Oracle® Communications Session Border Controller

Administrative Security Guide

Release S-Cz8.2.0 - for Service Provider and Enterprise

F20260-04
March 2021

ORACLE

Oracle Communications Session Border Controller Administrative Security Guide, Release S-Cz8.2.0 - for Service Provider and Enterprise

F20260-04

Contents

B Configuring IKEv2 Interfaces

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

Documentation Set

The following table describes the documentation set for this release:

Document Name	Document Description
Acme Packet 3900 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 3900.
Acme Packet 4600 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 4600.
Acme Packet 6100 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6100.
Acme Packet 6300 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6300.
Acme Packet 6350 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 6350.
Release Notes	Contains information about the current documentation set release, including new features and management changes.
ACLI Configuration Guide	Contains information about the administration and software configuration of the Service Provider 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.
SBC Family 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.
HMR Resource Guide	Contains information about configuring and using Header Manipulation Rules to manage service traffic.
TSCF SDK Guide	Contains information about the client-side SDK that facilitates the creation of secure tunnels between a client application and the TSCF of the OCSBC.
REST API Guide	Contains information about the supported REST APIs and how to use the REST API interface.

Revision History

Date	Description
December 2018	• Initial release
October 2019	• Fixes product name in "Admin Security ACP Feature"
June 2020	• Updates the **lockout-interval** parameter description for clarity.
July 2020	• Updates "Password Policy" for clarity. • Updates audit-trail parameter.
March 2021	• Adds 'hostname' to audit file filename

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 (such as !, ", #, $, %, &, ' , (,), *, +, , , -, ., /, :, ;, <, =, >, ?, @, [, \,], ^, _, `, {, |, }, ~)
  - passwords must differ from the prior password by at least 4 characters
  - passwords cannot contain, repeat, or reverse the entire 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 sequential characters from the user ID. This rule is not case sensitive. For example, if the username is "admin," the password cannot contain "ad" nor "AD."
- 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

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."

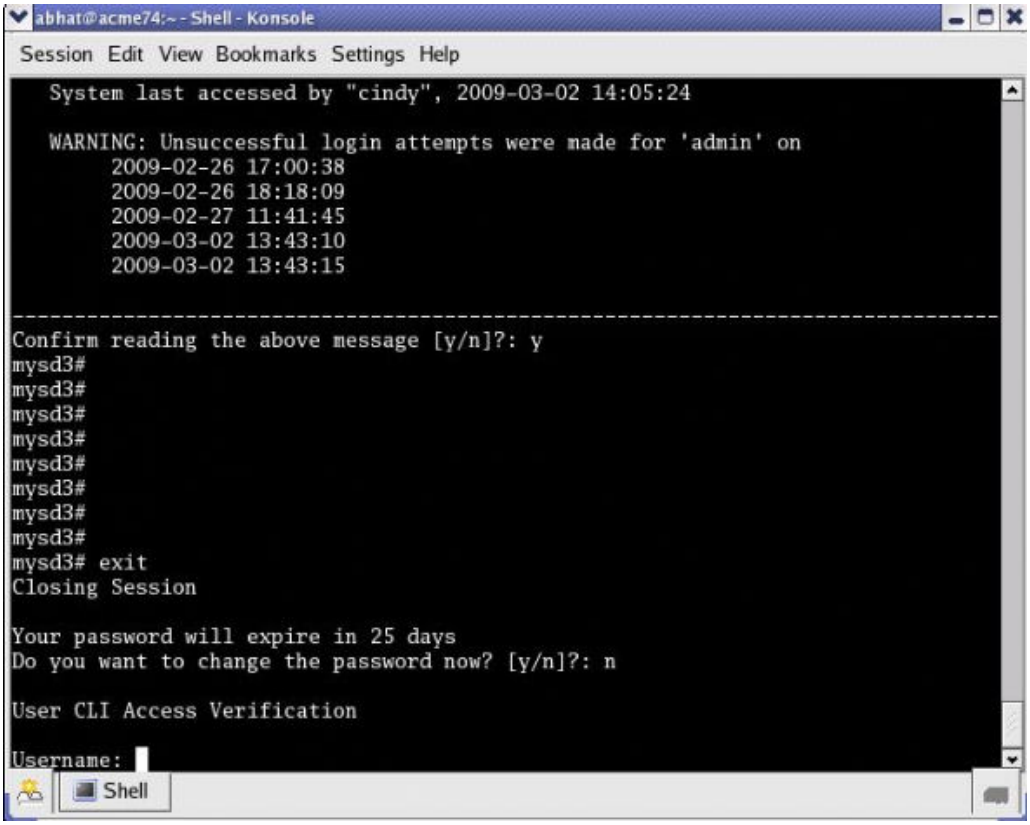
1. Access the **password-policy** configuration element.

