Oracle® Communications Session Border Controller

Administrative Security Guide

Release S-CZ8.1.0
F20247-01
October 2019

Oracle Communications Session Border Controller Administrative Security Guide, Release S-CZ8.1.0

F20247-01

Contents

2 Audit Log

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 (OCSBC).

This guide covers:

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

Oracle 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.

Related Documentation

The following table lists the members that comprise 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 Oracle Communications Session Border Controller.
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 Oracle Communications Session Border Controller 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 Oracle Communications Session Border Controller's accounting support, including details about RADIUS and Diameter accounting.
HDR Resource Guide	Contains information about the Oracle Communications Session Border Controller's Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Administrative Security Essentials	Contains information about the Oracle Communications Session Border Controller's support for its Administrative Security license.
Security Guide	Contains information about security considerations and best practices from a network and application security perspective for the Oracle Communications Session Border Controller 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.
Header Manipulation Rule Guide	Contains information about configuring and using Header Manipulation Rules to manage service traffic.

Revision History

Date	Description
November 2018	• Initial release
October 2019	• Fixes product name in "Admin Security ACP Feature"

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 (OCSBC).

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 OCSBC may retain sensitive information. To remove all sensitive data, you must perform a complete factory reset (zeroization). 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 OCSBC, 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 OCSBC 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
- SSH keys are denied
- history log access is denied
- password policy features are enabled in addition to some additional Admin Security specific password requirements
- access to the Session Element Manager (SEM) in the Session Delivery manager (SDM) is blocked
- ACP (Acme Control Protocol) is blocked

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
- access to the SEM in the SDM is granted
- ACP is blocked

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 ACLI Configuration Guide*.

Supported Platforms

The following platforms support Admin Security:

- Acme Packet 1100
- Acme Packet 3900
- Acme Packet 4600
- Acme Packet 6300
- VMWare

JITC Support

The Oracle Communications Session Border Controller (OCSBC) supports Joint Interoperability Testing Command (JITC). The Admin Security feature set largely encompasses JITC features with one main difference. Instead of sending ACP over TCP (potentially exposing sensitive information) JITC allows ACP over TLS.

> **Note:**
>
> The JITC feature set is supported only on OESBC releases only.

When both Admin Security and Federal Information Processing Standards (FIPS) feature sets are enabled on the OCSBC, . When both are provisioned and you execute the **show licenses** and **show entitlements** commands, the OCSBC displays JITC.

Provision the JITC feature by enabling the Advanced Security Suite 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 ACLI Configuration Guide*.

> **Note:**
>
> As of Release ECZ7.5.0 and later, JITC supersedes all Admin Security features, while behavior for Admin Security features acquired prior to ECZ7.5.0 remain unchanged.

Supported Platforms

The following platforms support JITC mode:

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

Admin Security ACP Feature

The Administrative Security ACP feature adds more password security and opens the ACP port, allowing the OCSBC to connect to the Oracle Communications Session Delivery Manager (OCSM).

The Admin Security ACP feature inherits the rules of the Admin Security feature set and imposes additional rules and restrictions to improve password strength. For information on obtaining an Admin Security with ACP license key, contact your Oracle representative.

For information on the additional password length/strength requirements supported with the Admin Security with ACP feature, see *Password Policy*.

Set the **password-policy**, **password-policy-strength** parameter to **enabled** to enable the enhanced password strength requirements. To retain only the password requirements defined by the Admin Security feature, leave this parameter set to **disabled**. For more information on configuring Admin Security with ACP password policies, see *Configuring the Admin Security with ACP Password Rules*.

Login Banner

Upon successful user authentication/authorization, the Oracle OCSBC displays the login banner.

Login Banner

- Last login: displays the date and time that the current user (admin in this case) 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 (admin in this case)
- 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 OCSBC.

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

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 restricts access to the ACP ports and mandates the following password length/strength requirements.

  - user password must contain at least 9 characters (Admin Security only)
  - 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
  - passwords cannot contain, repeat, or reverse the user name
  - passwords cannot contain three consecutive identical characters

The Admin Security ACP add-on feature imposes the same password length/strength requirements as above except for the minimum length requirement, and also provides access to the ACP ports.

When you set the **password-policy**, **password-policy-strength** config property to **enabled** as part of the Admin Security ACP feature, you impose the following requirements in addition to those enforced with the Admin Security feature:

- passwords cannot contain two or more characters from the user ID
- passwords cannot contain a sequence of three or more characters from any password contained in the password history cache
- passwords cannot contain a sequence of two or more characters more than once
- passwords cannot contain either sequential numbers or characters, or repeated characters more than once.

In the absence of the Admin Security ACP feature, you may safely ignore the **password-policy-strength** config property and retain the default value (**disabled**). For more information, see *Configuring the Admin Security with ACP Password Rules*.

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

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
   password-policy-strength    disabled
   ```

2. The **min-secure-pwd-length** command is ignored when the Admin Security with ACP feature is installed and the **password-policy-strength** configuration element is set to **enabled**.

3. 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)#
   ```

4. 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)#
   ```

5. 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)#
   ```

6. 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)#
   ```

7. 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)#
   ```

8. (Optional) Use the **password-policy-strength** command to enable the enhanced password strength requirements.

   In the absence of the Admin Security ACP feature set, this command can be safely ignored.

   password-policy-strength may be enabled when the Admin Security with ACP feature is enabled. This feature includes all of the password security features contained in the Admin Security feature set and also adds password strength requirements beyond those imposed by Admin Security. Specific new requirements are as follows:

- passwords cannot contain two or more characters from the user ID
  For example, given a user ID of administrator, the password thispasswordistragic is not allowed because istra is a substring of administrator
- passwords cannot contain a sequence of three or more characters from any password contained in the password history cache
- passwords cannot contain a sequence of two or more characters more than once
  For example, ...w29W29... is legal; ...w29W29&&29... is not.
- passwords cannot contain either sequential numbers or characters, or repeated characters more than once
  For example, ‘66666’, ‘aaaa’, ‘abcd’, ‘fedc’, ‘1234’, ‘7654'.

  For example, 666, aaa abcd, fedc, 1234, and 7654 all render a password illegal.

In the absence of the Admin Security ACP feature, retain the default value (**disabled**). With the Admin Security with ACP feature installed, use **enabled** to add the new password requirements as listed above; use **disabled** to retain only the password requirements defined by Admin Security.

```
ORACLE(password-policy)# password-policy-strength enabled
ORACLE(password-policy)#
```

9. 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)#
   ```

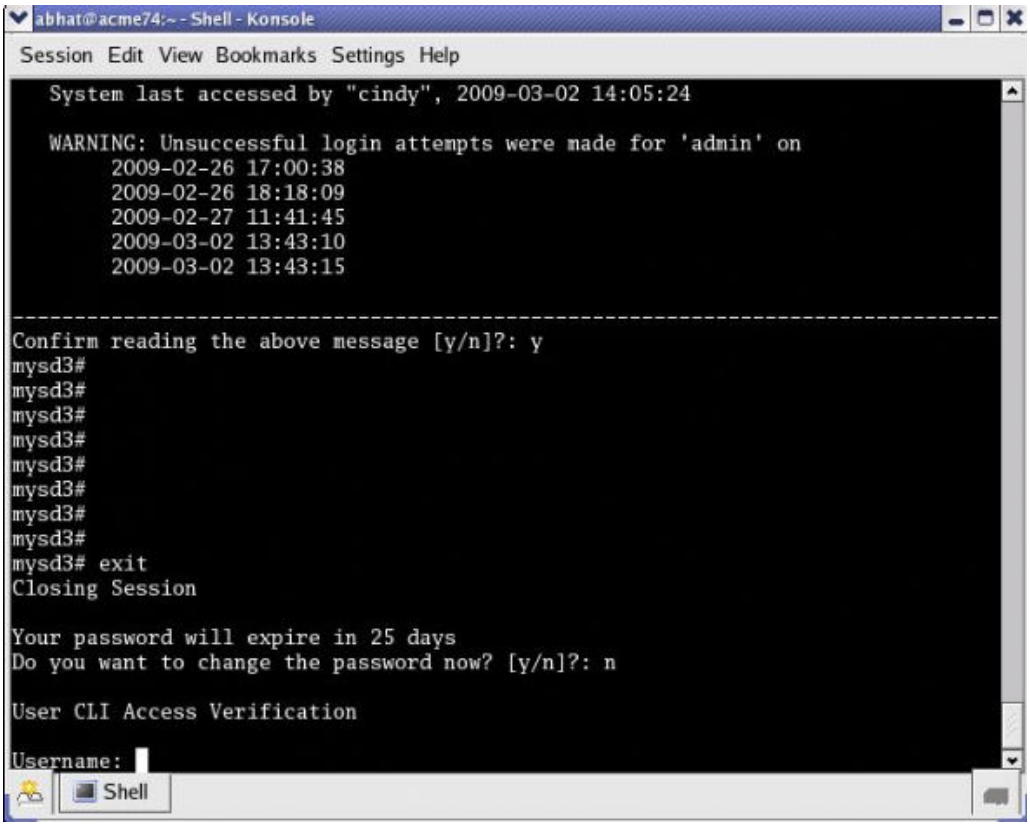
2. Type **select**, and press ENTER.
3. Type **show**, and press ENTER.
4. Configure the following password policy settings, as needed:

- **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.

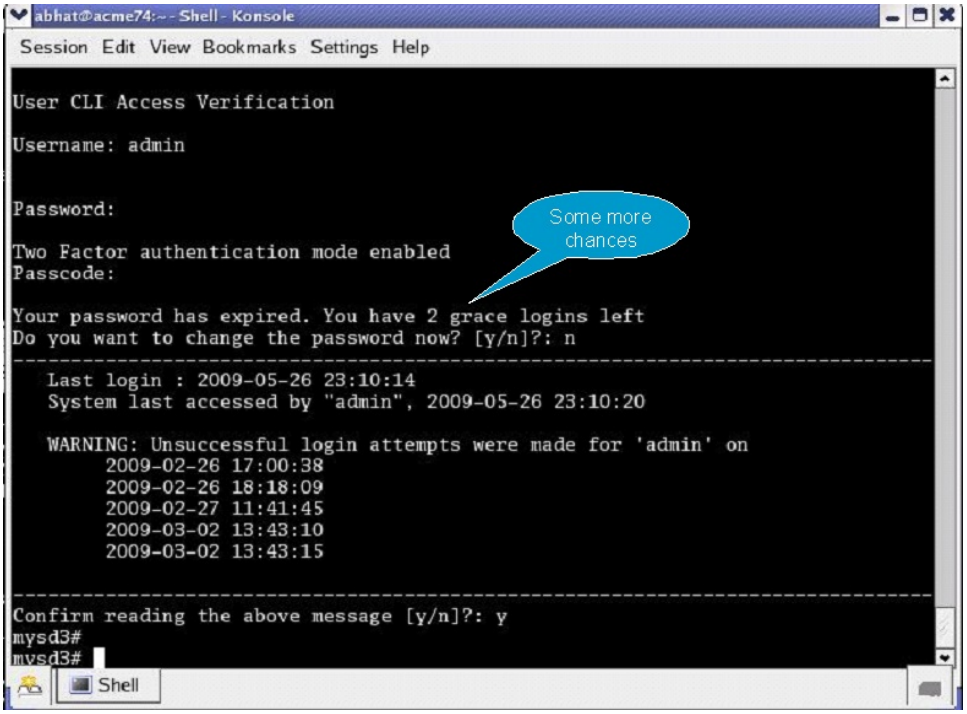
Changing a Password

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.



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.



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.

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

   Similarly, the admin 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 user password from the # (admin) prompt.

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 in response to the Enter new password : prompt.

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

3. Confirm the password in response to the Enter password again : prompt.

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

Changing the admin Password

Change the admin password from the # (admin) prompt.

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 in response to the Enter new password : prompt.

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

3. Confirm the password in response to the Enter password again : prompt.

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

Changing a Passcode

A passcode is a secondary credential passed to the authentication process when |two-factor authentication is enabled. Passcodes are subject to length/strength requirements imposed by the password policy, but are not bound by other policy mandates regarding history, re-use, and expiration.

The admin account can change passcodes using the ACLI as described below.

Change the user passcode from the # (admin) prompt.

1. Enter secret login passcode at the prompt.

