally stored public RSA key, but performs authorization and accounting with TACACS+. This means that, instead of adhering to the permissions configured when importing the public key, the SBC queries the TACACS+ server and generates start/stop accounting records using TACACS+ settings.

TACACS+ over IPsec

To run TACACS+ over IPsec on an SBC with the Admin Security entitlement set, you must complete the following steps:

1. Configure a phy-interface for wancom0.
For example:

```
ADMINSEC# conf term
ADMINSEC(configure)# system phy-interface
ADMINSEC(phy-interface)# name wancom0
ADMINSEC(phy-interface)# operation-type Control
ADMINSEC(phy-interface)# done
```

2. Configure a network-interface for wancom0.

For example:

```
ADMINSEC# conf term
ADMINSEC(configure)# system network-interface
ADMINSEC(network-interface)# name wancom0
ADMINSEC(network-interface)# ip-address 10.1.1.5
ADMINSEC(network-interface)# netmask 255.255.255.0
ADMINSEC(network-interface)# gateway 10.1.1.1
ADMINSEC(network-interface)# done
```

3. Configure a security-policy for wancom0.

For example:

```
ADMINSEC# conf term
ADMINSEC(configure)# security ipsec security-policy
ADMINSEC(security-policy)# name secureTacacs
ADMINSEC(security-policy)# network-interface wancom0:0
ADMINSEC(security-policy)# remote-ip-addr-match 10.1.1.42
ADMINSEC(security-policy)# remote-ip-mask 255.255.255.0
ADMINSEC(security-policy)# done
```

4. Configure the `ikev2-ipsec-wancom0-params` configuration element.

For example:

```
ADMINSEC# conf term
ADMINSEC(configure)# security ikev2-ipsec-wancom0-params
ADMINSEC(ikev2-ipsec-wancom0-params)# name TacPlusIPsec
ADMINSEC(ikev2-ipsec-wancom0-params)# remoteip 10.1.1.42
ADMINSEC(ikev2-ipsec-wancom0-params)# remotesubnet 10.1.1.42/32
ADMINSEC(ikev2-ipsec-wancom0-params)# localip 10.1.1.5
ADMINSEC(ikev2-ipsec-wancom0-params)# localsubnet 10.1.1.5/32
ADMINSEC(ikev2-ipsec-wancom0-params)# authby secret
ADMINSEC(ikev2-ipsec-wancom0-params)# shared-password
Enter password:
Retype password:
Password updated
ADMINSEC(ikev2-ipsec-wancom0-params)# done
```

5. Save and activate your configuration.

If no security-policy is configured, the SBC displays this verify-config error.

```
WARNING: Admin-security enabled. 10.1.1.42 tacacs server does not
match any of the configured security-policy's remote-ip-addr-match/
remote-ip-mask subnet. So communication from SBC to this server over
media ports will not be secured by IPsec.
```

SSH and SFTP

With the Admin Security or JITC feature sets enabled, the Secure Shell (SSH) and related Secure Shell File Transfer (SFTP) protocols provide for the secure transfer of audit files and for the secure transfer of management traffic across the wancom0 interface.

SSH Operations

SSH Version 2.0, the only version supported on the SBC, is defined by a series of five RFCs.

- RFC 4250, *The Secure Shell (SSH) Protocol Assigned Numbers*
- RFC 4251, *The Secure Shell (SSH) Protocol Architecture*
- RFC 4252, *The Secure Shell (SSH) Authentication Protocol*
- RFC 4253, *The Secure Shell (SSH) Transport Layer Protocol*
- RFC 4254, *The Secure Shell (SSH) Connection Protocol*

RFCs 4252 and 4253 are most relevant to SBC operations.

The transport layer protocol (RFC 4253) provides algorithm negotiation and key exchange. The key exchange includes server authentication and results in a cryptographically secured connection that provides integrity, confidentiality and optional compression. Forward security is provided through a Diffie-Hellman key agreement. This key agreement results in a shared session key. The rest of the session is encrypted using a symmetric cipher, currently 128-bit AES, Blowfish, 3DES, CAST128, Arcfour, 192-bit AES, or 256-bit AES. The client selects the encryption algorithm to use from those offered by the server. Additionally, session integrity is provided through a cryptographic message authentication code (hmac-md5, hmac-sha1, umac-64 or hmac-ripemd160).

The authentication protocol (RFC 4252) uses this secure connection provided and supported by the transport layer. It provides several mechanisms for user authentication. Two modes are supported by the SBC: traditional password authentication and public-key authentication.

Configuring SSH Properties

The single instance **ssh-config** configuration element specifies SSH re-keying thresholds.