   ```
   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.

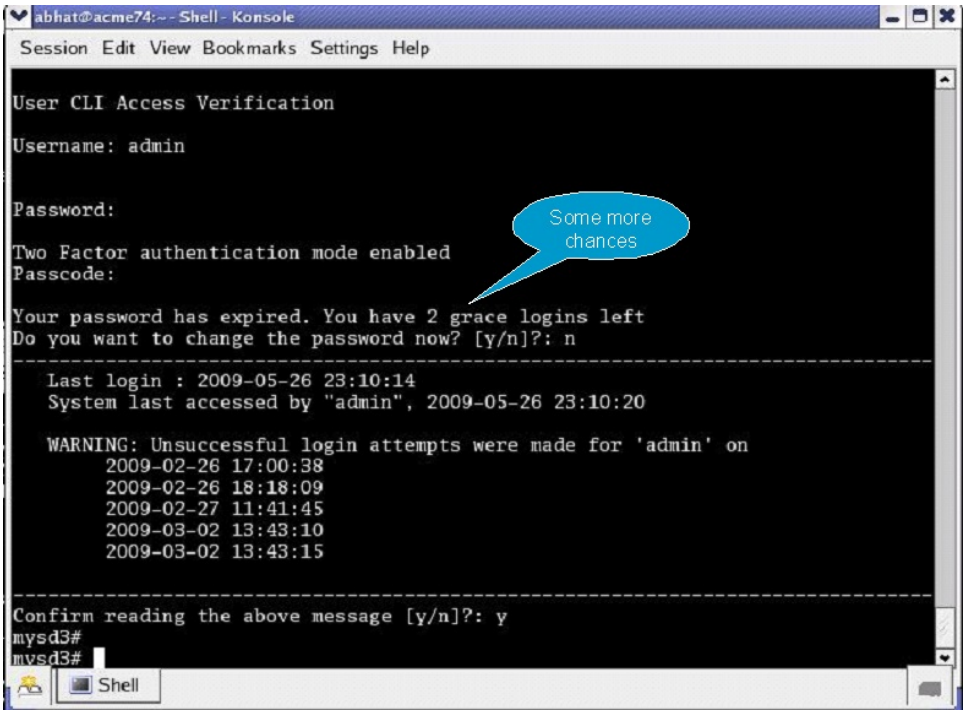
Changing a Password

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 from an interface are not allowed after the **max-login-attempts** threshold has been reached

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

   Retain the default value, or disable login banner display.

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

   A sample login policy configuration appears below:

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

   Defines a login-config configuration element that allows four simultaneous connections per 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:

- 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

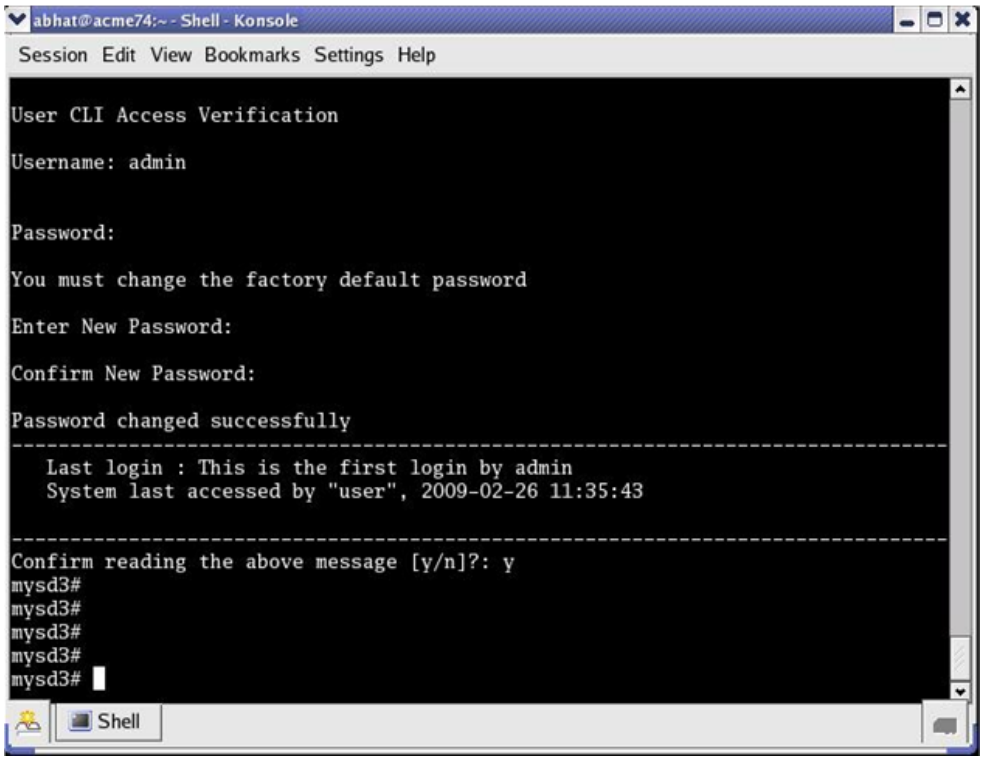
Upon initial login user and admin are required to change the respective password. Initial login is completed only after password change and acknowledgment of the login banner.

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.

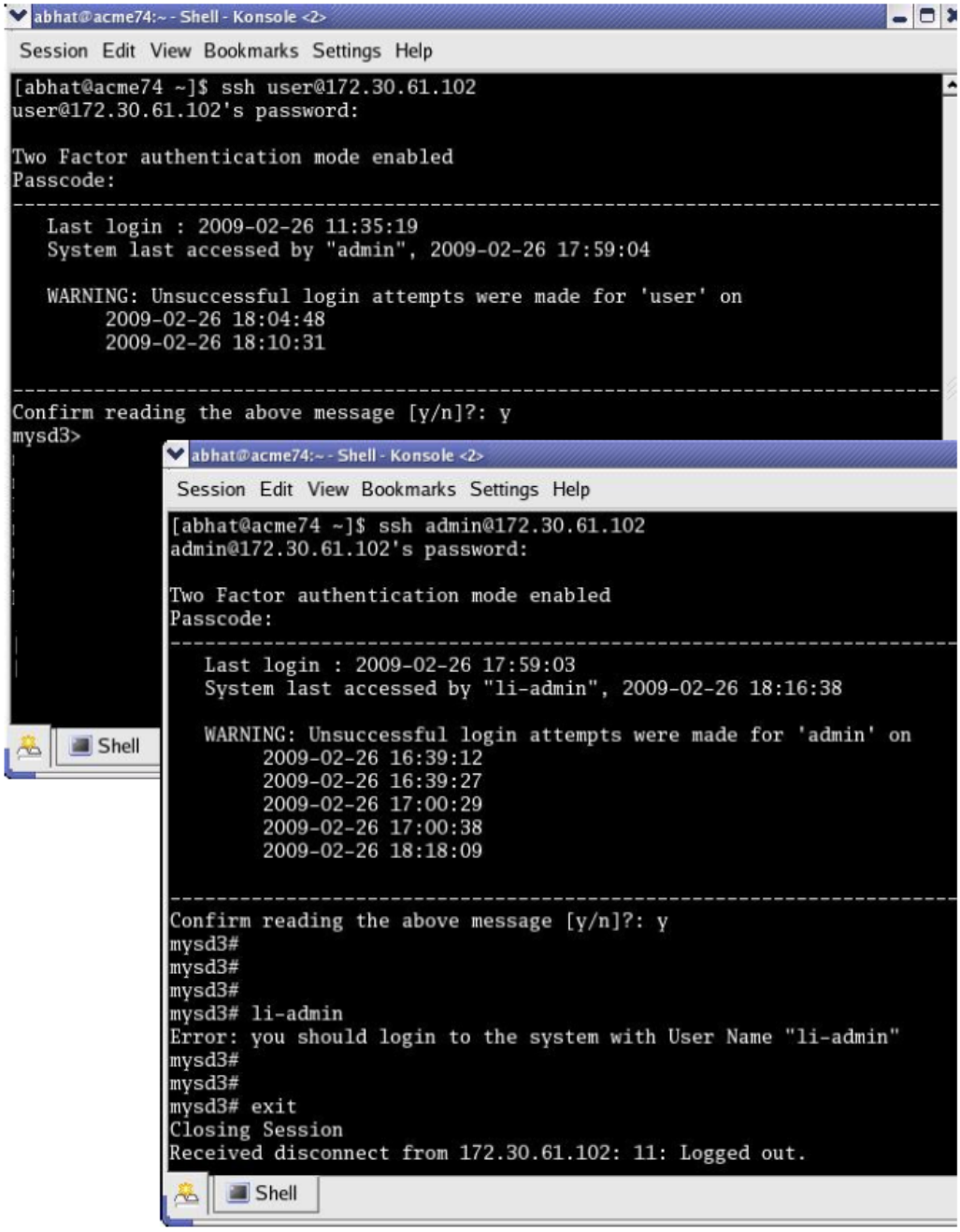
- user password must contain at least 9 characters
- admin password must contain at least 15 characters
- passwords must contain at least 2 lower case alphabetic characters
- passwords must contain at least 2 upper case alphabetic characters
- passwords must contain at least 2 numeric characters
- passwords 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.

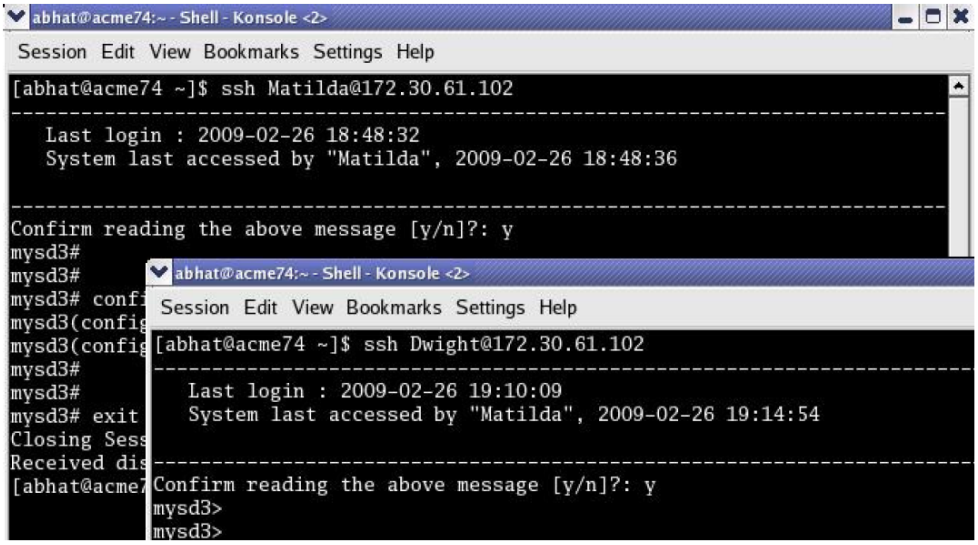
```
ORACLE# ssh-pub-key import authorized-key Matilda admin

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
qJIWdiZuSmny8HZIxTIC6na62iD25mlEdyLhlYOuknkYBCU7UsLwmx4dLDyHTbrQHz3b
1q
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.