```
ORACLE# secret login passcode
Enter Current Passcode :
```

2. Type the current passcode in response to the Enter Current Passcode : prompt.

```
ORACLE# secret login passcode
Enter Current Passcode :
Enter New Passcode :
```

3. Type the new passcode in response to the Enter New Passcode : prompt.

```
ORACLE# secret login password
Enter Current Passcode :
```

```
Enter New Passcode :
Confirm New Passcode :
```

4. Confirm the new passcode in response to the Confirm New Passcode : prompt.

```
ORACLE# secret login password
Enter Current Passcode :
Enter New Passcode :
Confirm New Passcode :
% Success
ORACLE#
```

Changing the admin Passcode

Change the admin passcode from the # (admin) prompt.

1. Enter secret enable passcode at the prompt.

```
ORACLE# secret enable passcode
Enter Current Passcode :
```

2. Type the current passcode in response to the Enter Current Passcode : prompt.

```
ORACLE# secret enable passcode
Enter Current Passcode :
Enter New Passcode :
```

3. Type the new passcode in response to the Enter New Passcode : prompt.

```
ORACLE# secret enable password
Enter Current Passcode :
Enter New Passcode :
Confirm New Passcode :
```

4. Confirm the new passcode in response to the Confirm New Passcode : prompt.

```
ORACLE# secret enable password
Enter Current Passcode :
Enter New Passcode :
Confirm New Passcode :
% Success
ORACLE#
```

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 OCSBC will not display the reason why the login failed.

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

1. From admin mode, use the following command path to 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 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 user name. 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; while several authentication methods are in use, authentication is most often performed by simple password verification.

Authorization, a process performed after authentication, determines the access or privilege level accorded an authenticated requester. Authorization answers two questions. Does this requester have access to a specific system resource (for example, a file or a configuration object)? If so, what kind of access (for example, create, destroy, or modify)? While there are several authorization methods, authorization is usually accomplished by assigning an authenticated requester to one of a number of pre-defined authorization classes. Conceptually, each class lists available objects, along with an associated object-access type (often expressed as read-only, write-only, or read-write).

Local Authentication and Authorization

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

The feature sets provide two pre-defined user names

- user
- admin

Each of the two user names is associated with an eponymous authorization class which defines the access/privilege level for that user.

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

Console Login

With the addition of the Admin Security feature, local login to the OCSBC is restricted to the two previously described usernames (user and admin) via the console/serial connection. The following table summarizes default authentication and authorization for local logins.

User Name	Logins into/prompt	Authentication	Authorization
user	user mode >	authenticated locally by OCSBC via password	authorized locally by OCSBC assigned to user class inherits access/privilege defined by that class
admin	admin mode #	authenticated locally by OCSBC via password	authorized locally by OCSBC assigned to admin class inherits access/privilege defined by that class

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 OCSBC reverts to its default behavior by providing serial port access.

To turn off access to the serial port:

1. 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

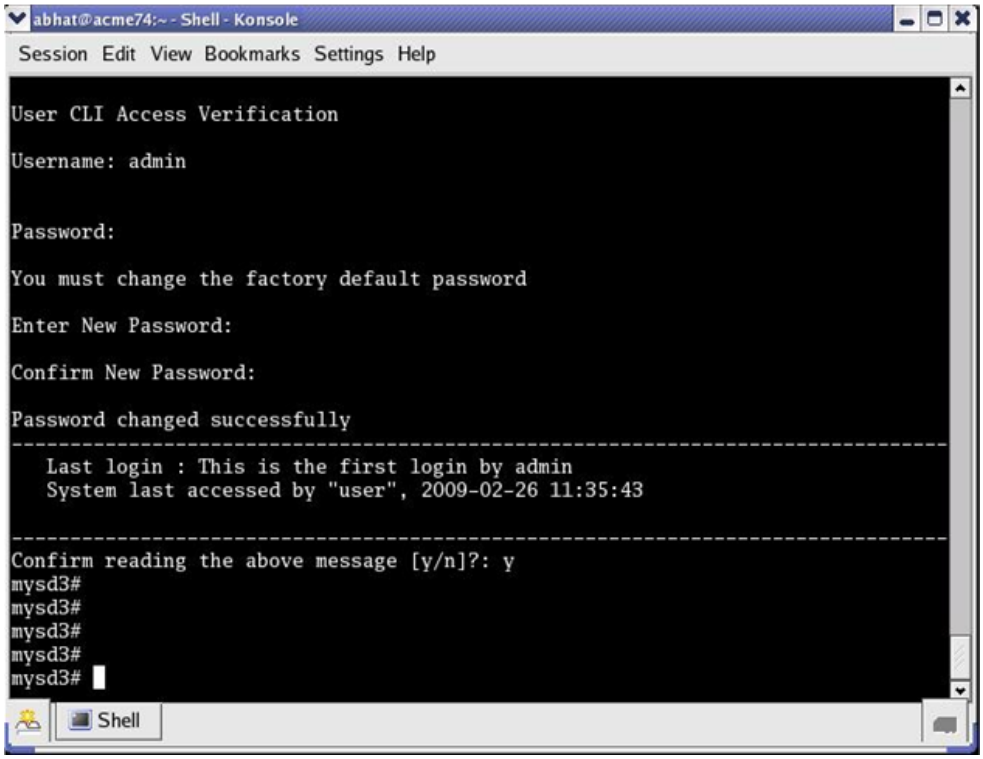
Upon initial login user and admin are required to change the respective password. Initial login is completed only after password change and acknowledgment of the login banner.

The following figure shows the initial login screen for the admin role (the user role views a nearly identical screen).

To complete initial login:

1. Enter one of the recognized user name (user or admin) in response to the **Username:** prompt.
2. Enter the factory default password in response to the **Password:** prompt.

   The factory default user password is acme; the factory default admin password is packet.



Initial admin Login (Console Access)

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

   Passwords must meet the following length/strength requirements.

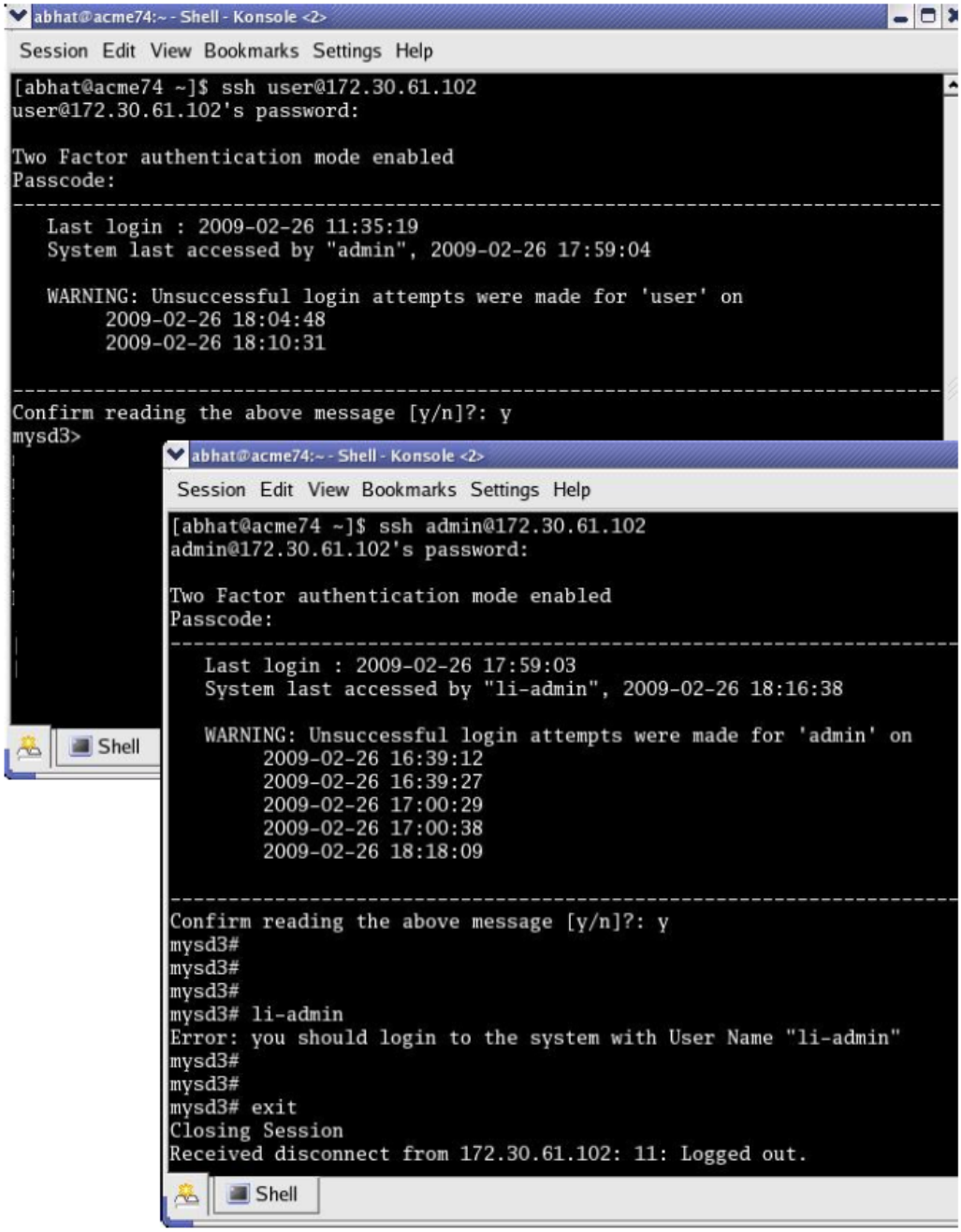
   - user password must contain at least 9 characters
   - admin password must contain at least 15 characters
   - passwords must contain at least 2 lower case alphabetic characters
   - passwords must contain at least 2 upper case alphabetic characters
   - passwords must contain at least 2 numeric characters
   - passwords must contain at least 2 special characters
   - passwords must differ from the prior password by at least 4 characters
   - 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

With the addition of the Admin Security feature, remote access, via the management interface (also referred to as wancom0), is available using SSH Version 2.

The following figure shows remote SSH access for both user and admin)



Remote SSH Login

The following table summarizes default authentication and authorization for remote SSH logins.

User Name	Logins into/prompt	Authentication	Authorization
user	user mode >	authenticated locally by OCSBC via password	authorized locally by OCSBC assigned to user class inherits access/privilege defined by that class

User Name	Logins into/prompt	Authentication	Authorization
admin	admin mode #	authenticated locally by OCSBC via password	authorized locally by OCSBC assigned to admin class inherits access/privilege defined by that class

Remote SSH Login with Public Key

The previous section described password-based SSH authentication. Alternatively, with the addition of the Admin Security feature, you can authenticate using SSH public keys.

Prior to using SSH-public-key-based authentication you must 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 OCSBC performs authentication.

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

Importing a public key requires access to the device on which the public key was generated, or on which it is currently stored with its associated private key. Access is generally attained with a terminal emulation program such as PuTTY, SecureCRT, or TeraTerm.

1. Use a terminal emulation program to access the system from which the public key will be obtained.
2. Copy the base64 encoded public key making sure in include the Begin and End markers as specified by RFC 4716, *The Secure Shell (SSH) Public Key File Format*.
3. Use the **ssh-pub-key** command to import the public key to the OCSBC.

   For importing a public key which will be used to authorize a user, this command takes the format:

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

   - where name is an alias or handle assigned to the imported public key, often the user's name.
   - where authorizationClass designates the authorization class assigned to this user, and takes the value user (the default) or admin.

   To import a public key for Dwight who will be authorized for user privileges, use the following command

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

   To import a public key for Matilda who will be authorized for admin privileges, use the following command

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

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

4. Paste the public key with the bracketing Begin and End markers at the cursor point.

5. Enter a semi-colon (;) to signal the end of the imported host key.
6. Follow directions to save and activate the configuration.

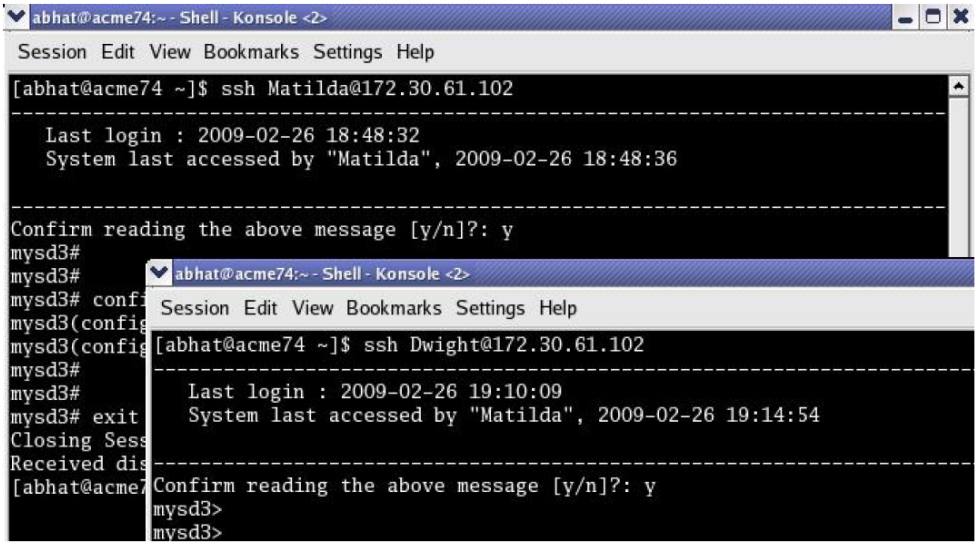
   The entire import sequence is shown below.