1. From admin mode, use the following command path to access the ssh configuration element:

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# ssh-config
ORACLE(ssh-config)#
```

ssh configuration element properties are shown below with their default values

```
rekey-interval      60
rekey-byte-count    31
```

2. **rekey-interval**—specifies the maximum allowed interval, in minutes, between SSH key negotiations

Allowable values are integers within the range 60 through 600, with a default of 60 (minutes). Shorter lifetimes provide more secure connections.

Works in conjunction with **rekey-byte-count**, which sets a packet-based threshold, to trigger an SSH renegotiation. If either trigger is activated, an SSH renegotiation is begun.

Retain the default value, or specify a new value.

```
ORACLE (ssh-config) # rekey-interval 20
ORACLE (ssh-config)
```

- rekey-byte-count**—specifies the maximum allowed send and receive packet count, in powers of 2, between SSH key negotiations

Allowable values are integers within the range 20 (1,048,576 packets) through 31 (2,147,483,648 packets), with a default of 31 (2^{31}). Smaller packet counts provide more secure connections.

Works in conjunction with **rekey-interval**, which sets a time-based threshold, to trigger an SSH renegotiation. If either trigger is activated, an SSH renegotiation is begun.

Retain the default value, or specify a new value.

```
ORACLE (ssh-config) # rekey-packet-count 24
ORACLE (ssh-config)
```

A sample SSH configuration appears below:

```
ORACLE (ssh-config) # rekey-interval 20
ORACLE (ssh-config) # done
ORACLE (ssh-config) # exit
ORACLE (security) #
```

Specifies a key renegotiation every 20 minutes, or at the reception/transmission of 2,147,483,648 packets, whichever comes first.

Manage SSH Keys

Use the **ssh-key** command to manage SSH keys for the SBC.

Add an SSH Authorized Key

To authenticate to the SBC using public key authentication rather than a password, use the **ssh-key** command with the **authorized-key import** argument.

- On the SSH client, convert the public key of the SSH client into RFC 4716 format.

Note:

Valid RSA key sizes are 2048, 3072, or 4096 bytes. The only valid DSA key size is 1024 bytes.

To do this on Oracle Linux, use the **ssh-keygen** command.

```
[bob@client ~]$ ssh-keygen -e -f .ssh/id_rsa.pub
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by bob@client from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAQADOTDujYoQXzjTt9I8YvJMvfSVlWZ6iDzfrx06R
```

```

31
Rj/lrjxlWDMc/Y/uEd2sJ+5wdlCnJPREOuCGbU8S6295486D1kbu76cEDxE+adca3/9+qo
7FQVugkRjBD0ZOj/3qcuKDOh6ZsalF9LaaNMPNWNiQ5n3bWBnQ1tMMEes58JvoNgjn9FOz
hbOdOe91K/OdRA0/YzrguaCA6/vE/tUP+xDD/GOu7KyvN1dsgo1vnYZLG7p8vGgt61eTyC
V6qMEkceGatQvfiBb4XZCeODtC2KBv4pbJpt1zPKOpF4XFb2LferPxAL9rsSRSUOk9tZnc
x1GM3+UUYwt9dF8bcUfomZCKd07kzPh206nZr/uCElXVtCqghgVRQW8uiFRh6ycVWY/pBq
uhPfiHKHilZEah00c08ax14XTK89ovJzjbHezaV/NghkfWpn3W7gDNJTbLbXPbrLDkJBPJ
IltJ5QqwVK/Hi+69x9CxFOkyNpxWFexHPIeq4q0liPoah42MBPAQ130bWULgBP+K0ugzqQ
cSPAhi9FMq6ZVFTmaiPX8JH8JAcswd500x9jMmV91obzTZmXAQsfVpi0asxRhfficeIifs
UJ/FHwW2p13YmDVH1AjVmCDn9T46I05Cq+ImrUBX+JAEa6yQU6R6/s7maVDqpdtkpFp0q1
CWQHHw9J1fYS4w==
---- END SSH2 PUBLIC KEY ----
[user@client ~]$

```

2. On the SBC, use the **ssh-key** command with the **authorized-key import** argument.

The command syntax:

```
ssh-key authorized-key import <name> <class>
```

The **<name>** parameter is the identifier for the SSH client. The **<class>** is one of the two authorization classes on the SBC: either **user** or **admin**.

```
ORACLE# ssh-key authorized-key import bob admin
```

IMPORTANT:

Please paste SSH public key in the format defined in RFC 4716.
Terminate the key with ";" to exit.....