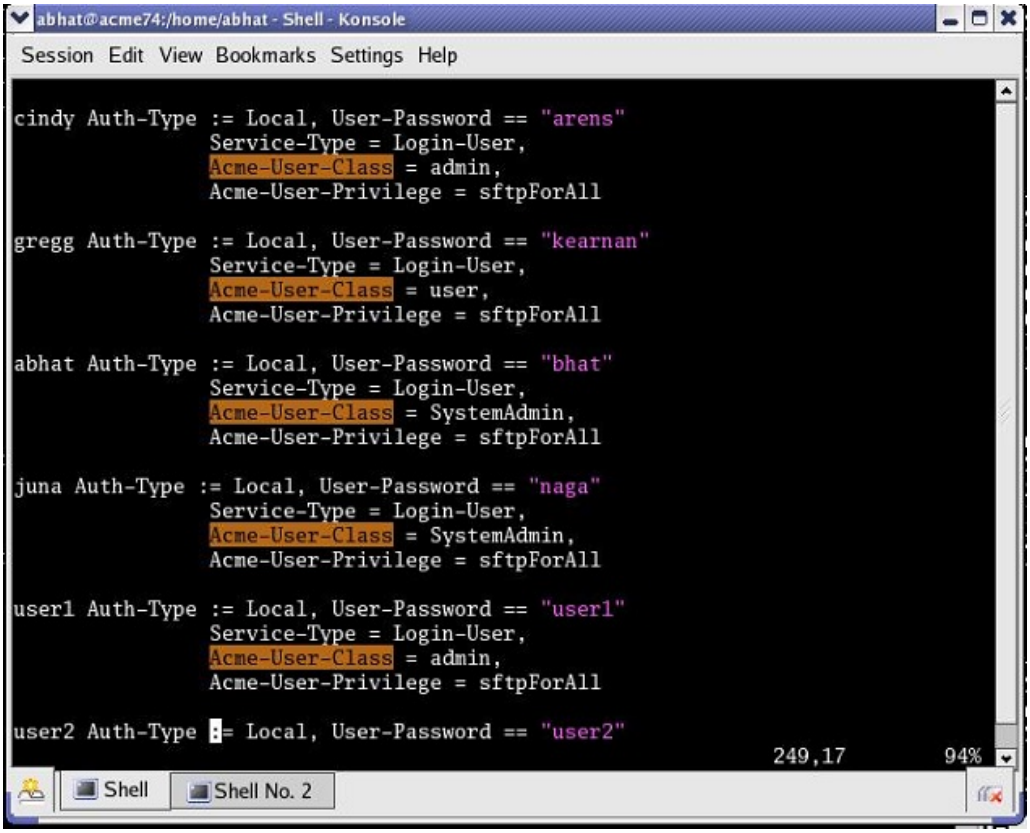
5. Enter the passcode.
6. Confirm the passcode.
7. Type **done** to save the configuration.

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)

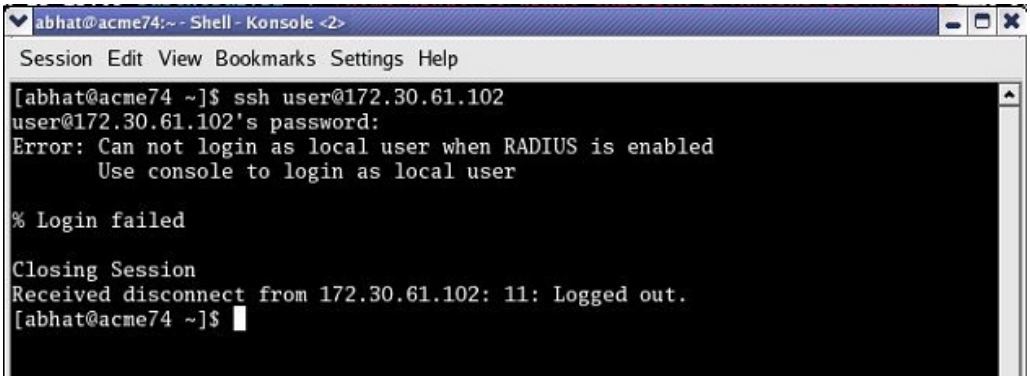
- provides read-write access for system configuration (not including cryptographic keys and certificates)
- The login prompt for this user is ORACLE$

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"
AAAAB3NzaC1yc2EAAAABIwAAAQEA7OBf08jJe7MSMgerjDTgZpbPblrX4n17LQJgPC7c
lL
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
qJIWdiZuSmny8HZIxTIC6na62iD25mlEdyLhlYOuknkYBCU7UsLwmx4dLDyHTbrQHz3b
1q
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+r86kDru88JkUqpelJDx1AR718Ic
pr7ZaAx2L+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:

AAAAB3NzaC1yc2EAAAABIwAAAQEA7OBf08jJe7MSMgerjDTgZpbPblrX4n17LQJgPC7c
lLcDGEtKSiVt5MjcSav3v6AEN2pYZihOxd2Zzismpoo019kkJ56s/
IjGstEzqXMKHKUr9mBVqvqIEOTqbowEi5sz2AP31GUjQTCKZRF1XOQx8A44vHZCum93/
jfNRsnWQ1mhHmaZMmT2LShOr4J/
Nlp+vpsvpdrolV6Ftz5eiVfgocxrDrjNcVtsAMyLBpDdL6e9XebQzGSS92TPuKP/
yqzLJ2G5NVFhxdw5i+FvdHz1vBdvB505y2QPj/iz1u3TA/
3O7tyntBOb7beDyIrg64Azc8G7E3AGiH49LnBtlQf/aw==

modulus: (256)
ECE05FD3C8C97BB3123207AB8C34E06696CF6E5AD7E27D7B2D02603C2EDC94B70318
4B4A4A256DE4C8DC49ABF7BFA004376A5866284EC5DD99CE2B26A68A34D7D924279E
ACFC88C6B2D133A9730A1CA52BF66055AAFA8810E4EA6E8C048B9B33D803F7D46523
41308A6511755CE431F00E38BC7642BA6F77FE37CD46C9D64359A11E66993264F62D
284EAF827F365A7EBE9B2FA5DAE8955E85B73E5E8957E0A1CC6B0EB8CD715B6C00CC
8B0690DD2FA7BD5DE6D0CC6492F764CFB8A3FFCAACCB2761B9355161C5DC398BE16F
747CF5BC176F079D39CB640F8FF8B3D6EDD303FDCEEEDCA7B4139BEDB783C88AE0EB
803373C1BB137006887E3D2E706D9507FF6B
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:

AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbET
W6ToHv8D1UJ/
z+zHo9Fiko5XybZnDIaBDHtblQ+Yp7StxyltHnXF1YLfKD1G4T6JYrdHYI14Om1eg9e4
NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7Yisk/
gF+1VAAAAFQDb8D5cvwHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblw
XdkPggA4pfdtW9vGfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/
FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/
FAAvioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAA
ACBAN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO+JsvphVMBJc
9HSn24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42Z7d
LM5sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV

p: (128)
F63C64E1D8DB2152240E97602F47470347C5A7A1BF1E70389D2BCD9773A12397C5B1
135BA4E81EFF03D5427FCFECC7A3D162928E57C9B6670C86810C7B5B950F98A7B4AD
C7296D1E75C5D582DF283D46E13E8962B747608D783A6D5E83D7B836709195E6AAA1
93C5DD419F6626BA6D7AC64D07F7809AB67BB622B24FE017ED55
q: (20)
DBF03E5CBF01D64D90CF7D7D03DACF5177B341BD
g: (128)
94DF76F816FB0F828B624DC8C116D76E5C177643E0800E297DDB56F6F19F274FD11D
DF8D8C1E1EA350FED1D8B1EAD5F060637B3CA4B947F1573CDC311CF6A9723F6E2F52
67D80590D9DB249DFFA2FC5000BE2A143E499D31CD33B96A12384B12361543B57DD6
76F55C19C06AF5C7ADCEBB4E2963A8709989F34A9A7714D11ED5
pub_key: (128)
DEC263E28ABF5807A51CC5C1D426EC72BD6DBD4B028D8AC1AA179DA74581EA6D3414
1E4971B5BCEF89B2FA6154C04973D1D29F6E1562D62DB0CBBBE2A5EF8988F3895B9C
58A8E32846F5D63BAA9C5D060E50775559B11CB9B19C0CFAE3758AE3667B74B339B1
8DBDA2E7B3BF85F3D8FB8C721E5518F3FE083AB308CE25A16815

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

Secure Radius Connection

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

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

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

- **security**, **authentication**
  - **type** (set to **radius**)
  - **server-assigned-privilege** (set to **enabled**)
  - **authentication-over-ipsec** (set to **enabled**)
  - **management-servers**
- **security**, **authentication**, **radius-server**
  - **address** (the Radius server IP)
  - **secret**
  - **nas-id**
  - **realm-id**
- **security**, **ike**, **ike-config**
  - **log-level**
  - **phase1-dh-mode**
  - **phase2-exchange-mode**
  - **red-port-options**
- **security**, **ike**, **ike-interface**
  - **ike-version** (set to **2**)
  - **address**
  - **realm-id**
  - **ike-mode**
  - **esnSupport** (set to **enabled**)
  - **shared-password**
  - **eap-protocol**
- **security**, **ike**, **ike-sainfo**
  - **name**
  - **tunnel-local-addr**

  - **tunnel-remote-addr**
- **security**, **ipsec**, **security-policy**
  - **name**
  - **network-interface**
  - **priority**
  - **local-ip-addr-match**
  - **remote-ip-addr-match**
  - **ike-sainfo-name**

Factory Reset for the Oracle Communications Session Border Controller

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

> **Caution:**
>
> Factory reset erases 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.