```
ORACLE# ssh-pub-key import authorized-key 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 abhat@acme74"
AAAAB3NzaC1yc2EAAAABIwAAAIEAxcYTV595VqdHy12P+mIZBlpeOZx9sX/mSAFihDJYdL
qJIWdiZuSmny8HZIxTIC6na62iD25mlEdyLhlYOuknkYBCU7UsLwmx4dLDyHTbrQHz3b1q
3Tb8auz97/J1p4pw39PT42CoRODzPBrXJV+OglNE/83C1y0SSJ8BjC9LEwE=
---- 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# save-config
checking configuration
------------------------------------------------------------------------
...
...
...
------------------------------------------------------------------------
Save-Config received, processing.
waiting for request to finish
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'.
ORACLE# activate-config
Activate-Config received, processing.
waiting for request to finish
SD is not QOS-capable
Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete
ORACLE#
```

7. 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 SSH-public-key based authentication of Matilda, who has logged in with admin privileges, and Dwight who has logged in with user privileges.



Note in the figure above 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, Dwight is a user instance, and Matilda is a admin instance.

The following table summarizes default authentication and authorization for remote SSH logins.

User Name	Logins into/prompt	Authentication	Authorization
not relevant	user mode > or admin mode #	authenticated locally by OCSBC via SSH public key	authorized locally by OCSBC authorization determined by authorizationClass command argument (user or admin) inherits access/privilege defined by the specified class

Two-Factor Authentication

Two-factor authentication provides an extra level of security for the Oracle Communications Session Border Controller (OCSBC) by requiring users to enter a Passcode during login, in addition to their Username and Password credentials. Two-factor authentication applies to the Super User for both local and SSH login to the ACLI, and for HTTPS login to the Web GUI.

The two-factor authentication option requires the Admin Security feature be provisioned, and you must enable the option by setting `login-auth-method` to "two-factor" and saving the configuration. After you set "two-factor" and save the configuration, the OCSBC prompts you to set the Passcode.

The following illustration shows the configuration workflow on the ACLI.

```
SBC(configure)# security
SBC(security)# admin-security
SBC(admin-security)# login-config
SBC(login-config)# login-auth-method two-factor
SBC# save-config
Checking configuration.
**********************************************************
Admin passcode has not been set. Please set passcode now.
```

```
*********************************************************
Enter New Passcode:
Confirm New Passcode:
Save-Config received, processing.
Waiting for request to finish.
Request to Save-Config has finished.
Save complete.
```

The following illustration shows the user login experience on the ACLI after you enable two-factor authentication.