```

---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by bob@client from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAQADOTDujYoQXzjTt9I8YvJMvfSV1WZ6iDzfrX06R3l
Rj/lrjxlWDMc/Y/uEd2sJ+5wdlCnJPREOuCGbU8S6295486D1kbu76cEDxE+adca3/9+qo
7FQVugkRjBD0ZOj/3qcuKDOh6ZsalF9LaaNMPNWNiQ5n3bWBnQ1tMMEes58JvoNgjn9FOz
hbOdOe91K/OdRA0/YzrguaCA6/vE/tUP+xDD/GOu7KyvN1dsgo1vnYZLG7p8vGgt61eTyC
V6qMEkceGatQvfiBb4XZCeODtC2KBv4pbJpt1zPKOpF4XFb2LferPxAL9rsSRSUOk9tZnc
x1GM3+UUYwt9dF8bcUfomZCKd07kzPh206nZr/uCElXVtCqghgVRQW8uiFRh6ycVWY/pBq
uhPfiHKHilZEah00c08ax14XTK89ovJzjbHezaV/NghkfWpn3W7gDNJTbLbXPbrLDkJBPJ
IltJ5QqwVK/Hi+69x9CxFOkyNpxWFexHPIeq4q0liPoah42MBPAQ130bWULgBP+K0ugzqQ
cSPAhi9FMq6ZVFTmaiPX8JH8JAcswd500x9jMmV91obzTZmXAQsfVpi0asxRhfficeIifs
UJ/FHwW2p13YmDVH1AjVmCDn9T46I05Cq+ImrUBX+JAEa6yQU6R6/s7maVDqpdtkpFp0q1
CWQHHw9J1fYS4w==
---- END SSH2 PUBLIC KEY ----;

```

 **Note:**

If the Admin Security entitlement is enabled, the SSH client keys must be at least 2048 bits.

 **Note:**

Oracle recommends keys be at least 2048 bits.

3. Save and activate the configuration.

Export an Authorized Key

To export a previously imported SSH public key, use the **ssh-key** command with the **authorized-key export** argument.

1. List the available **ssh-key** elements.

```
ORACLE# show running-config ssh-key
ssh-key
      name                bob
      type                authorized-key
      encryption-type     rsa
      size                4096
      last-modified-by    admin@10.0.0.20
      last-modified-date  2020-05-12 13:58:39
ssh-key
      name                alice
      type                authorized-key
      encryption-type     rsa
      size                4096
      last-modified-by    admin@10.0.0.37
      last-modified-date  2020-05-12 14:23:47
ssh-key
      name                logserver
      type                known-host
      encryption-type     rsa
      size                2048
      last-modified-by    admin@10.0.0.37
      last-modified-date  2020-05-11 15:18:36
```

2. For any **ssh-key** element whose type is **authorized-key**, use the **ssh-key authorized-key export <name>** command to export the user's public key.

```
ORACLE# ssh-key authorized-key export bob
public-key 'bob' (RFC 4716/SECSH format):

---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit rsa"
AAAAB3NzaC1yc2EAAAADAQABAAQADOTDujYoQXzjTt9I8YvJMvfSVlWZ6iDzfrx06R
3l
Rj/lrjxlWDMc/Y/
uEd2sJ+5wdlCnJPREOuCGbU8S6295486D1kbu76cEDxE+adca3/9+qo
7FQVugkRJBDOZOj/
3qcuKDOh6ZsalF9LaaNMPNWNiQ5n3bWBnQ1tMMEes58JvoNgjn9FOz
hbOdOe91K/OdRA0/YzrguaCA6/vE/tUP+xDD/
GOu7KyvN1dsgo1vnYZLG7p8vGgt61eTyC
V6qMEkceGatQvfiBb4XZCeODtC2KBv4pbJpt1zPKOpF4XFb2LferPxAL9rsSRSUOk9tZ
Nc
```

```
x1GM3+UUYwT9dF8bcUfomZCKd07kzPh206nZr/uCElXVtCqghgVRQW8uiFRh6ycVWVY/pBq
uhPfiHKHilZEah00c08ax14XTK89ovJzjbHezaV/NghkfWpn3W7gDNJTbLbXPbrLDkJBPJ
IltJ5QqWVK/Hi+69x9CxFOkyNpxWFexHPieq4q0liPoah42MBPAQl30bWULgBP+K0ugzqQ
cSPAhi9FMq6ZVFTmaiPX8JH8JAcswd500x9jMmV91obzTZmXAQsfVpi0asxRhfficeIifs
UJ/FHwW2p13YmDVH1AjVmCDn9T46I05Cq+ImrUBX+JAEa6yQU6R6/s7maVDqpdtkpFp0q1
CWQHhw9J1fYS4w==
---- END SSH2 PUBLIC KEY ----
```

```
ORACLE#
```

Delete an Authorized Key

To delete a previously imported SSH public key, use the **ssh-key** command with the **authorized-key delete** argument.

1. List the available **ssh-key** elements.

```
ORACLE# show running-config ssh-key
ssh-key
      name                bob
      type                authorized-key
      encryption-type    rsa
      size                4096
      last-modified-by   admin@10.0.0.20
      last-modified-date 2020-05-12 13:58:39
ssh-key
      name                alice
      type                authorized-key
      encryption-type    rsa
      size                4096
      last-modified-by   admin@10.0.0.37
      last-modified-date 2020-05-12 14:23:47
ssh-key
      name                logserver
      type                known-host
      encryption-type    rsa
      size                2048
      last-modified-by   admin@10.0.0.37
      last-modified-date 2020-05-11 15:18:36
```

2. For any **ssh-key** element whose type is **authorized-key**, use the **ssh-key authorized-key delete <name>** command to delete the user's public key.