   The audit log file is stored on the target SFTP server or servers with a filename that takes the format:

   ```
   <hostname>-audit<timestamp>
   ```

   Where:

   - <hostname> is the name of the host to which the log gets sent.
   - <timestamp> is a 12-digit string that takes the format YYYYMMDDHHMM.

   ```
   myhost-audit-200903051630
   ```

   Names an audit log file transferred to an SFTP server named 'myhost' 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'
```

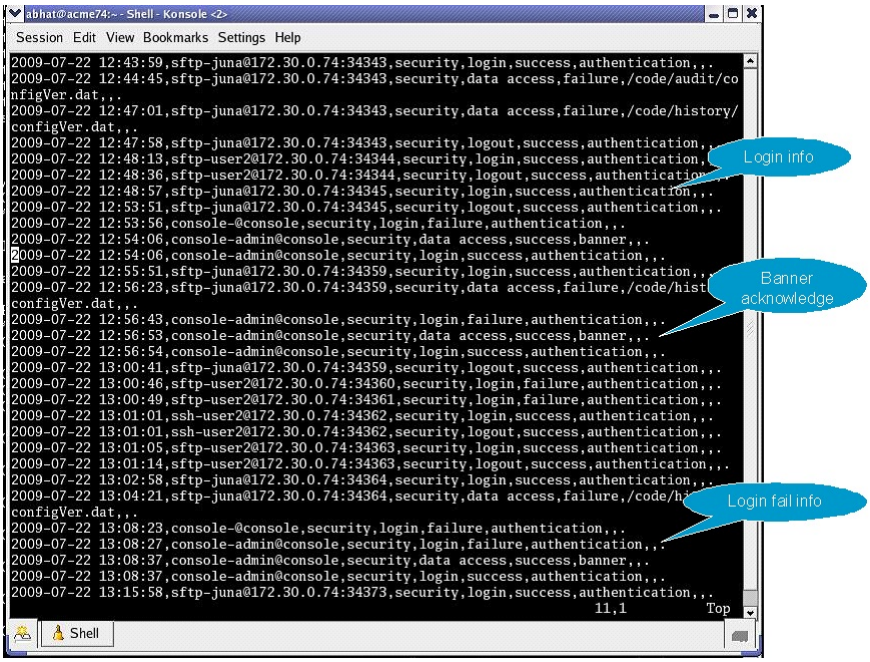
```
  pubKey=''
  privKey=''
  fingerPrint=''
  fingerPrintRaw=''
  lastModifiedBy='acmin@console'
  lastModifiedDate='2009-03-05 15:48:01>
</sshPubKeyRecord
```

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

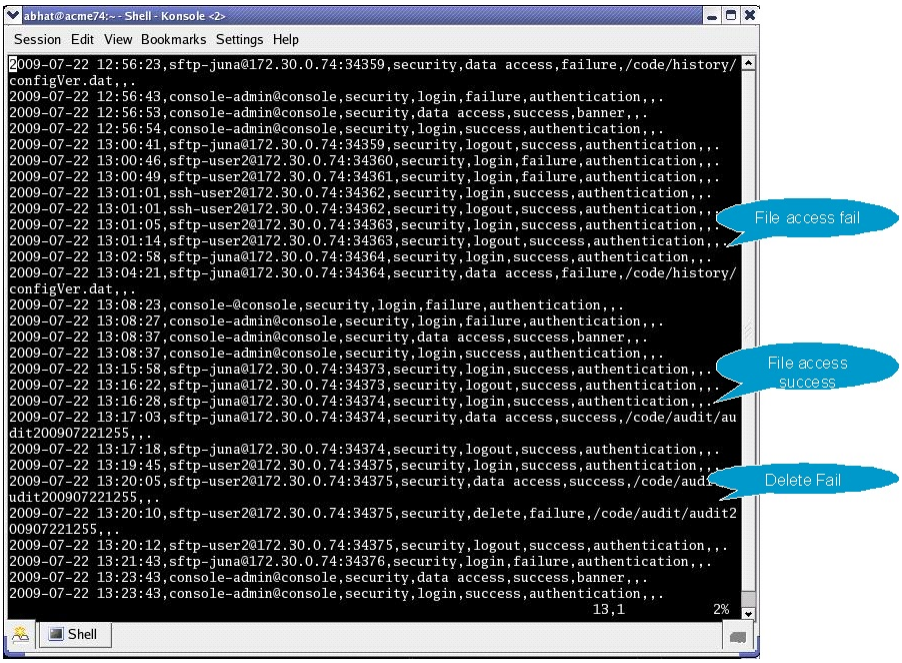
```
2009-03-05 15:51:39,acliConsole-
admin@console,configuration,delete,
success,public-key,
Element=
<?xml version='1.0' standalone='yes'?>
<sshPubKeyRecord
  name='dummy'
  comment=''
  keyType='2'
  encrType='2'
  keySize='1024'
  pubKey=''
  privKey=''
  fingerPrint=''
  fingerPrintRaw=''
  lastModifiedBy='acmin@console'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

Audit Log Samples

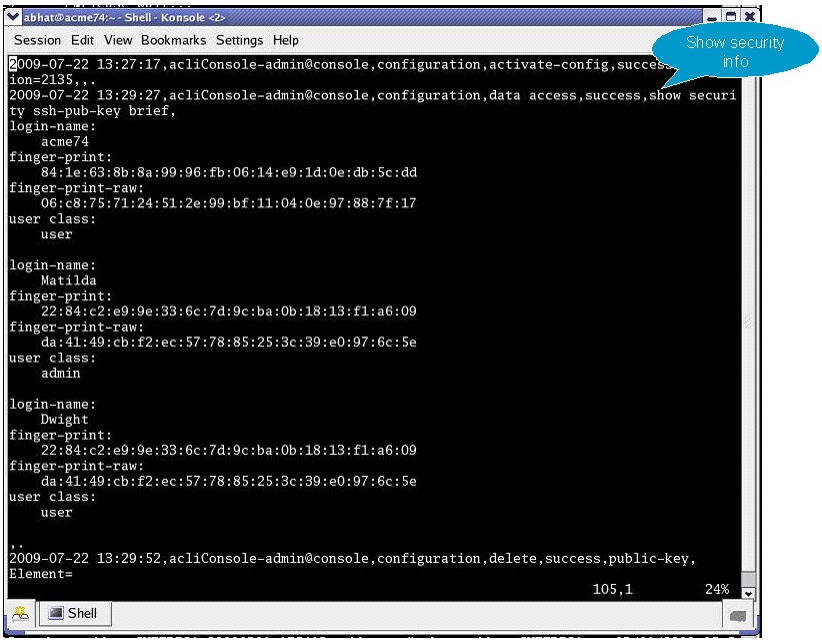
The follow screen captures provide samples of specific audit log entries.