```
Username: ABCDEF
Password: *****
Two Factor authentication mode enabled
Passcode:

------------------------------------------------------------------------------
Last login : 2017-03-28 11:07:27
System last accessed by "admin", 2017-03-28 11:07:36
WARNING: Unsuccessful login attempts were made for 'admin'
on 2017-03-28 11:12:24
------------------------------------------------------------------------------
Confirm reading the above message [y/n]?: y
```

Passcodes must conform to the length and strength requirements specified in "Enable Two-Factor Authentication."

When you want to change the Passcode in the future, use the **secret** command that you also use for changing the Username and Password.

You can enable two-factor authentication only from the ACLI.

Two-factor authentication does not support RADIUS, TACACS, and HTTP.

Enable Two-Factor Authentication

To enable two-factor authentication for local or SSH login, you must set two-factor as the login authentication method and set the Passcode.

1. Import the local certificate and the local certificate CA into the OCSBC.
2. Configure the Web server for HTTPS.
3. Provision the Admin Security feature.

The passcode must meet the following length and strength requirements:

- Must contain only upper and lower case alphabetical letters, numbers, and punctuation characters
- Must contain a minimum of fifteen characters
- Must contain two lower-case alphabetical letters
- Must contain two upper-case alphabetical letters
- Must contain two numerals
- Must contain two special characters
- Cannot contain, repeat, or reverse the user name
- Can not contain three of the same characters used consecutively

- Must differ from the previous passcode by at least four characters
- Must differ from the last three previous passcodes
- Cannot change more than once every 24 hours

1. Access the **login-config** configuration element.

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

2. Press Enter.

   The system displays the ORACLE (login-config) prompt.

3. Type **# login-auth-method two-factor**.
4. Save the configuration.

   The system prompts you to set the passcode.

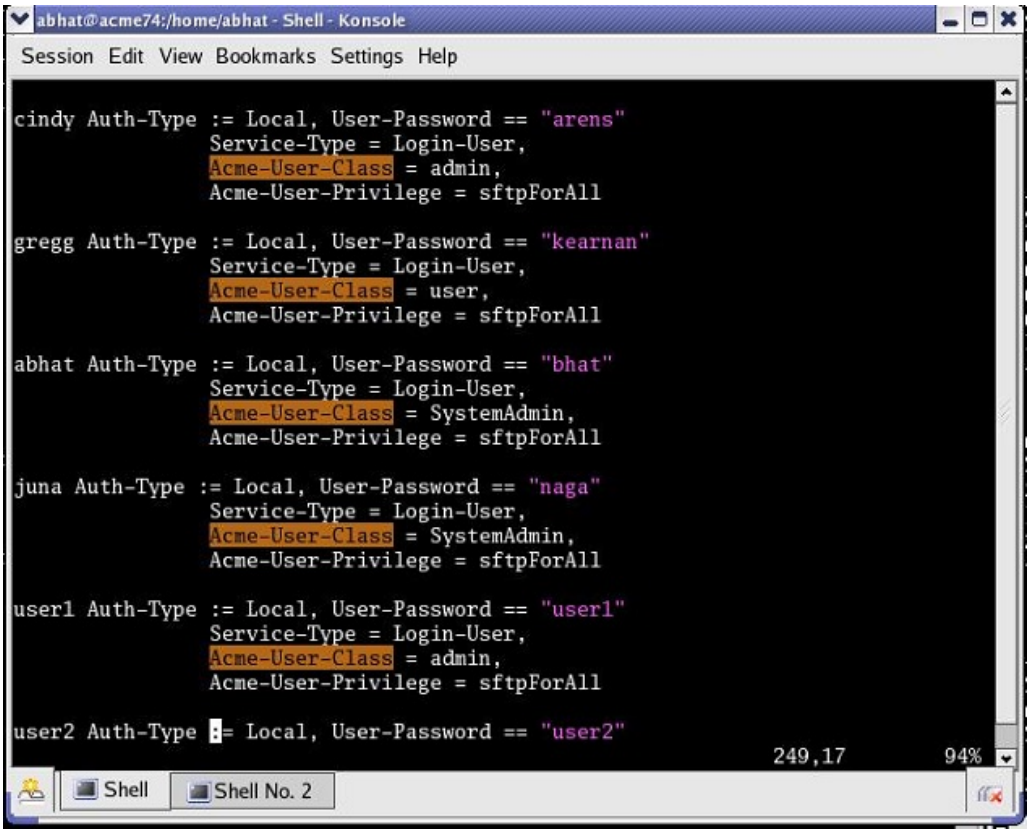
5. Enter the passcode.
6. Confirm the passcode.
7. Type **done** to save the configuration.

RADIUS Authentication and Authorization

As an alternative to the local authentication/authorization described in previous sections, users may prefer to use a RADIUS server or server group for authentication and authorization.

For information on configuring between RADIUS servers and the OCSBC refer to RADIUS Authentication 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.



RADIUS Users File

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

Successful authentication generates a Access Accept message to the OCSBC; 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 OCSBC functions as an enforcement point.

RADIUS Authorization Classes

The RADIUS authorization classes, as specified by the Acme-User-Class VSA, do not coincide directly with those used to authorize the two pre-defined local usernames (user and admin). The RADIUS authorization classes are as follows:

user (RADIUS Acme-User-Class = user)

- provides read-only for all system configuration (including cryptographic keys and certificates)
- The login prompt for this user is ORACLE>

SystemAdmin (RADIUS Acme-User-Class = SystemAdmin)

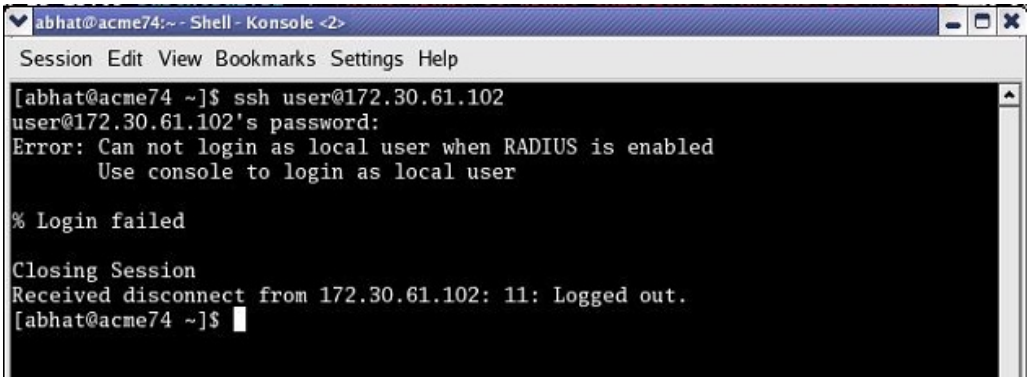
- provides read-write access for system configuration (not including cryptographic keys and certificates)
- The login prompt for this user is ORACLE$

Admin (RADIUS Acme-User-Class = admin)

- provides read-write access for all system configuration (including cryptographic keys and certificates.
- The login prompt for this user is ORACLE#

RADIUS and SSH

When logging in via SSH and authenticating with RADIUS, username/password authentication for the two pre-defined user names (user, admin) is disabled. Attempts to login via SSH are rejected as shown in the following figure.



Local User Login with SSH (RADIUS Enabled)

If you want to enable user and admin access via SSH with RADIUS configured, you must explicitly define users on the RADIUS server with appropriate Acme-User-Class.

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 OCSBC supports TACACS+ in both Admin Security mode and JITC. The OCSBC 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 OCSBC, the OCSBC 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 OCSBC queries the TACACS+ server and generates start/stop accounting records using TACACS+ settings.

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 OCSBC, 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 OCSBC 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-bitAES, 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 crypto-graphic 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 OCSBC: 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)# admin-security
   ORACLE(admin-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)
   ```

3. **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(admin-security)#
   ```

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

Managing SSH Keys

Use the following procedure to import an SSH host key.

Importing a host key requires access to the SFTP server or servers which receive audit log transfers. Access is generally most easily accomplished with a terminal emulation program such as PuTTY, SecureCRT, or TeraTerm.

1. Use a terminal emulation program to access the SSH file system on a configured SFTP server.
2. Copy the server's base64 encoded public file making sure in include the Begin and End markers as specified by RFC 4716, *The Secure Shell (SSH) Public Key File Format*.

   For OpenSSH implementations host files are generally found at `/etc/ssh/ssh_host_dsa_key.pub`, or `etc/ssh/sss_host_rsa.pub`. Other SSH implementations can differ.
3. From admin mode use the **ssh-pub-key** command to import the host key to the OCSBC.

   For importing a host key, this command takes the format:

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

where name is an alias or handle assigned to the imported host key, generally the server name or a description of the server function.

```
ORACLE# ssh-pub-key import known-host fedallah

IMPORTANT:
    Please paste ssh public key in the format defined in rfc4716.
    Terminate the key with ";" to exit.......
```

4. Paste the public key with the bracketing Begin and End markers at the cursor point.
5. Enter a semi-colon (;) to signal the end of the imported host key.
6. Follow directions to save and activate the configuration.

The entire import sequence is shown below.

```
ORACLE# ssh-pub-key import known-host fedallah

IMPORTANT:
 Please paste ssh public key in the format defined in rfc4716.
 Terminate the key with ";" to exit.......

---- BEGIN SSH2 PUBLIC KEY ----
Comment: "2048-bit RSA, converted from OpenSSH by klee@acme54"
AAAAB3NzaC1yc2EAAAABIwAAAQEA7OBf08jJe7MSMgerjDTgZpbPblrX4n17LQJgPC7clL
cDGEtKSiVt5MjcSav3v6AEN2pYZihOxd2Zzismpoo019kkJ56s/IjGstEzqXMKHKUr9mBV
qvqIEOTqbowEi5sz2AP31GUjQTCKZRF1XOQx8A44vHZCum93/jfNRsnWQ1mhHmaZMmT2LS
hOr4J/Nlp+vpsvpdrolV6Ftz5eiVfgocxrDrjNcVtsAMyLBpDdL6e9XebQzGSS92TPuKP/
yqzLJ2G5NVFhxdw5i+FvdHz1vBdvB505y2QPj/iz1u3TA/3O7tyntBOb7beDyIrg64Azc8
G7E3AGiH49LnBtlQf/aw==
---- 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# save-config
checking configuration
------------------------------------------------------------------------
...
...
...
------------------------------------------------------------------------
Save-Config received, processing.
waiting for request to finish
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'.
ORACLE# activate-config
Activate-Config received, processing.
waiting for request to finish
SD is not QOS-capable
Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete
ORACLE#
```

Importing SSH Keys

Use the following procedure to import an SSH public key.

Prior to using SSH-public-key-based authentication you must import a copy 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. Upon receiving the offered public key, the SBC validates it against the previously obtained trusted copy of the key to identify and authenticate the user.

Importing a public key requires access to the device on which the public key was generated, or on which it is currently stored with its associated private key. Access is generally attained with a terminal emulation program such as PuTTY, SecureCRT, or TeraTerm.

1. Use a terminal emulation program to access the system from which the public key will be obtained.
2. Copy the base64 encoded public key making sure to include the Begin and End markers as specified by RFC 4716, *The Secure Shell (SSH) Public Key File Format*.
3. From admin mode use the **ssh-pub-key** command to import the public key to the OCSBC.

   For importing a public key which will be used to authorize a user, this command takes the format:

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

   - where name is an alias or handle assigned to the imported public key, often the user's name.
   - where authorizationClass optionally designates the authorization class assigned to this user, and takes the value user (the default) or admin.

   To import a public key for Matilda who will be authorized for admin privileges, use the following command

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

   IMPORTANT:
     Please paste ssh public key in the format defined in rfc4716.
     Terminate the key with ";" to exit.......
   ```
4. Paste the public key with the bracketing Begin and End markers at the cursor point.
5. Enter a semi-colon (;) to signal the end of the imported host key.
6. Follow directions to save and activate the configuration.