```
ORACLE# ssh-key authorized-key delete bob
SSH public key deleted successfully...
WARNING: Configuration changed, run "save-config" command to save it
and run "activate-config" to activate the changes
ORACLE#
```

3. Save and activate the configuration.

Add an SSH Known Host Key

For the SBC to authenticate over SSH to an SFTP server, the public key of the SFTP server needs to be imported into the `known_hosts` file of the SBC.

1. Convert the public key of the SFTP server into RFC 4716 format.

There are two ways to do this.

- a. SSH to the SFTP server and run the **ssh-keygen** command on the server's host key.

For OpenSSH implementations, host keys are generally found at `/etc/ssh/ssh_host_rsa_key.pub`. Other SSH implementations may differ. To do this on Oracle Linux, use the **ssh-keygen** command.

```
[user@logserver ~]$ ssh-keygen -e -f /etc/ssh/
ssh_host_rsa_key.pub
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "2048-bit RSA, converted by user@logserver from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAQDwifpOpBKoDhzJXglzdoOfZ39TiU7jhygbP
GQTW0
j3zISW57PRbSulVwlhBHwqJwZZc6nr1JXaiHN7ieYT/
96QCXQ56JH9Lcjej6iHplfhJO44
qIgzI1RtD0e5y6YBzDgcI3T8J6n0jHwksvwKttObk8SoZ11mqE4xPXSiTVB1PzMNx
F0dWV
rgvGK227PsOfPLypL3RhnmqFbVRihMKW7a80p7I+T6mAoq8UdzejbyhEK+e0Ge3F9
ilg49
oHWHNnSvU64F1ADybbZrclvvt8vofIzraGMBRjLs5Yl8bbdId/
4UBcilfONmIUzxVse5NM
PwNj0cjevNPS1/LOcKUgQxN
---- END SSH2 PUBLIC KEY ----
[user@logserver ~]$
```

- b. Run the **ssh-keyscan** command from a Linux client and convert that key with the **ssh-keygen** command.

```
ssh-keyscan -t rsa 10.0.0.6 | sed 's/.*ssh/ssh/' > key.pub
ssh-keygen -ef key.pub
```

2. On the SBC, use the **ssh-key** command to import the host key of the SFTP server into the `known_hosts` file of the SBC.

The command syntax:

```
ssh-key known-host import <name>
```

For SFTP push to work properly, the `<name>` parameter must be the IP address or hostname of the SFTP server.

```
ORACLE# ssh-key known-host import 10.0.0.12
```

3. Paste the public key with the bracketing Begin and End markers at the cursor point.
4. Enter a semi-colon (;) to signal the end of the imported host key.

The entire import sequence is shown below.

```
ORACLE# ssh-key known-host import 10.0.0.12
```

IMPORTANT:

Please paste SSH public key in the format defined in RFC 4716.
Terminate the key with ";" to exit.....

```
---- BEGIN SSH2 PUBLIC KEY ----
```

```
Comment: "2048-bit RSA, converted by user@logserver from OpenSSH"
```

```
AAAAB3NzaC1yc2EAAAADAQABAAQDJXglzdiU7jhywifpOpBKOdhoOfZ39TzgbPGQTW0
j357PRbSulHwaiHN7zEVwlhBISWie6nrQ56JH9Lcjej1JX96QCYT/qJwZZcX6iHplfhJO4
q8J6nIlRtD0e5y60jHwgZYBzDksvwKk8SSiTVB10ttObdWVoZl1mqPzMNxFE4xPXIgcI3T
rgvGKR27PsOfPLY80p7IpLhnmqFjbyhEK+e0KW7a+T6mbV23RIhMzeAoq8UdGe3F9i1g49
oHws5mDybHnBRjLbZrcSvU64F1AMlvvtUzxVse5NM8vofIzraGIYl8bbdId/4UBci1fON
PwNPS1/LONj0cjcKUGQxN
```

```
---- END SSH2 PUBLIC KEY ----;
```

SSH public key imported successfully....

WARNING: Configuration changed, run "save-config" command to save it
and run "activate-config" to activate the changes

Import both the RSA key and the DSA key if you are not sure which one the SFTP server uses.

5. Save and activate the configuration.

Delete an SSH Known Hosts Key

Delete expired SSH keys from the known_hosts file of the SBC.

1. List the available **ssh-key** elements.