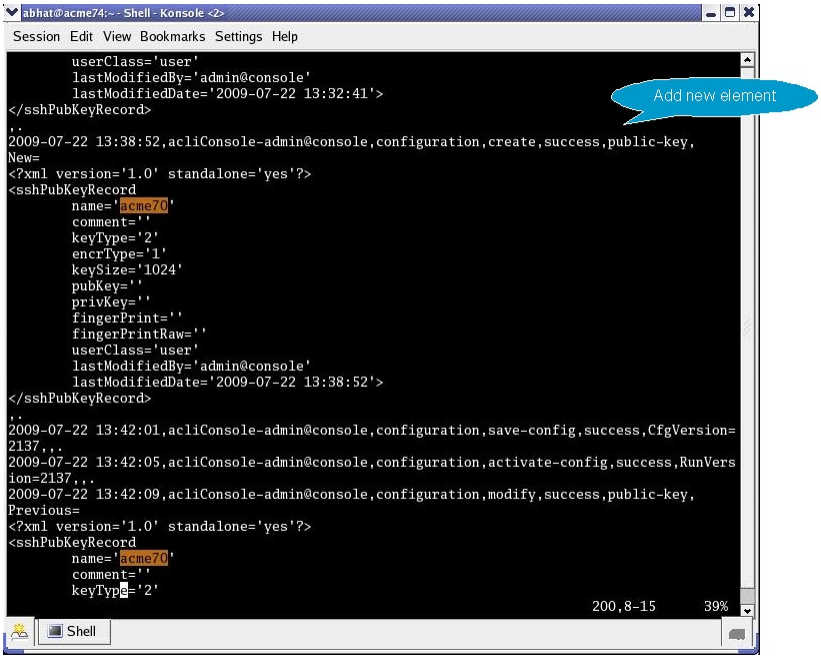
Login Reporting



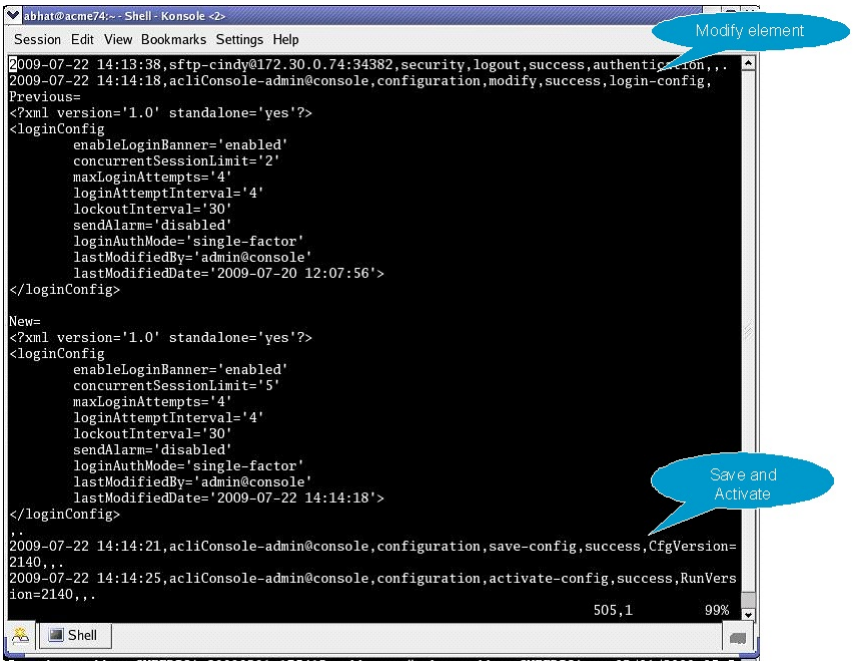
File Access Reporting



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. **state**—Enables or disables audit logging.

   Use enabled to enable audit logging. 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. **audit-trail**—Enables logging every command that is successfully processed by the OCSBC.

Use enabled to enable the audit logging all successful commands. Retain the default value (disabled) to log only relevant information. The value of **state** must be set to enabled for **audit-trail** to work.

> **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.

5. **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.

6. **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.

7. **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.**

8. **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.**

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

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

10. **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.

11. **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.

12. 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.

A

IKEv2 Support

The Oracle Communications Session Border Controller supports version 2 of the Internet Key Exchange (IKE) protocol. IKEv2 provides an initial handshake in which IKE peers negotiate cryptographic algorithms, mutually authenticate, and establish session keys to create an IKEv2 Security Association (SA) and an IPsec SA.

At the IKEv2 global configuration level, users can do the following:

1. Configure IKEv2 global parameters.
2. Configure a default certificate profile.
3. Configure one or more RADIUS authentication servers (optional).
4. Configure one or more RADIUS authorization servers (optional).
5. Configure the default address pool (optional).
6. Configure pre-shared-keys if authentication is based on the contents of the IKEv2 Identification payload (optional).

IKEv2 Global Configuration

A parameter within the global **ike-config** element can be overridden by the same parameter within the **ike-interface** element.

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

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

2. **state**—Set to **enable**.
3. **ike-version**—Select the IKE protocol version **2**.

   > **WARNING:**
   >
   > Enabling version 2 in the **ike-config** element disables version 1 globally.

4. **log-level**—Specify the level of the IKEv2-related logs.

   Log messages are listed below in descending order of severity.

   - emergency
   - critical
   - major
   - minor

- warning
- notice
- info — (default)
- trace — (test/debug, not used in production environments)
- debug — (test/debug, not used in production environments)
- detail — (test/debug, not used in production environments)

5. **udp-port**—Set to **500**.
6. **v2-ike-life-secs**—Specify the default lifetime (in seconds) of the IKEv2 SA.

   Allowable values are within the range 1 through 999999999.
7. **v2-ipsec-life-secs**—Specify the default lifetime (in seconds) for the IPsec SA.

   Allowable values are within the range 1 through 999999999.
8. **v2-rekey** —Enable or disable the re-keying of expired IKEv2 or IPsec SAs.

   When **v2-rekey** is enabled, the OCSBC initiates a new negotiation to restore an expired IKEv2 or IPsec SA. The OCSBC makes a maximum of three retransmission attempts before abandoning the re-keying effort.
9. **sd-authentication-method**—Select the method used for local authentication of the IKEv2 peer.

   Two authentication methods are supported:

   - shared-password — (the default) uses a pre-shared key (PSK) to authenticate the IKEv2 peer.
   - certificate — uses an X.509 certificate to authenticate the IKEv2 peer.

   > **Note:**
   >
   > If using a certificate for authentication, see the "Certificate Configuration Process" section in the Security chapter of the *ACLI Configuration Guide*.

   The **sd-authentication-method** value can be overridden at the **ike-interface** level.
10. **certificate-profile-id**—If using a certificate, specify the **ike-certificate-profile** configuration element that contains identification and verification credentials required for PKI certificate-based IKEv2 authentication.

    The **ike-certificate-profile** value can be over-ridden at the **ike-interface** level.
11. **shared-password**—If using a shared password, provide the PSK used while authenticating the remote IKEv2 peer.

    Ensure the remote peer is configured with the same PSK.

    The value of **shared-password** in the **ike-interface** configuration element takes precedence over this value.
12. **id-auth-type** —(Optional) Specify that the PSK used while authenticating the remote IKEv2 peer is associated with the asserted identity contained within an IKEv2 Identification payload.

> **Note:**
>
> This attribute can be safely ignored if the PSK is defined globally or at the IKEv2 Interface level.

Available values are:

- idi—use IDi KEY_ID for authentication
- idr—use IDr KEY_ID for authentication

13. **addr-assignment**—(Optional) Select the method used to assign a local address in response to an IKEv2 configuration payload request.

    Available values are:

    - local—(the default) use local address pool
    - radius-only—obtain local address from RADIUS server
    - radius-local—try RADIUS server first, then local address pool

14. **eap-bypass-identity**—(Optional) Specify whether or not to bypass the EAP identity phase.
15. **dpd-time-interval** —(Optional) Specify the maximum period of inactivity (in seconds) before the DPD protocol is initiated on an endpoint.

    Values are within the range 1 through 999999999 (seconds).

16. **anti-replay** —(Optional) Enable or disable anti-replay protection on IPsec SAs.
17. **account-group-list**—(Optional) Designate one or two existing IPsec accounting groups as available to support IPsec accounting transactions.
18. Type **done**.

RADIUS Authentication

All EAP-based authentication is performed by RADIUS servers. When such authentication is specified, the Oracle Communications Session Border Controller operates as a relay between the remote IKVv2 peer and a RADIUS authentication server.