   The entire import sequence is shown below.

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

   IMPORTANT:
     Please paste ssh public key in the format defined in rfc4716.
     Terminate the key with ";" to exit.......

   ---- BEGIN SSH2 PUBLIC KEY ----
   Comment: "1024-bit RSA, converted from OpenSSH by abhat@acme74"
   AAAAB3NzaC1yc2EAAAABIwAAAIEAxcYTV595VqdHy12P+mIZBlpeOZx9sX/mSAFihDJYdL
   qJIWdiZuSmny8HZIxTIC6na62iD25mlEdyLhlYOuknkYBCU7UsLwmx4dLDyHTbrQHz3b1q
   3Tb8auz97/J1p4pw39PT42CoRODzPBrXJV+OglNE/83C1y0SSJ8BjC9LEwE=
   ```

```
---- 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# save-config
checking configuration
------------------------------------------------------------------------
...
...
...
------------------------------------------------------------------------
Save-Config received, processing.
waiting for request to finish
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'.
ORACLE# activate-config
Activate-Config received, processing.
waiting for request to finish
SD is not QOS-capable
Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete
ORACLE#
```

Generating an SSH Key Pair

Use the following procedure to generate an SSH key pair.

The initial step in generating an SSH key pair is to configure a public key record which will serve as a container for the generated key pair.

1. Navigate to the **public-key** configuration element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# security
   ORACLE(security)# public-key
   ORACLE(public-key)#
   ```

2. Use the **name** command to provide the object name, and the **show** command to verify object creation.

   ```
   ORACLE(public-key)# name tashtego
   ORACLE(public-key)# show public-key
    name                  tashtego
    type                  rsa
    size                  1024
    last-modified-by
    last-modified-date

   ORACLE(public-key)#
   ```

   creates a public key record named tashtego.

3. Use the **done** command to complete object creation.

   ```
   ORACLE(public-key)# done
   public-key
   name                  tashtego
   type                  rsa
   size                  1024
   last-modified-by      admin@console
   ```

```
last-modified-date     2009-03-06 11:18:00
ORACLE(public-key)#
```

4. Make a note of the **last-modified-date** time value.
5. Move back to admin mode, and save and activate the configuration.

```
ORACLE(public-key)# exit
ORACLE(security)# exit
ORACLE(configure)# exit
ORACLE#
ORACLE# save-config
...
...
...
ORACLE# activate-config
...
...
...
ORACLE#
```

6. Now use the **ssh-pub-key generate** command, in conjunction with the name of the public key record created in Step 3, to generate an SSH key pair.

   For importing an SSH key pair, this command takes the format:

   ```
   ssh-pub-key generate <name>
   ```

   where name is an alias or handle assigned to the generated key pair, generally the client name or a description of the client function.

   ```
   ORACLE# ssh-pub-key generate tashtego
   Please wait...
   public-key 'tashtego' (RFC 4716/SECSH format):
   ---- BEGIN SSH2 PUBLIC KEY ----
   Comment: "1024-bit rsa"
   AAAAB3NzaC1yc2EAAAABIwAAAIEArZEP1/WiYsdGd/Pi8V6pnSwV4cVG4U+jVOwiSwNJCC9Nk82/
   FKYleLZevy9D3lrZ8ytvu+sCYy0fNk4nwvz20c2N+r86kDru88JkUqpelJDx1AR718Icpr7ZaAx2L
   +e7cpyRSXCgbQR7rXu2H3bp9Jc0VhR2fmkclmrGAIr7Gnc=
   ---- END SSH2 PUBLIC KEY ----
   SSH public-key pair generated successfully....
   WARNING: Configuration changed, run "save-config" command to save
                   it and run "activate-config" to activate the changes
   ORACLE#
   ```

7. Copy the base64-encoded public key. Copy only the actual public key — do not copy the bracketing Begin and End markers nor any comments. Shortly you will paste the public key to one or more SFTP servers.
8. Save and activate the configuration.

   ```
   ORACLE# save-config
   ...
   ...
   ...
   ORACLE# activate-config
   ...
   ...
   ...
   ```

9. Return to the public-key configuration object, and select the target public key record instance.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# security
   ```

```
ORACLE(security)# public-key
ORACLE(public-key)# sel
<name>:
1: acme01
2: acme02
3: tashtego

selection: 3
ORACLE(public-key)# show
public-key
      name                    tashtego
      type                    rsa
      size                    1024
      last-modified-by        admin@console
      last-modified-date      2009-03-06 11:24:32
ORACLE(public-key)#
```

10. Verify that the record has been updated to reflect key generation by examining the value of the last-modified-date field.

Copying Public Key to SFTP Server

Use the following procedure to copy a client public key to an SFTP server.

Copying the client public key to an SFTP server requires server access generally using a terminal emulation program such as PuTTY, SecureCRT, or TeraTerm.

1. Use a terminal emulation program to access the SSH file system on a configured SFTP server.
2. Copy the client key to the SFTP server.

   On OpenSSH implementations, public keys are usually stored in the `~/.ssh/authorized_keys file`. Each line this file (1) is empty, (2) starts with a pound (#) character (indicating a comment), or (3) contains a single public key.

   Refer to the sshd man pages for additional information regarding file format.

   Use a text editor such as vi or emacs to open the file and paste the public key to the tail of the authorized_keys file.

   For SSH implementations other than OpenSSH, consult the system administrator for file structure details.

   Use the following procedure to view an imported SSH key.

   You can use the show security ssh-pub-key command to display information about SSH keys imported to the OCSBC with the ssh-pub-key command; you cannot display information about keys generated by the ssh-pub-key command.

```
ORACLE# show security ssh-pub-key brief
login-name:
    acme74
finger-print:
    51:2f:f1:dd:79:9e:64:85:6f:22:3d:fe:99:1f:c8:21
finger-print-raw:
    0a:ba:d8:ef:bb:b4:41:d0:dd:42:b0:6f:6b:50:97:31
login-name:
    fedallah
finger-print:
    c4:a0:eb:79:5b:19:01:f1:9c:50:b3:6a:6a:7c:63:d5
finger-print-raw:
```

```
    ac:27:58:14:a9:7e:83:fd:61:c0:5c:c8:ef:78:e0:9c
ORACLE#
```

displays summary information for all SSH imported keys

- login-name—contains the name assigned to the RSA or DSA public key when it was first imported
- finger-print—contains the output of an MD5 hash computed across the base64-encoded public key
- finger-print-raw—contains the output of an MD5 hash computed across the binary form of the public key

```
ORACLE# show security ssh-pub-key brief fedallah
login-name:
     fedallah
finger-print:
     c4:a0:eb:79:5b:19:01:f1:9c:50:b3:6a:6a:7c:63:d5
finger-print-raw:
     ac:27:58:14:a9:7e:83:fd:61:c0:5c:c8:ef:78:e0:9c
ORACLE#
```

displays summary information for a specific SSH public key (in this case fedallah)

```
ORACLE# show security ssh-pub-key detail fedallah
host-name:
     fedallah
comment:
     "2048-bit RSA, converted from OpenSSH by klee@acme54"
finger-print:
     c4:a0:eb:79:5b:19:01:f1:9c:50:b3:6a:6a:7c:63:d5
finger-print-raw:
     ac:27:58:14:a9:7e:83:fd:61:c0:5c:c8:ef:78:e0:9c
pub-key:

AAAAB3NzaC1yc2EAAAABIwAAAQEA7OBf08jJe7MSMgerjDTgZpbPblrX4n17LQJgPC7clLcDGEtKS
iVt5MjcSav3v6AEN2pYZihOxd2Zzismpoo019kkJ56s/
IjGstEzqXMKHKUr9mBVqvqIEOTqbowEi5sz2AP31GUjQTCKZRF1XOQx8A44vHZCum93/
jfNRsnWQ1mhHmaZMmT2LShOr4J/Nlp
+vpsvpdrolV6Ftz5eiVfgocxrDrjNcVtsAMyLBpDdL6e9XebQzGSS92TPuKP/
yqzLJ2G5NVFhxdw5i+FvdHz1vBdvB505y2QPj/iz1u3TA/
3O7tyntBOb7beDyIrg64Azc8G7E3AGiH49LnBtlQf/aw==

modulus: (256)
ECE05FD3C8C97BB3123207AB8C34E06696CF6E5AD7E27D7B2D02603C2EDC94B703184B4A4A256
DE4C8DC49ABF7BFA004376A5866284EC5DD99CE2B26A68A34D7D924279EACFC88C6B2D133A973
0A1CA52BF66055AAFA8810E4EA6E8C048B9B33D803F7D4652341308A6511755CE431F00E38BC7
642BA6F77FE37CD46C9D64359A11E66993264F62D284EAF827F365A7EBE9B2FA5DAE8955E85B7
3E5E8957E0A1CC6B0EB8CD715B6C00CC8B0690DD2FA7BD5DE6D0CC6492F764CFB8A3FFCAACCB2
761B9355161C5DC398BE16F747CF5BC176F079D39CB640F8FF8B3D6EDD303FDCEEEDCA7B4139B
EDB783C88AE0EB803373C1BB137006887E3D2E706D9507FF6B
exponent: (1)
23

ORACLE#
```

displays detailed information for specific SSH public key (in this case fedallah, an RSA key)

- host-name—contains the name assigned to the RSA key when it was first imported

- finger-print—contains the output of an MD5 hash computed across the base64-encoded RSA public key
- finger-print-raw—contains the output of an MD5 hash computed across the binary form of the RSA public key
- public key—contains the base64-encoded RSA key
- modulus—contains the hexadecimal modulus (256) of the RSA key
- exponent—(also known as public exponent or encryption exponent) contains an integer value that is used during the RSA key generation algorithm. Commonly used values are 17 and 65537. A prime exponent greater than 2 is generally used for more efficient key generation.