```
ORACLE# show running-config ssh-key
```

```
ssh-key
  name                bob
  type                authorized-key
  encryption-type     rsa
  size                4096
  last-modified-by    admin@10.0.0.20
  last-modified-date  2020-05-12 13:58:39
ssh-key
  name                alice
  type                authorized-key
  encryption-type     rsa
  size                4096
  last-modified-by    admin@10.0.0.37
  last-modified-date  2020-05-12 14:23:47
ssh-key
  name                10.0.0.12
  type                known-host
  encryption-type     rsa
  size                2048
```



```
last-modified-by      admin@10.0.0.37
last-modified-date    2020-05-11 15:18:36
```

2. Use the **ssh-key** command to remove a key whose type is known-host.

The command syntax:

```
ssh-key known-host delete <name>
```

The <name> parameter is an alias or handle assigned to the imported host key.

```
ORACLE# ssh-key known-host delete 10.0.0.12
```

3. Save and activate the configuration.

Add a Certificate Authority Key

When authenticating with certificates, clients send certificates to establish their identity and authorization. The public key of the Certificate Authority (CA) used for signing these client certificates must be imported into the SBC.

1. On the server you'll use for a certificate authority, create a `keys` directory for storing keys.

```
[user@host ~]$ mkdir keys
[user@host ~]$ cd keys/
```

2. Generate an SSH key pair to use for signing certificates.

```
[user@host keys]$ ssh-keygen -t rsa -b 4096 -f ./ca_key
```

3. Export the CA key to RFC 4716 format.

```
[user@host keys]$ ssh-keygen -ef ./ca_key.pub
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by user@host from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAQDOTDujYoQXzjTt9I8YvJMvfSVlWZ6iDzFRx06R
3l
Rj/lrjxlWDMc/Y/
uEd2sJ+5wdlCnJPREOuCGbU8S6295486D1kbu76cEDxE+adca3/9+qo
7FQVugkrJBD0Z0j/
3qcuKDOh6ZsalF9LaaNMPNWNiQ5n3bWBnQ1tMMEes58JvoNgjn9FOz
hbOdOe91K/OdRA0/YzrguaCA6/vE/tUP+xDD/
GOu7KyvN1dsgo1vnYZLG7p8vGgt61eTyC
V6qMEkceGatQvfiBb4XZCeODtC2KBv4pbJpt1zPKOpF4XFb2LferPxAL9rsSRSUOk9tZ
Nc
x1GM3+UUYwT9dF8bcUfomZCKd07kzPh206nZr/
uCElXVtCqghgVRQW8uiFRh6ycVWY/pBq
uhPfiHkHilZEah00c08ax14XTK89ovJzjbHezaV/
NghkfWpn3W7gDNJTB LbxpbrLDkJBJ
IltJ5QqwVK/
Hi+69x9CxFOkyNpxWFexHPieq4q0liPoah42MBPAQ130bWULgBP+K0ugzqQ
cSPAhi9FMq6ZVFTmaiPX8JH8JAcswd500x9jMmV91obzTZmXAQsfVpi0asxRhfficeI
fs
```

```

UJ/FHwW2p13YmDVH1AjVmCDn9T46I05Cq+ImrUBX+JAEa6yQU6R6/s7maVDqpdtkpFp0q1
CWQHHw9J1fYS4w==
---- END SSH2 PUBLIC KEY ----
[user@host keys]$

```

4. Import the CA key into the SBC using the **ssh-key** command with the **ca-key import** argument.

The command syntax:

```
ssh-key ca-key import <key-name> <class>
```

The **<key-name>** parameter is the key identifier or key ID that will be used when signing client keys as the value of the **-I** argument in the **ssh-keygen** command. The **<class>** is one of the two authorization classes on the SBC: either **user** or **admin**.

```
ORACLE# ssh-key ca-key import rootCA admin
```

IMPORTANT:

Please paste SSH public key in the format defined in RFC 4716.
Terminate the key with ";" to exit.....

```

---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by user@server from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAQDOTDujYoQXzjTt9I8YvJMvfSV1WZ6iDzfrX06R3l
Rj/lrjxlWDMc/Y/uEd2sJ+5wdlCnJPREOuCGbU8S6295486D1kbu76cEDxE+adca3/9+qo
7FQVugkRjBD0Z0j/3qcuKDOh6ZsalF9LaaNMPNWNiQ5n3bWBnQ1tMMEes58JvoNgjn9FOz
hbOdOe91K/OdRA0/YzrguaCA6/vE/tUP+xDD/GOu7KyvN1dsgo1vnYZLG7p8vGgt61eTyC
V6qMEkceGatQvfiBb4XZCeODtC2KBv4pbJpt1zPKOpF4XFb2LferPxAL9rsSRSUOk9tZnc
x1GM3+UUYwt9dF8bcUfomZCKd07kzPh206nZr/uCElXVtCqghgVRQW8uiFRh6ycVWVY/pBq
uhPfiHkHilZEah00c08ax14XTK89ovJzjbHezaV/Nghkfwpn3W7gDNJTbLbXPbrLDkJBPJ
IltJ5QqWVK/Hi+69x9CxFOkyNpxWFexHPIeq4q0liPoah42MBPAQl30bWULgBP+K0ugzqQ
cSPAhi9FMq6ZVFTmaiPX8JH8JAcswd500x9jMmV91obzTZmXAQsfVpi0asxRhfficeIifs
UJ/FHwW2p13YmDVH1AjVmCDn9T46I05Cq+ImrUBX+JAEa6yQU6R6/s7maVDqpdtkpFp0q1
CWQHHw9J1fYS4w==
---- END SSH2 PUBLIC KEY ----;

```

 **Note:**

If the Admin Security entitlement is enabled, the key must be at least 2048 bits.

5. Save and activate the configuration.
6. For each SSH client, copy the client's public key into the **keys** directory.