Configuring RADIUS Authentication

RADIUS authentication support requires:

- configuration of a pool of RADIUS authentication servers, with each server configuration record providing all values required for server access
- configuration of a RADIUS Authentication Servers List designating specific pool member as being available for authentication purposes
- assignment of the RADIUS Authentication Servers List to the authentication configuration object

Configure a RADIUS Server

1. Access the **radius-servers** configuration element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# security
   ORACLE(security)# authentication
   ORACLE(authentication)# radius-servers
   ORACLE(radius-servers)#
   ```

2. **state**—Set the operational state of this RADIUS authentication server.

   Retain the default value, enabled, to identify this RADIUS authentication server as operational. Use disabled to place this RADIUS authentication server in a non-operational mode.

3. **authentication-methods**—Specify the authentication methods supported by this RADIUS authentication server.

   Valid values are:

   - pap
   - chap
   - mschapv2
   - eap
   - all

4. **address**—Specify the IP address of this RADIUS authentication server.
5. **port**—Specify the remote port monitored for RADIUS authentication requests.

   Valid values are:

   - 1645
   - 1812

6. **realm-id**—Identify the realm that provides transport services to this RADIUS authentication server.
7. **secret**—Specify the shared secret between the Oracle Communications Session Border Controller and this RADIUS authentication server.
8. **nas-id**—Provide a string that uniquely identifies the OCSBC to this RADIUS authentication server.

   For example:

   ```
   ORACLE(radius-servers)# nas-id nas-id-170-30-0-1
   ORACLE(radius-servers)#
   ```

9. **retry-limit**—Specify the number of times the OCSBC retransmits an unacknowledged authentication request to this RADIUS authentication server.

   - Min: 1
   - Max: 5

10. **retry-time**—Specify the interval (in seconds) between unacknowledged authentication requests.
    - Min: 5
    - Max: 10
11. **dead-time**—Specify the length (in seconds) of the quarantine period imposed an unresponsive RADIUS authentication server.
    - Min: 10
    - Max: 10000
12. **maximum-sessions**—Specify the maximum number of outstanding sessions for this RADIUS authentication server.
    - Min: 1
    - Max: 255
13. **class**—Select the RADIUS authentication server class, either primary or secondary.

    The OCSBC tries to initiate contact with primary RADIUS authentication servers first, and only turns to secondary RADIUS authentication servers if no primaries are available.

    If more than one RADIUS authentication server is designated as primary, the OCSBC uses a round-robin strategy to distribute authentication requests among available primaries.
14. Type **done** to save your configuration.
15. If necessary, configure additional RADIUS authentication servers.

Configure a RADIUS Authentication Servers List

1. Access the **auth-params** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# auth-params
ORACLE(auth-params)#
```

2. **name**—Provide a unique name for this RADIUS Authentication Servers List.
3. **servers**—Compile a RADIUS Authentication Servers List.

   Provide the IP address of a previously configured RADIUS authentication server to add that server to this list.

```
ORACLE(auth-params)# servers 172.30.0.1 172.30.0.15 168.27.3.3
ORACLE(auth-params)#
```

4. Type **done** to save your configuration.
5. If necessary, configure additional RADIUS Authentication Servers Lists.
6. Access the **authentication** configuration element.

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

```
ORACLE(security)# authentication
ORACLE(authentication)#
```

7. **ike-radius-params-name**—Assign a previously configured RADIUS Authentication Servers List to the authentication configuration element.
8. Type **done** to save your configuration.

Tearing Down IPsec Tunnels

If EAP-based authentication is used in conjunction with RADIUS-based assignment of requested local addresses, the Oracle Communications Session Border Controller responds to a Disconnect-Request message (as defined in RFC 5176, Dynamic Authorization Extensions to Remote Authentication Dial-In User Service) received from a configured RADIUS server.

The OCSBC parses the received Disconnect-Request for User-Name and Framed-IP-address attribute values. If the User-Name value matches the authenticated EAP identity, and the Framed-IP-address value matches the inner IP address assigned to the authenticated endpoint, the OCSBC deletes the IPsec tunnel described by the received values. Tunnel deletion is reported to the RADIUS server with a Disconnect-ACK message, which, in conformity to Section 3.5 of RFC 5176, contains an Error Cause of 201 indicating Residual Session Context Removed.

If the IPsec tunnel cannot be deleted because of faulty/incorrect User-Name and/or Framed-IP-address values, the OCSBC returns a Disconnect-NACK message, which, in conformity to Section 3.5 of RFC 5176, contains an Error Cause of 404 indicating Invalid Request.

Enable RADIUS Authorization

Complete RADIUS authorization configuration by enabling RADIUS authorization on an IKEv2 interface.

1. Access the **ike-interface** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# ike
ORACLE(ike)# ike-interface
ORACLE(ike-interface)#
```

2. Use the **select** command to specify the target interface.
3. **authorization**—Enable RADIUS authorization on the selected interface.
4. Type **done** to save your configuration.
5. If necessary, enable RADIUS authorization on additional IKEv2 interfaces.

Local Address Pool Configuration

If your network environment requires local address pools that serve as a source of IPv4 or IPv6 addresses temporarily leased for use by remote IKEv2 peers, use the procedures in the following two sections to configure such pools.

During the IKE_AUTH exchange, the IPsec initiator (the remote endpoint) often requests an internal IP address from an IPsec responder (the Oracle Communications Session Border Controller). Refer to Section 2.19 of RFC 7296, Internet Key Exchange (IKEv2) Protocol, for a description of the request process. Procuring such a local IP address ensures that traffic returning to the endpoint is routed to the OCSBC, and then tunneled back to the endpoint. Local address pools provide the source of these addresses available for temporary endpoint lease.

After address assignment from the local address pool, the endpoint retains rights to that IP address for the tunnel lifetime. Tunnels are terminated either by an INFORMATIONAL exchange, defined in Section 1.4 of RFC 7296, or by expiration of the tunnel SAs as specified by the **v2-ike-life-seconds** and **v2-ipsec-life-seconds** configuration parameters. In either case, a subsequent request for an assigned IP address may, or may not result, in the assignment of the previous IP address. However, the OCSBC can be configured to ensure that a prematurely terminated tunnel, resulting for example from the reset or re-boot of the remote IP peer, can be restored with that previous address. Refer to Persistent Tunnel Addressing in this chapter for operational and configuration details.

During the IKE_AUTH request phase, the IKEv2 initiator can use the Configuration payload in conjunction with either the INTERNAL_IP4_DNS or INTERNAL_IP6_DNS attribute to request the addresses of DNS providers from the OCSBC. In environments where authorization is performed by a RADIUS AAA server, there are two potential sources of DNS information: local OCSBC DNS configuration elements, and external RADIUS servers that may provide DNS information in the Access-Accept packet that concludes a successful authentication effort. The source of DNS information provided by the OCSBC to an IKEv2 peer is subject to user configuration.

A RADIUS source of DNS information is enabled by support for certain Microsoft vendor-specific RADIUS attributes specified in RFC2548, Microsoft Vendor-Specific RADIUS Attributes. Operationally, the OCSBC extracts the values of the MS-Primary-DNS-Server and MS-Secondary-DNS-Server attributes from an Access-Accept packet and returns these values to the IKEv2 initiator.

When the DNS information is from external source, the OCSBC installs a NAT flow (a static traffic path) that provides access to the DNS server. The NAT flow is calculated based on the location of the DNS server IP returned from RADIUS AAA server and configured realm information.

Configuration of DNS information services takes place at the local address pool and IKEv2 interface levels.

Data Flow Configuration

If you need to configure address pools, first configure data flows and then assign them to a specific local address pool. A data flow establishes a static route between a remote IKEv2 peer and a core gateway or router which provides routing services after the associated traffic exits the Oracle Communications Session Border Controller.

1. Access the **data-flow** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# ike
ORACLE(ike)# data-flow
ORACLE(data-flow)#
```

2. **name**—Provide a unique identifier for this data-flow instance.
3. **realm-id**—Identify the realm that supports data-flow upstream traffic, that is traffic toward the network core.
4. **group-size**—(Optional) Specify the maximum number of user elements grouped together by this **data-flow** instance.

   The size of the associated local-address-pool is divided by this value to segment the address pool into smaller groups. After determining the start address for each of the smaller address groups, the OCSBC uses the **data-flow** configuration to establish two static flows for each of the address groups — a downstream data-flow, in the access direction, and an upstream data-flow (via the realm specified by the **realm-id** parameter) toward a core gateway/router which provides forwarding service for the pass-thru data-flow.

   Allowable values are the powers of 2 between 1 through 256.

   ```
   ORACLE(data-flow)# group-size 32
   ```

5. **upstream-rate**—Specify the allocated upstream bandwidth.
   - Min: 0 (allocate all available bandwidth)
   - Max: 122070
6. **downstream-rate**—Specify the allocated downstream bandwidth.
   - Min: 0 (allocate all available bandwidth)
   - Max: 122070
7. Type **done** to save your configuration.

Local Address Pool Configuration

You configure an address pool by associating a contiguous range or ranges of IPv4 or IPv6 addresses with an existing data-flow.

> **Note:**
>
> An address pool can contain multiple contiguous ranges of IP addresses. However, all defined ranges must specify the same type of IP address: You cannot include IPv4 and IPv6 addresses in the same address pool.