```
ORACLE# show security ssh-pub-key detail acme74
host-name:
     acme74
comment:
     DSA Public Key
finger-print:
     51:2f:f1:dd:79:9e:64:85:6f:22:3d:fe:99:1f:c8:21
finger-print-raw:
     0a:ba:d8:ef:bb:b4:41:d0:dd:42:b0:6f:6b:50:97:31
pub-key:

AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbETW6ToHv8D1
UJ/z+zHo9Fiko5XybZnDIaBDHtblQ
+Yp7StxyltHnXF1YLfKD1G4T6JYrdHYI14Om1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Y
isk/gF
+1VAAAAFQDb8D5cvwHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pfd
tW9vGfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/
FAAvioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACBAN7CY
+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO
+JsvphVMBJc9HSn24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z
7dLM5sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV

p: (128)
F63C64E1D8DB2152240E97602F47470347C5A7A1BF1E70389D2BCD9773A12397C5B1135BA4E81
EFF03D5427FCFECC7A3D162928E57C9B6670C86810C7B5B950F98A7B4ADC7296D1E75C5D582DF
283D46E13E8962B747608D783A6D5E83D7B836709195E6AAA193C5DD419F6626BA6D7AC64D07F
7809AB67BB622B24FE017ED55
q: (20)
DBF03E5CBF01D64D90CF7D7D03DACF5177B341BD
g: (128)
94DF76F816FB0F828B624DC8C116D76E5C177643E0800E297DDB56F6F19F274FD11DDF8D8C1E1
EA350FED1D8B1EAD5F060637B3CA4B947F1573CDC311CF6A9723F6E2F5267D80590D9DB249DFF
A2FC5000BE2A143E499D31CD33B96A12384B12361543B57DD676F55C19C06AF5C7ADCEBB4E296
3A8709989F34A9A7714D11ED5
pub_key: (128)
DEC263E28ABF5807A51CC5C1D426EC72BD6DBD4B028D8AC1AA179DA74581EA6D34141E4971B5B
CEF89B2FA6154C04973D1D29F6E1562D62DB0CBBBE2A5EF8988F3895B9C58A8E32846F5D63BAA
9C5D060E50775559B11CB9B19C0CFAE3758AE3667B74B339B18DBDA2E7B3BF85F3D8FB8C721E5
518F3FE083AB308CE25A16815

ORACLE#
```

displays detailed information for specific SSH public key (in this case acme74, a DSA key)

- host name—contains the name assigned to the DSA public key when it was first imported
- comment—contains any comments associated with the DSA key

- finger-print—contains the output of an MD5 hash computed across the base64-encoded DSA public key
- finger-print-raw—contains the output of an MD5 hash computed across the binary form of the DSA public key
- public key—contains the base64 encoded DSA key
- p—contains the first of two prime numbers used for key generation
- q—contains the second of two prime numbers used for key generation
- g—contains an integer that together with p and q are the inputs to the DSA key generation algorithm

```
ORACLE# show security ssh-pub-key detail
...
...
...
ORACLE#
```

displays detailed information for all SSH imported keys.

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 directolls [ls-options [path]Display local directory listing of either path or current directory if path is not specified.
- lmkdir 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

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 all 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.

Using the Oracle Rescue Account for PNF Zeroization

To enable the Oracle Rescue Account:

1. Connect to the OCSBC's serial console.
2. Reboot the OCSBC 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.

2
Audit Log

Overview

The audit log records creation, modification, and deletion of all user-accessible configuration elements, access to critical security data such as public keys. For each logged event it provides associated user-id, date, time, event type, and success/failure data for each event. As a result, the log supports after the fact investigation of loss or impropriety, and appropriate management response. Only admin-level users have audit log access. These users can retrieve, read, copy, and upload the audit log. The original log cannot be deleted or edited by any operator action.

The audit log is transferred to a previously configured SFTP server or servers when one of three specified conditions is satisfied.

1. A configurable amount of time has elapsed since the last transfer.
2. The size of the audit log (measured in Megabytes) has reached a configured threshold.
3. The size of the audit log has reached a configured percentage of the allocated storage space.

   Transfer is targeted to a designated directory of each SFTP target server. The audit log file is stored on the target SFTP server or servers with a filename that takes the format:

   audit<timestamp>

   where <timestamp> is a 12-digit string that takes the format YYYYMMDDHHMM.

   audit200903051630

   names an audit log file transferred to an SFTP server on March 5, 2009 at 4:30 PM.

Audit Log Format

Audit log events are comma-separated-values (CSV) lists that have the following format:

```
{TimeStamp,user-id@address:port,Category,EventType,Result,Resource,Details,...}
```

```
{2009-0305 15:19:27,sftp-
elvis@192.2.0.10:22,security,login,success,authentication,,.}
```

TimeStamp specifies the time that the event was written to the log

Category takes the values: security | configuration | system

EventType takes the values: create | modify | delete | login | logout | data-access | save-config | reboot | acquire-config

Result takes the values: successful | unsuccessful

Resource identifies the configuration element accessed by the user

Details (which is displayed only in verbose mode) provides fine-grained configuration details

- If EventType = create, details is “New = element added”
- If EventType = modify, details is “Previous = oldValue New = newValue”
- If EventType = delete, details is “Element = deleted element”
- If EventType = data-access, details is “Element = accessed element”

The following lists and describes the actions that generate audit log events.

- Login—Every login attempt

  ```
  2009-03-05 17:31:14,sftp-elvis@192.2.0.10:22,security,login,
  success,authentication,,.
  ```

- Logout—Every logout attempt

  ```
  2009-03-05 18:44:03,sftp-
  elvis@192.2.0.10:22,security,logout,success,authentication,,.
  ```

- save-config—Every save-config CLI command

  ```
  2009-03-05 15:45:29,acliConsole-admin@console,configuration,
  save-config,success,CfgVersion=111,,.
  ```

- activate-config—Every activate-config CLI command

  ```
  2009-03-05 15:45:36,acliConsole-admin@console,configuration,activate-
  config,success,RunVersion=111,,.
  ```

- DataAccess
  - a) attempt to retrieve data using SFTP
  - b) attempt to export using ssh-pub-key export
  - c) attempt to display security info using show security
  - d) attempt to kill a session using kill

  ```
  2009-03-05 15:25:59,sftp-elvis@192.2.0.10:22,security,data-access,
  success,code/auditaudit200903051518,,.
  ```

- Create
  - a) any action that creates a configuration property
  - b) any action that creates a file

  ```
  2009-03-05 15:45:01,acliConsole-admin@console,configuration,create,
  success,public-key,
  Element=
  <?xml version='1.0' standalone='yes'?>
  <sshPubKeyRecord
    name='dummy'
    comment=''
    keyType='2'
    encrType='1'
    keySize='1024'
    pubKey=''
    privKey=''
    fingerPrint=''
    fingerPrintRaw=''
    lastModifiedBy='acmin@console'
    lastModifiedDate='2009-03-05 15:45:01>
  </sshPubKeyRecord
  ```

- Modify

- a) any action that modifies a configuration property

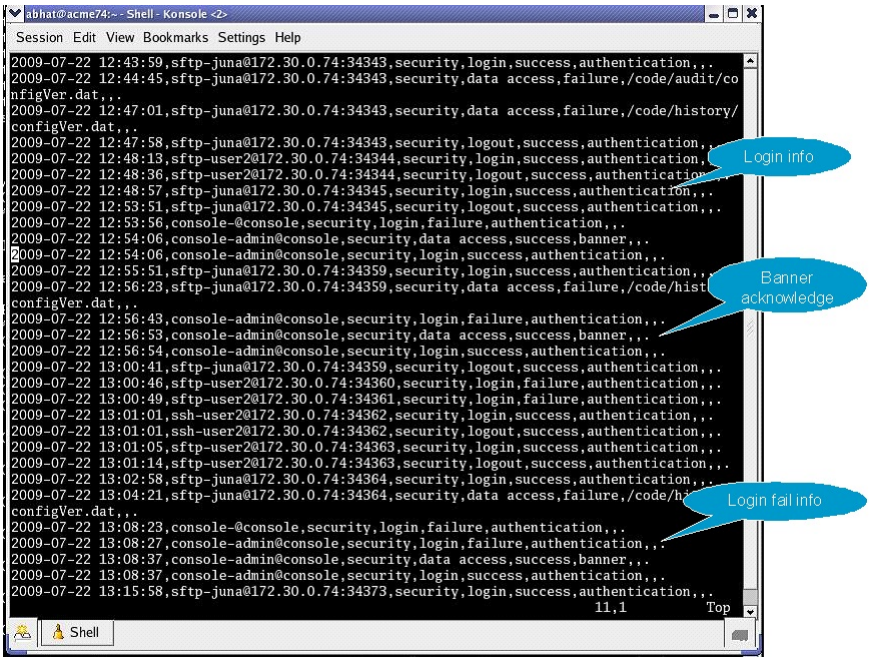
```
2009-03-05 15:48:01,acliConsole-admin@console,configuration,modify,
success,public-key,
Previous=
<?xml version='1.0' standalone='yes'?>
<sshPubKeyRecord
  name='dummy'
  comment=''
  keyType='2'
  encrType='1'
  keySize='1024'
  pubKey=''
  privKey=''
  fingerPrint=''
  fingerPrintRaw=''
  lastModifiedBy='acmin@console'
  lastModifiedDate='2009-03-05 15:45:01>
</sshPubKeyRecord