```
[user@host keys]$ scp acme@client1.com:~/.ssh/id_rsa.pub ./id_rsa.pub
```

7. Sign the key with the **ssh-keygen** command.

Use the following arguments:

- Use **-s** to identify the private key of the CA key used to sign.

- Use `-z` to specify the serial number to be embedded in the certificate to distinguish this certificate from others signed by the same CA.
- Use `-n` to specify the username of the client to be included in the certificate.
- Use `-I` to specify the key ID. This key ID must match the `<key-name>` specified when importing the signing CA key into the SBC.
- Use `-v` to set the validity interval. To set the validity for one year, starting the previous day, use `-1d:+52w`.

! Important:

The username passed with the `-n` argument of the **ssh-keygen** command must match the username used to authenticate.

Note:

If the **type** attribute of the **authentication** element is set to **local**, the username passed with the `-n` argument must be set to `admin`.

```
[user@host keys]$ ssh-keygen -s ca_key -z 1 -n admin -I rootCA -V
-1d:+52w id_rsa.pub
Signed user key id_rsa.pub: id "rootCA" serial 1 for admin valid
from 2020-06-21T09:26:41 to 2021-06-21T09:26:41
[user@host keys]$
```

8. Copy the certificate to the client's `.ssh` directory.

```
[user@host keys]$ scp id_rsa-cert.pub acme@client1.com:~/.ssh/
```

9. Verify the SSH client can connect with the certificate.

Delete a Certificate Authority Key

To delete a previously imported Certificate Authority (CA) key, use the **ssh-key** command with the **ca-key delete** argument.

1. List the available **ssh-key** elements.

```
ORACLE# show running-config ssh-key
ssh-key
    name                bob
    type                authorized-key
    encryption-type     rsa
    size                4096
    last-modified-by    admin@10.0.0.20
    last-modified-date  2020-05-12 13:58:39
ssh-key
    name                alice
    type                authorized-key
    encryption-type     rsa
```

```

size 4096
last-modified-by admin@10.0.0.37
last-modified-date 2020-05-12 14:23:47
ssh-key
name rootCA
type ca-key
encryption-type rsa
size 4096
last-modified-by admin@10.0.0.37
last-modified-date 2020-05-11 15:18:36

```

2. For any **ssh-key** element whose type is **ca-key**, use the **ssh-key ca-key delete <key-name>** command to delete the CA key.

```

ORACLE# ssh-key ca-key delete rootCA
SSH public key deleted successfully...
WARNING: Configuration changed, run "save-config" command to save it
and run "activate-config" to activate the changes
ORACLE#

```

3. Save and activate the configuration.

Revoke a User Key

To revoke access to a specific user whose public key was signed by your CA key, import the user's public key into the revocation list.

1. On the SBC, use the **ssh-key** command with the **ca-user-revoke import** argument.

The command syntax:

```
ssh-key ca-user-revoke import <key-name>
```

The **<key-name>** parameter uniquely identifies the key you want to revoke.

```
ORACLE# ssh-key ca-user-revoke import bob
```

IMPORTANT:

Please paste SSH public key in the format defined in RFC 4716.
Terminate the key with ";" to exit.....