1. Access the **local-address-pool** configuration element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# security
   ORACLE(security)# ike
   ORACLE(ike)# local-address-pool
   ORACLE(local-address-pool)#
   ```

2. **name**—Provide a unique identifier for this local-address-pool instance.

3. **dns-assignment**—Identify the DNS source used to respond to incoming IKE_AUTH requests for DNS information.
   - **local**—Use locally configured configuration data as the source of DNS information
   - **radius**—Use a remote RADIUS AAA server as the source of DNS information.
   - **radius-local**—Use a remote RADIUS AAA server as the preferred source of DNS information. If no DNS data is available from the RADIUS server, use locally configured DNS information.
4. **dns-realm-id**—Provide the name of the realm that supports transit to that RADIUS server.

   The **dns-realm-id** parameter can be safely ignored if **local** is specified as the DNS source.
5. **data-flow**—Identify the data-flow configuration element assigned to this local-address-pool instance.
6. **address-range**—Access the **address-range** configuration mode.
   - If building an address pool of contiguous IPv4 addresses, use **network-address** with **subnet-mask** to define a contiguous range of IPv4 addresses.

     ```
     ORACLE(address-range)# network-address 192.168.0.0
     ORACLE(address-range)# subnet-mask 255.255.255.96
     ```

   - If building an address pool of contiguous IPv6 addresses, use **network-address** parameter to provide both the IPv6 address and the bit length of the network prefix (an integer within the range 1 through 128). Leave the **subnet-mask** blank.

     ```
     ORACLE(address-range)# network-address 1080::ac10:202/96
     ```

7. Type **done** to save your configuration. and **exit** to complete configuration of the address-range instance.
8. If required, add additional address ranges to this address-range instance
9. Type **done** to complete configuration of the local-address-pool instance.

Persistent Tunnel Addressing

After address assignment from the local address pool, the endpoint retains rights to that IP address for the tunnel lifetime. Tunnels can be terminated either by an INFORMATIONAL exchange, defined in Section 1.4 of RFC 7296, or by expiration of the tunnel SAs as specified by the **v2-ike-life-seconds** and **v2-ipsec-life-seconds** parameters. In either case, a subsequent request for an assigned IP address may, or may not result, in the assignment of the previous IP address. However, the Oracle Communications Session Border Controller can be configured to ensure that a prematurely terminated tunnel can be restored with that previous address.

Tunnels are usually prematurely terminated because of re-boot or reset of the remote endpoint. In either case, the endpoint's IKEv2 and IPsec SAs are lost and the tunnel no longer exists. From the point of view of the OCSBC, however, the tunnel remains live. The local IKEv2 and IPsec SAs still exist, and the tunnel remains available in

an active state until the expiration of the lifetime timers. Similarly, the IP address assignment from the local address poll remains in effect until timer expiration.

When a crashed endpoint attempts to re-establish a tunnel, it can insert a Notify payload in the initial IKE_AUTH request. The Notify payload contains an INITIAL_CONTACT message that asserts a prior connection between the endpoint and the OCSBC. When receiving an INITIAL_CONTACT message, the OCSBC checks for the existence of a live tunnel with the requesting endpoint. If such a tunnel is found, the OCSBC stores the assigned IP address, tears down the tunnel by removing the supporting IKEv2 and IPsec SAs, and authenticates the endpoint. Assuming authentication succeeds, the OCSBC, retrieves the previously assigned IP address and returns it to the endpoint.

If a live tunnel is not found (meaning that the tunnel has timed out during the interval between the endpoint reset/re-boot and the new IKE_AUTH), the assertion of a prior connection is ignored, and address assignment is made from the local address pool.

You can use a global configuration option (**assume-initial-contact**) to enable persistent address processing with or without the reception of an INITIAL_CONTACT message. With this option enabled, all IKE_AUTH requests are processed as if they contained an INITIAL_CONTACT message.

Persistent Tunnel Addressing Configuration

Use the following command sequence to enable persistent tunnel addressing.

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

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

2. **options**—Enable address persistence.

```
ORACLE(ike-config)# options +assume-initial-contact
ORACLE(local-address-pool)#
```

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

ike-key-id Configuration

If authentication between IKEv2 peers is based on a PSK associated with an identity asserted in the IKE Identification Payload, associate received asserted identities with a specified PSK.

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

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# ike
ORACLE(ike)# ike-key-id
ORACLE(ike-key-id)#
```

2. **name**—Provide a unique identifier for this ike-keyid instance.
3. Use **keyid** and **presharedkey** parameters to associate an asserted identity with a PSK.

```
ORACLE(ike-keyid)# keyid 172.16.20.20
ORACLE(ike-keyid)# presharedkey **********************
```

4. Type **done**to save your configuration.
5. Repeat to configure additional ike-keyid instances.

B

Configuring IKEv2 Interfaces

After configuring global IKE parameters, use the procedures described in this chapter to configure and monitor IKEv2 interfaces.

IKEv2 interface configuration consists of the following steps.

1. Configure IKE interface attributes
2. Configure Security Associations
3. Configure Security Policies
4. Configure the Dead Peer Detection Protocol (optional)
5. Configure the Online Certificate Status Protocol or Certificate Revocation List Support (optional)
6. Configure Threshold Crossing Alerts (optional)
7. Configure access control whit/black lists (optional)

EAP-based Authentication

RFC 3748, Extensible Authentication Protocol (EAP) describes a flexible and extensible framework that enables authentication services. While the RFC itself describes only a single authentication method, MD5-Challenge, the provided framework support numerous authentication methods.

The current release supports the seven EAP-based authentication methods described in the following sections. Note that for all currently supported EAP authentication methods that the actual authentication is provided by an adjacent RADIUS server. During the EAP-based authentication exchange the OCSBC functions as a packet relay between the authenticating client(s) and the RADIUS server.

EAP Authentication Methods

EAP supports several authentication methods.

EAP-MD5

EAP-MD5 is based on RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*. This RFC describes an authentication method that uses an agreed-upon hashing algorithm, a random challenge value, and a shared secret known only to the authenticator and the EAP peer. In the case of EAP-MD5 the hashing algorithm, which produces a 128-bit message-digest or fingerprint, is described in RFC 1321, *The MD5 Message-Digest Algorithm*.

Using EAP-MD5, authentication of the EAP peer is accomplished as follows.

1. The authenticator issues a Challenge packet, which contains, among other fields, an Identifier field that serves to correlate message exchanges, and a Data field that contains an arbitrary challenge string.

2. The peer concatenates the contents of the Identifier field, the shared-secret, and the challenge string. The peer inputs the concatenated string to the MD5 hash function, computes the 128-bit fingerprint, and returns that value to the authenticator in a Response packet.
3. The authenticator performs the same calculation, and compares its results with those reported by the EAP peer.
4. If the fingerprints are identical, the authenticator issues a Success packet; otherwise the authenticator issues a Failure packet.

> **Note:**
>
> EAP-MD5 does not provide for mutual authentication; the authenticator does not authenticate to the EAP peer.

EAP-MSCHAPv2

EAP-MSCHAPv2 is based on RFC 2759, *Microsoft PPP CHAP Extensions, Version 2*. This RFC describes an authentication method that uses a user-name and password model in conjunction with Microsoft encryption routines. Using EAP-MSCHAPv2, mutual authentication of the EAP peer and authenticator is accomplished as follows:

1. The authenticator issues a Challenge packet, which contains, among other fields, an Identifier field that serves to correlate message exchanges, and a Data field that contains an arbitrary 16-octet challenge string.
2. The peer returns a Response packet that includes the user name, a newly-generated 16-octet challenge for the authenticator, and a one-way encryption of the received challenge string, the generated challenge string, the contents of the Identifier field, and the user password.
3. The authenticator performs the same calculation as was performed by the EAP peer, and compares its results with those reported by the peer. If the results are identical, the authenticator issues a Success packet which also contains a one-way encryption of the authenticator-originated challenge string, the peer-originated challenge string, the encrypted string received from the peer in the Response packet, and the user password.
4. The EAP peer performs the same calculation as was performed by the authenticator, and compares its results with those reported by the authenticator. If the results are identical, the peer uses the mutually authenticated connection; otherwise, it drops the connection.

EAP-AKA

The Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA) was devised by the 3GPP (3rd Generation Partnership Project), and made available to the Internet community in RFC 4187. EAP-AKA makes use of the Universal Subscriber Identity Module (USIM), an application resident on the smart card inserted in a 3G mobile phone. The USIM has access to authentication data stored on the smart card.