New=
<?xml version='1.0' standalone='yes'?>
<sshPubKeyRecord
  name='dummy'
  comment=''
  keyType='2'
  encrType='2'
  keySize='1024'
  pubKey=''
  privKey=''
  fingerPrint=''
  fingerPrintRaw=''
  lastModifiedBy='acmin@console'
  lastModifiedDate='2009-03-05 15:48:01>
</sshPubKeyRecord
```

- Delete
  - a) any action that deletes a configuration property
  - b) any action that deletes a file

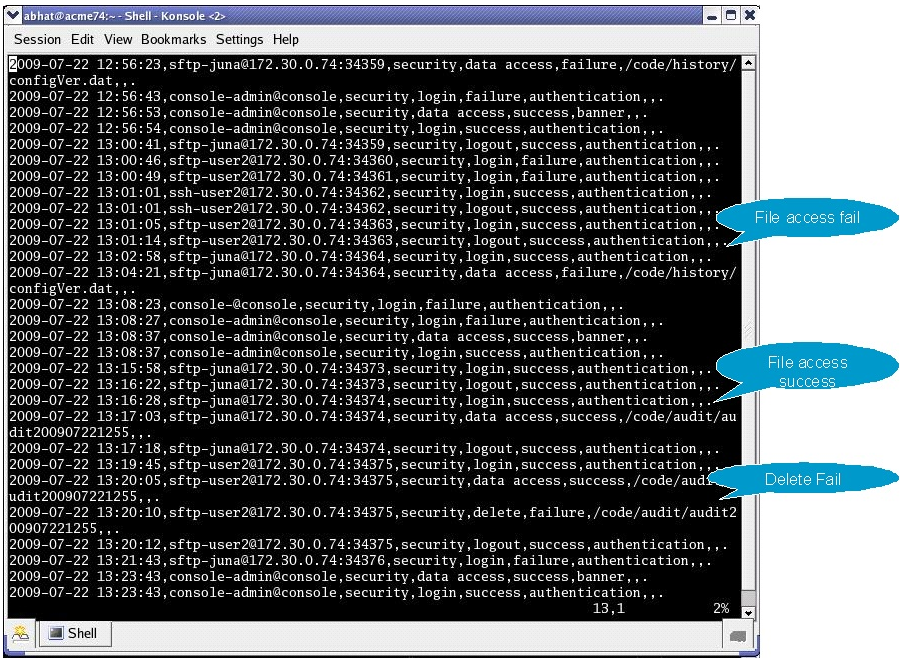
```
2009-03-05 15:51:39,acliConsole-admin@console,configuration,delete,
success,public-key,
Element=
<?xml version='1.0' standalone='yes'?>
<sshPubKeyRecord
  name='dummy'
  comment=''
  keyType='2'
  encrType='2'
  keySize='1024'
  pubKey=''
  privKey=''
  fingerPrint=''
  fingerPrintRaw=''
  lastModifiedBy='acmin@console'
  lastModifiedDate='2009-03-05 15:51:39>
</sshPubKeyRecord
```

Audit Log Samples

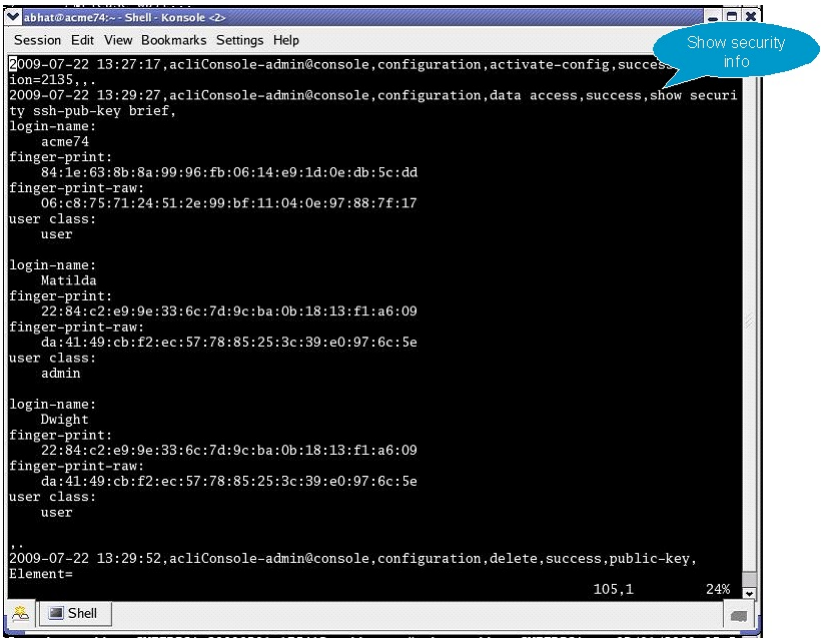
The follow screen captures provide samples of specific audit log entries.



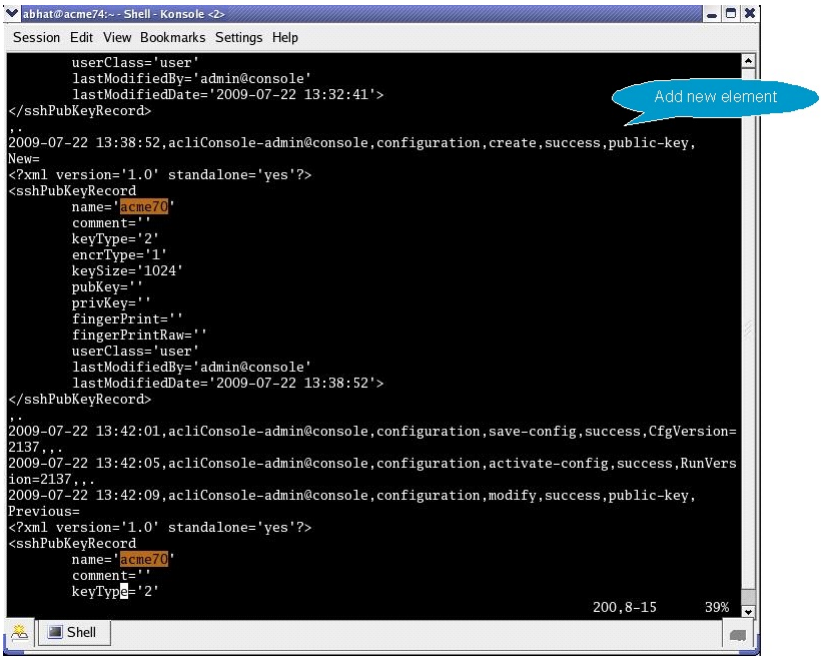
Login Reporting



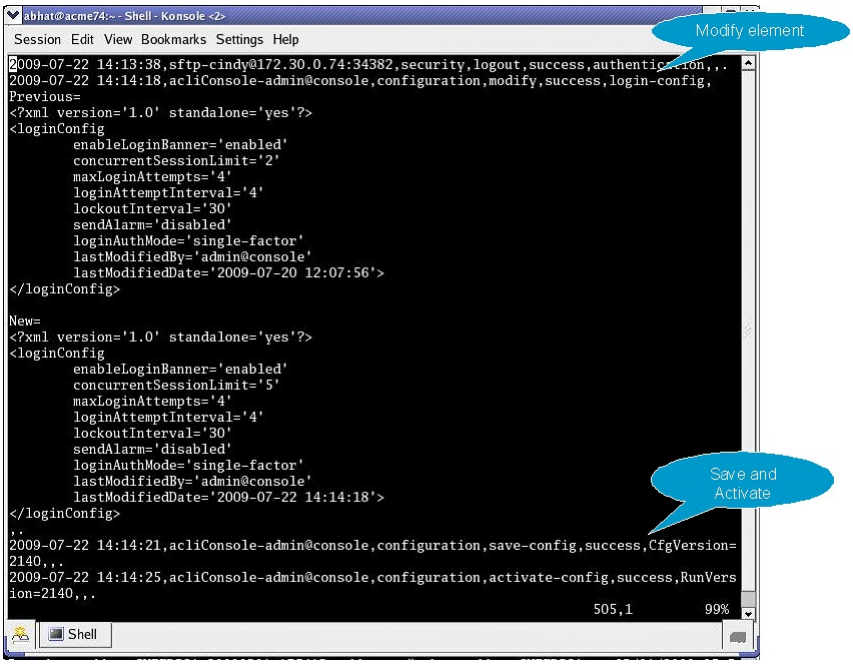
File Access Reporting



show security Reporting



Create Element Reporting



Modify Element/Activate Reporting

Viewing the Audit Log

The audit log can be displayed only after transfer to an SFTP server, either by (1) automatic transfer triggered by a timer, or space-based threshold as previously described; or by (2) manual SFTP transfer accomplished by the admin user.

Configure the Audit Log

The single instance **audit-logging** configuration element enables, sizes, and locates the audit log within the local file structure. It also specifies the conditions that trigger transfer of the log to one or more SFTP servers.

1. Access the audit-logging configuration element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# security
   ORACLE(security)# admin-security
   ORACLE(admin-security)# audit-logging
   ORACLE(audit-logging)#
   ```

2. **admin-state**—Enables or disables the audit log.

   Use enabled to enable the audit log. Retain the default value (disabled) to disable the log.

3. **detail-level**—Specifies the level of detail associated with audit log entries.

   Retain the default value (brief) to write succinct log entries; use verbose to generate more detailed entries.

4. **file-transfer-time**—Specifies the maximum interval (in hours) between audit-log transfers to a previously-configured SFTP server or servers.

   Allowable values are integers within the range 0 through 65535.

The value 0 disables time-based-transfer of the audit log. Consequently, upload to an SFTP server is triggered only by exceeding the percentage-based or absolute-size-based thresholds established by the **percentage-full** and **max-file-size** properties, or by manual SFTP file transfer performed by a properly privileged admin-level user.