```

---- BEGIN SSH2 PUBLIC KEY ----
Comment: "4096-bit RSA, converted by user@server from OpenSSH"
AAAAB3NzaC1yc2EAAAADAQABAAQADODTdujYoQXzjTt9I8YvJMvfSVlWZ6iDzfRx06R3l
Rj/lrjxlWDMc/Y/uEd2sJ+5wdlCnJPREOuCGbU8S6295486D1kbu76cEDxE+adca3/9+qo
7FQVugkRJBDOZoj/3qcuKDOh6ZsalF9LaaNMPNWNiQ5n3bWBnQ1tMMEes58JvoNgjn9FOz
hbOdOe91K/OdRA0/YzrguaCA6/vE/tUP+xDD/GOu7KyvN1dsgolvnYZLG7p8vGgt61eTyC
V6qMEkceGatQvfiBb4XZCeODtC2KBv4pbJpt1zPKOpF4XFb2LferPxAL9rsSRSUOk9tZnc
x1GM3+UUYwT9dF8bcUfomZCKd07kzPh206nZr/uCElXVtCqghgVRQW8uiFRh6ycVWY/pBq
uhPfiHkHilZEah00c08ax14XTK89ovJzjbHezaV/NghkfWpn3W7gDNJTBbLbpxbrLDkJBPJ
IltJ5QqgVK/Hi+69x9CxFOkyNpxWFexHPIeq4q0liPoah42MBPAQl30bWULgBP+K0ugzqQ
cSPAhi9FMq6ZVFTmaiPX8JH8JAcswd500x9jMmV91obzTZmXAQsfVpi0asxRhfficeIifs
UJ/FHwW2p13YmDVH1AjVmCDn9T46I05Cq+ImrUBX+JAEa6yQU6R6/s7maVDqpdtkpFp0ql

```

```
CWQHw9J1fYS4w==  
---- END SSH2 PUBLIC KEY ----;
```

2. Save and activate the configuration.

The user's key is added to the revocation list. When authenticating to the SBC, the user may no longer use his or her key or certificate, even though that key was signed by the CA key.

Unrevoke a Revoked User Key

If a user key is added to the revocation list, that user will not be able to authenticate to the SBC. To delete a key from the revocation list, use the **ssh-key** command with the **ca-user-revoke delete** argument.

1. List the available **ssh-key** elements.

```
ORACLE# show running-config ssh-key  
ssh-key  
    name                bob  
    type                authorized-key  
    encryption-type    rsa  
    size                4096  
    last-modified-by   admin@10.0.0.20  
    last-modified-date 2020-05-12 13:58:39  
ssh-key  
    name                alice  
    type                authorized-key  
    encryption-type    rsa  
    size                4096  
    last-modified-by   admin@10.0.0.37  
    last-modified-date 2020-05-12 14:23:47  
ssh-key  
    name                alice  
    type                ca-user-revoke  
    encryption-type    rsa  
    size                4096  
    last-modified-by   admin@10.0.0.37  
    last-modified-date 2020-05-11 15:18:36
```

2. For any **ssh-key** element whose type is **ca-user-revoke**, use the **ssh-key ca-user-revoke delete <key-name>** command to delete the CA key.

```
ORACLE# ssh-key ca-user-revoke delete alice  
SSH public key deleted successfully...  
WARNING: Configuration changed, run "save-config" command to save it  
and run "activate-config" to activate the changes  
ORACLE#
```

3. Save and activate the configuration.

Once the user key is removed from the revocation list, the functionality of any existing key is restored.

SFTP Operations

SFTP performs all operations over an encrypted SSH connection. It may also use many features of SSH, such as public key authentication and compression. SFTP connects and logs into the specified host, then enters an interactive command mode.

Once in interactive mode, SFTP understands a set of commands similar to those of FTP. Commands are case insensitive and pathnames may be enclosed in quotes if they contain spaces.

The following lists supported SFTP commands:

- `bye`—Quit SFTP.
- `cd pathChange`—Remote directory to path.
- `lcd pathChange`—Local directory to path.
- `chgrp grp path`—Change group of file path to group. group must be a numeric GID.
- `chmod mode path`—Change permissions of file path to mode.
- `chown own path`—Change owner of file path to own. own must be a numeric UID.
- `dir (or ls)`—List the files in the current directory.
- `exit`—Quit SFTP.
- `get [flags] remote-path [local-path]`—Retrieve the remote-path and store it on the local machine. If the local path name is not specified, it is given the same name it has on the remote machine. If the `-P` flag is specified, then the file's full permission and access time are copied too.
- `help`—Display help text.
- `lcd`—Change the directory on the local computer.
- `lls`—See a list of the files in the current directory.
`lls [ls-options [path]]`—Display local directory listing of either path or current directory if path is not specified.
- `mkdir path`—Create local directory specified by path.
- `ln oldpath newpath`—Create a symbolic link from oldpath to newpath.
- `lpwd`—Print local working directory.
- `ls [path]`—Display remote directory listing of either path or current directory if path is not specified.
- `lumask umask`—Set local umask to umask.
- `mkdir path`—Create remote directory specified by path.
- `put [flags] local-path [local-path]`—Upload local-path and store it on the remote machine. If the remote path name is not specified, it is given the same name it has on the local machine. If the `-P` flag is specified, then the file's full permission and access time are copied too.
- `pwd`—Display remote working directory.
- `quit`—Quit SFTP.
- `rename oldpath newpath`—Rename remote file from oldpath to newpath.
- `rmdir path`—Remove remote directory specified by path.

- `rm path`—Delete remote file specified by path.
- `symlink oldpath newpath`—Create a symbolic link from oldpath to newpath.
- `! command`—Execute command in local shell.
- `!—Escape to local shell.`
- `?—Synonym for help.`



Note:

Command availability is subject to Oracle authorization/privilege classes. Some SFTP commands are available to only certain users; some commands are available to no users.

RADIUS file access privileges are specified by the Acme-User-Privilege VSA, which can take the following values.

- `sftpForAudit`—allows audit log access
- `sftpForAccounting`—allows system logs to be accessed
- `sftpForHDR`—allows HDR (Historical Data Records) to be accessed
- `sftpForAll`—allows all logs to be accessed

Secure Radius Connection

The ESBC can connect to a Radius server over a secure IPSec/IKEv2 connection over a media interface.



Note:

You must have the IPSec license installed to enable Radius over a secure IPSec/IKEv2 connection.

To properly configure a secure Radius connection, the following config elements and parameters must be configured:

- **security, authentication**
 - **type** (set to **radius**)
 - **server-assigned-privilege** (set to **enabled**)
 - **management-servers**
- **security, authentication, radius-server**
 - **address** (the Radius server IP)
 - **secret**
 - **nas-id**
 - **realm-id**

- **security, ike, ike-config**
 - **log-level**
 - **phase1-dh-mode**
 - **phase2-exchange-mode**
 - **red-port-options**
- **security, ike, ike-interface**
 - **ike-version** (set to 2)
 - **address**
 - **realm-id**
 - **ike-mode**
 - **esnSupport** (set to **enabled**)
 - **shared-password**
 - **eap-protocol**
- **security, ike, ike-sainfo**
 - **name**
 - **tunnel-local-addr**
 - **tunnel-remote-addr**
- **security, ipsec, security-policy**
 - **name**
 - **network-interface**
 - **priority**
 - **local-ip-addr-match**
 - **remote-ip-addr-match**
 - **ike-sainfo-name**

Factory Reset for the Oracle Communications Session Border Controller

If you attempt to remove the Admin Security feature, some irrevocable changes and information remain on the system. You can return your platforms to their initial factory settings (zeroization) to truly remove all traces of the previous implementation. Depending on if you are performing this on an Acme Packet hardware platform or a Virtual platform, the process is different.

▲ Caution:

Factory reset erases system data, including licenses and configuration, and reboots the supported Acme Packet platforms using the factory default `/boot/bzImage` file. If the factory image file has been removed, the system will NOT be recoverable without manual intervention, and you may have to return the system to Oracle for factory re-initialization.

Factory reset is not a secured erase process. After a factory reset there is system data in `/opt/logs` and `/opt/crash`. Use the ACLI command **format system-disk** to erase the data from `/opt` directory.

Using the Oracle Rescue Account for PNF Zeroization

To enable the Oracle Rescue Account:

1. Connect to the SBC's serial console.
2. Reboot the SBC and press the spacebar to interrupt the 5 second bootloader countdown.
3. Select **o** to access the Oracle Rescue Account.

A challenge string displays in the console.

4. Contact Oracle Support and provide the challenge string and the system serial number.

Oracle Support verifies the challenge string and provides a response string.

5. Enter the response string.

If it is validated, access is granted to the Oracle Rescue Account and a sub-menu appears providing three menu options:

- **f**—Factory default
- **!**—Start debug shell
- **x**—Exit to main menu

```
Starting acmeboot...
```

```
ACME bootloader Acme Packet SCZ<build#> RTM (Build 59) 201706021530
```

```
Press the space bar to stop auto-boot...
```

```
28
```

```
Please contact Oracle Product Support to obtain a Response Key
```

```
You will need to provide the following information:
```

1. Serial number of the system
2. This Challenge Key: 069-033-231-180

```
Note: Keys are valid for a limited period only, typically 1 day
```

```
Enter response key:      006-163-164-054
```

```
Oracle Rescue Access Menu
```

```
PROCEED WITH CAUTION: You are now in privileged access mode.
Use of these commands is permitted by authorised personnel only.
f          - factory default
!          - start debug shell

x          - exit to main menu

[Oracle Rescue Access]: f

WARNING WARNING WARNING
This command will permanently erase the hard disk, nvram and flash,
returning the system to a factory-default state.

Type: "ERASE_ALL" to confirm factory default, anything else will abort.
[Confirm Factory Default]: ERASE_ALL

Proceeding with factory default. DO NOT INTERRUPT
Removing hard disk user data partitions...
Wiping /code filesystem...
Zeroizing /code filesystem...
Wiping /boot filesystem...
Zeroizing /boot filesystem...
Zeroizing NVRAM...
Checking for NVRAM zeroization...
Setting default boot params...

Completed factory default. Reboot or power off now

Rebooting...
```

Reinstalling the VM for VNF Installation

To perform zeroization on a VM, you must perform a complete image reinstallation.