Retain the default value (720 hours/30 days), or provide an alternate value to trigger time-based-transfer. With time-based-transfer enabled, automatic upload of the audit file to an SFTP server or servers is triggered when the interval decrements to 0. At that time the audit log is transferred, an alarm alerting the recipient to the transfer is generated, and the timer re-sets to its configured value. Assuming the file transfer succeeds, the audit log is deleted. If the file transfer fails, the audit log is retained until it exceeds the value specified by **max-storage-space**.

> **Note:**
>
> The file-transfer-time interval is reset to its configured value with any audit log transfer regardless of cause.

5. **max-storage-space**—Specifies the maximum disk space (measured in Megabytes) available for audit log storage.

   Allowable values are integers within the range 1 through 32.

   Allocate space for the audit log by retaining the default value, or by selecting a new value from within the allowable range.

6. **percentage-full**—Specifies a file size threshold (expressed as a percentage of max-storage-space) that triggers audit file transfer to a previously-configured SFTP server or servers.

   Allowable values are integers within the range 0 through 99.

   The value 0 disables percentage-based-transfer of the audit log. Consequently, upload to an SFTP server is triggered only by exceeding the time-based and absolute-size-based thresholds established by the **file-transfer-time** and **max-file-size properties**, or by manual SFTP file transfer performed by a properly privileged admin-level user.

   Retain the default value (75 percent), or provide an alternate value to trigger percentage-based-transfer. With percentage-based-transfer enabled, automatic upload of the audit file to an SFTP server or servers is triggered when audit log size exceeds the value **max-storage-spac**e x (**percentage-full**/100). At that time the audit log is transferred, and an alarm alerting the recipient to the transfer is generated. Assuming the file transfer succeeds, the audit log is deleted. If the file transfer fails, the audit log is retained until it exceeds the value specified by **max-storage-space.**

7. **max-file-size**—Specifies a file size threshold (expressed as an absolute file size measured in Megabytes) that triggers audit file transfer to a previously-configured SFTP server or servers.

   Allowable values are integers within the range 0 through 10.

   The value 0 disables absolute-size-based-transfer of the audit log. Consequently, upload to an SFTP server is triggered only by exceeding the time-based and percentage-based thresholds established by the **file-transfer-time** and **percentage-full** properties, or by manual SFTP file transfer performed by a properly privileged admin-level user.

   Retain the default value (5 Megabytes), or provide an alternate value to trigger absolute-size-based-transfer. With absolute-size-based-transfer enabled, automatic upload of the audit file to an SFTP server or servers is triggered when audit log size exceeds the value

max-file-size. At that time the audit log is transferred and an alarm alerting the recipient to the transfer is generated. Assuming the file transfer succeeds, the audit log is deleted. If the file transfer fails, the audit log is retained until it exceeds the value specified by **max-storage-space.**

8. **storage-path**—Specifies the directory that houses the audit log.

   Retain the default value (/code/audit), or identify another local directory.

9. **audit-trail**—Enables logging every command that is processed by the OCSBC.

   Use enabled to enable the audit logging all commands. Retain the default value (disabled) to log only relevant information.

   > **Note:**
   >
   > When enabled, the OCSBC logs only commands that the SBC is able to process. For example, if a command is entered incorrectly, it will not be logged.

10. **audit-record-output**—Indicates how the OCSBC logs audit records.

    - syslog—The OCSBC logs audit records over syslog.
    - file—The OCSBC logs audit records to a file. This is the default value.
    - both—The OCSBC logs audit records over both syslog and to a file.

    A sample audit log configuration appears below:

    ```
    ORACLE(admin-security)# admin-state enabled
    ORACLE(admin-security)# file-transfer-time 1
    ORACLE(admin-security)# percentage-full 0
    ORACLE(audit-logging)# max-file-size 0
    ```

    This configuration allocates 32MB (the default value) for audit logging, which is enabled in brief mode. Audit log transfer to a configured SFTP server or servers occurs on an hourly schedule.; other transfer triggers are disabled.

11. Type **done** to save your configuration.

Configure SFTP Audit Log Transfer

Prior to using SFTP-enabled file transfer you must import a copy of each SFTP server's host key to the OCSBC. The host key identifies the server as a trusted entity when the OCSBC is operating as an SSH or SFTP client.

The SSH protocol requires the server to present its host key to a client during the SSH handshake. The client validates the offered key against the previously obtained trusted copy of the key to identify and authenticate the server.

You must also generate an SSH public and private key pair for the OCSBC in support of its operations as an SSH client. Just as the host key authenticates the SSH server to the SSH client, the generated public key authenticates the SSL client to the SSH server. After generating the SSH key pair, you copy the public key to each configured SFTP server. During the authentication process, the server validates the offered client key against this trusted copy to identify and authenticate the client.

To provide needed keys:

1. Use the procedure described in Importing a Host Key to import the host key of each SFTP server.
2. Use the procedure described in Generating an SSH Key Pair to generate an SSH public and private key.
3. Use the procedure described in Copying a Client Key to an SSH or SFTP Server to copy the public key to the SFTP server.

Configuring SFTP Servers

The multi-instance **push-receiver** configuration element identifies remote SFTP servers that receive audit log transfers.

1. Access the audit-logging configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# admin-security
ORACLE(admin-security)# audit-logging
ORACLE(audit-logging)# push-receiver
ORACLE(push-receiver)#
```

2. Select the **push-receiver** object to edit.

```
ORACLE(push-receiver)# select
<server>:<port>:
1: 192.168.54.55:22 server = 192.168.54.55, port = 22

selection: 1
ORACLE(push-receiver)#
```

3. **server**—in conjunction with port, specifies an SFTP server IP address:port pair

   Provide the IP address of an SFTP server that receives transferred audit logs. For example,

```
ORACLE(push-receiver)# server 192.0.2.100
ORACLE(push-receiver)#
```

4. **port**—in conjunction with server, specifies an SFTP server IP address:port pair

   Provide the port number monitored by server for incoming audit log transfers. This parameter defaults to port 22, the well-known Secure Shell (SSH) port. Retain the default value, or identify the monitored port with an integer within the range from 1 through 65535.

```
ORACLE(push-receiver)# port 2222
ORACLE(push-receiver)#
```

5. **remote-path**—specifies the absolute file path to the remote directory that stores transferred audit log file

   Provide the file path to the remote directory. For example,

```
ORACLE(push-receiver)# remote-path /home/acme/auditLogs
ORACLE(push-receiver)#
```

6. **filename-prefix**—specifies an optional prefix that can be appended to the audit log file name when transferred to an SFTP server

   Provides an optional prefix which is appended to the audit log filename. For example,

```
ORACLE(push-receiver)# filename-prefix auvik
ORACLE(push-receiver)#
```

7. **auth-type**—specifies the authentication type required by this remote SFTP server

   Two authentication types are supported — simple password, or public keys.

   Refer to SSH Configuration for more information on SSH authentication.

   Enter either **password** (the default) or **publickey**. For example,

   ```
   ORACLE(push-receiver)# auth-type publickey
   ORACLE(push-receiver)#
   ```

8. **username**—specifies the username used to authenticate to this SFTP server

   Provide the username used to authenticate/login to this server. For example,

   ```
   ORACLE(push-receiver)# username acme1
   ORACLE(push-receiver)#
   ```

9. **password**—required when **auth-type** is **password**, and otherwise ignored, specifies the password used in conjunction with **username** to authenticate the SSH client to this SFTP server

   Provide the username used to authenticate/login to this server. For example,

   ```
   ORACLE(push-receiver)# password =yetAnotherPW!
   ORACLE(push-receiver)#
   ```

10. **public-key**—required when **auth-type** is **publickey**, and otherwise ignored, identifies the certificate used in conjunction with **username** to authenticate the SSH client to this SFTP server

    Identify the certificate used to authenticate/login to this server. For example,

    ```
    ORACLE(push-receiver)# publickey certSFTP-1
    ORACLE(push-receiver)#
    ```

11. Type **done** to save your configuration.

Audit Log Alarms and Traps

Three audit log alarms and traps are provided to report significant or anomalous audit log activity.

The ALARM_AUDIT_LOG_FULL trap/alarm is generated in response to (1) the expiration of the file-transfer-time interval, (2) the crossing of the percentage-full threshold, or (3) the crossing of the max-file-size threshold. This trap/alarm is cleared when storage apace becomes available, generally upon successful transfer of the audit log to a remote SFTP server or servers.

The ALARM_ADMIN_AUDIT_PUSH_FAIL trap/alarm is generated in response to failure to transfer the audit log to a designated SFTP server. This trap/alarm is cleared when a subsequent transfer to the same recipient succeeds.

The ALARM_AUDIT_WRITE_FAILED trap/alarm is generated in response to failure to record an auditable event in the audit log. This trap/alarm is cleared when a subsequent write succeeds.

Configure Login Timeouts

Use the **ssh-config** configuration element to set the SSH and TCP timeout values.

1. Access the **ssh-config** element.

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

2. **rekey-interval**—Set the time in minutes after which the OCSBC rekeys an SSH or SFTP session.
   - Min: 60
   - Max: 600
   - Default: 60
3. **rekey-byte-count**—Set the number of bytes transmitted, in powers of 2, before rekeying an SSH or SFTP session.

   For example, entering a value of 24 sets this parameter to 2^24 (16777216) bytes.
   - Min: 20
   - Max: 31
   - Default: 31
4. **proto-neg-time**—Set the time in seconds to complete the SSH protocol negotiation, establishing the secure connection.
   - Min: 30
   - Max: 60
   - Default: 60
5. **keep-alive-enable**—Enable the TCP keepalive timer. Valid Values are:
   - enabled | disabled
   - Default: enabled
6. **keep-alive-idle-timer**—Set the interval in seconds between the last data packet sent and the first keepalive probe.
   - Min: 15
   - Max: 1800
   - Default: 15
7. **keep-alive-interval**—Set the interval in seconds between two successful keepalive transmissions.
   - Min: 15
   - Max: 120
   - Default: 15
8. **keepalive-retries**—Set the number of retransmission attempts before the OCSBC declares the remote end is unavailable.
   - Min: 2
   - Max: 10
   - Default: 2
9. Type done to save the configuration.