EAP-SIM

The EAP-SIM Protocol specifies an authentication method for GSM (Global System for Mobile Communication) subscribers. GSM is a second generation mobile standard,

and still the most widely used. The authentication method is described in RFC 4186, Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identify Modules (EAP-SIM). Originally developed by the 3GPP, the EAP-SIM protocol specifies an EAP method for authentication and session key distribution using the GSM Subscriber Identity Module (SIM), a smart card installed in the GSM phone.

EAP-TLS

EAP-TLS uses a Transport Layer Security (TLS) handshake, encapsulated within the secure tunnel, to mutually authenticate client and server (or an AAA backend) with certificates. The OCSBC acts in EAP pass-through mode to communicate the EAP-TLS negotiation between the device and the AAA server.

EAP-TTLS

The EAP-TTLS authentication method is useful when there is no certificate-based infrastructure present for the operator to configure a certificate for each device. EAP-TTLS consists of a Tunneled Transport Layer Security (TTLS) handshake phase (similar to EAP-TLS) and a data phase. During the data phase, the client is authenticated to the server (or the client and server are mutually authenticated) using an arbitrary authentication mechanism encapsulated within the secure tunnel. Thus, EAP-TTLS allows legacy password-based authentication protocols to be used against existing authentication databases, while protecting the security of these legacy protocols against eavesdropping, man-in-the-middle, and other attacks.

EAP-AKA

EAP-AKA' is a small revision to the EAP-AKA (Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement) method. The change is a new key derivation function that binds the keys derived within the method to the name of the access network. The new key derivation mechanism has been defined in the 3rd Generation Partnership Project (3GPP). This feature allows its use in EAP in an interoperable manner. Additionally, EAP-AKA' employs SHA-256 instead of SHA-1 as the Secure Hash Algorithm.

Multiple Authentication

The Oracle Communications Session Border Controller supports multiple authentication exchanges during IKEv2 negotiation. These exchanges are defined in RFC 4739, Multiple Authentication Exchanges in the Internet Key Exchange (IKEv2) Protocol. Multiple authentication enables the OCSBC to engage in an initial certificate-based or shared-secret-based authentication with a remote IKEv2 peer (for example, a femtocell), followed by a subsequent EAP-AKA or EAP-SIM authentication of the remote mobile subscriber.

Multiple authentication exchanges require the use of two specific Notify payloads, MULTIPLE_AUTH_SUPPORTED and ANOTHER_AUTH_FOLLOWS (Notify message type s16404 and 16405) defined in Sections 3.1 and 3.2 of RFC 4739.

Message exchange is as follows.

```
Initiator (IKEv2 peer)
Responder
1. HDR, SAi1, KEi, Ni --->
2.            <--- HDR, SAr1, KEr, Nr, CERTREQ, N
```

```
(MULTIPLE_AUTH_SUPPORTED)
3. HDR, { IDi, CERT, CERTREQ, {IDr], AUTH, SAi2, TSi, TS2
   (MULTIPLE_AUTH_SUPPORTED) N (ANOTHER_AUTH_FOLLOWS) } --->
4.                                          <--- HDR, { IDr, CERT,
AUTH }
5. HDR, { IDi } --->
6.                                             <--- HDR, { EAP
(Request)}
7. HDR, { EAP (Response) } --->
8.                                             <--- HDR, { EAP
(Request)}
9. HDR, { EAP (Response) } --->
10.                                            <--- HDR, { EAP
(Success)}
11. HDR, { AUTH } --->
12.                                    <--- HDR, { AUTH, SAr2, TSi,
TSr }
```

In Step 2 the responder advertises support for multiple authentication via the MULTIPLE_AUTH_SUPPORTED Notification Payload.

In Step 3 the initiator advertises support for multiple authentication and, using the ANOTHER_AUTH_FOLLOWS Notification Payload, signals its readiness for such authentication.

Step 4 completes mutual certificate authentication.

In Step 5 the initiator discloses its identity.

In Step 6 the responder initiates the EAP process

In Steps 7 and 8 the initiator and responder exchange authentication information for the remote peer.

In Steps 9 and 10 the initiator and responder exchange authentication information for the mobile subscriber.

Steps 11 and 12 report successful authentication.

IPv6 Inner Tunnel Address Assignment

The Oracle Communications Session Border Controller supports the assignment of IPv6 inner tunnel addresses utilizing an external RADIUS server as the IPv6 address source. During the EAP authentication of an IPsec host, neither the OCSBC nor the RADIUS authentication server has any knowledge of the traffic type (IPv4 or IPv6) that the IPsec host intends to transmit through the tunnel. Consequently, the RADIUS authentication server may send both IPv4 and IPv6 attributes in the RADIUS Access-Accept message, leaving it to the OCSBC to select the appropriate attribute and ignore the other.

The OCSBC makes its decision based on the contents of the Configuration Payload received from the IPsec host. If the payload contains an INTERNAL_IP4_ADDRESS attribute, the IPv4 address received in the Access-Accept message is forwarded to the IPsec host. In a similar fashion, if the payload contains an INTERNAL_IP6_ADDRESS attribute, the IPv6 address received in the
Access-Accept message is forwarded to the IPsec host.

Assignment of IPv6 addresses requires support for the following RADIUS attributes:

- Framed-IPv6-Prefix (Type 97) — also used in RADIUS accounting
- Framed-IPv6-Pool (Type 100)

Framed-IPv6-Pool, which can be returned by a RADIUS authentication server in an Access-Accept message, contains the name of an address pool that should be used by the OCSBC as a source of IPv6 addresses.Use of Framed-IPv6-Pool requires the pre-configuration of the identified address pool on the OCSBC.

EAP-only Authentication

IKEv2 specifies that Extensible Authentication Protocol (EAP) authentication must be used together with responder authentication based on public key signatures. This is necessary with old EAP methods that provide only unilateral authentication using, for example, one-time passwords or token cards. With EAP-SIM, EAP-AKA, EAP-AKA', EAP-TTLS, and EAP-TLS, which provide mutual authentication and key agreement, extensible responder authentication for IKEv2 based on methods other than public key signatures can be used. This feature causes the OCSBC to default to EAP-only authentication without using public-key-based responder authentication unless the operator selects otherwise.

The Extensible Authentication Protocol, defined in RFC3748, is an authentication framework that supports multiple authentication mechanisms. One of the advantages of the EAP architecture is its flexibility. Rather than requiring the authenticator (for example, a wireless LAN access point) to be updated to support each new authentication method, EAP permits the use of a backend authentication server that may implement some or all authentication methods. The OCSBC uses a backend authentication server (for example, 3GPP AAA) and is in pass-through mode for EAP.

IKEv2 is a component of IPsec used for performing mutual authentication and establishing and maintaining Security Associations (SAs) for IPsec Encapsulating Security Payload (ESP) and Authentication Header (AH). In addition to supporting authentication using public key signatures and shared secrets, IKEv2 also supports EAP authentication. By using EAP, IKEv2 can leverage existing authentication infrastructure and credential databases, such as Home Subscriber Server (HSS), as EAP allows users to choose a method suitable for existing credentials, and also makes separation of the IKEv2 responder (OCSBC) from the EAP authentication endpoint (back-end Authentication, Authorization, and Accounting (AAA) server) easier. IKEv2 specifies that these EAP methods must also be used together with responder authentication based on public key signatures. For the public key signature authentication of the OCSBC to be effective, a deployment of Public Key Infrastructure (PKI) is required, which has to include management of trust anchors on all supplicants. This may not realistic in WiFi calling environments, in which the security of the OCSBC public key is the same as the security of a self-signed certificate. Mutually authenticating EAP methods alone can provide a sufficient level of security.

Because of these reasons, the OCSBC now defaults to EAP-only authentication without using public-key-based responder authentication unless the operator selects otherwise by disabling the new parameter **eap-only-support** in the **ike-interface** configuration element.

EAP-only Authentication Configuration

1. Access the **ike-interface** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# ike
ORACLE(ike)# ike-interface
ORACLE(ike-interface)#
```

2. Select the **ike-interface** object to edit.

```
ORACLE(ike-interface)# select
<address>:

ORACLE(ike-interface)#
```

3. **eapOnlyAuthSupport** — The default is **enabled**. Set the value to **disabled** to use EAP authentication together with responder authentication based on public key signatures.
4. Type **done** to save your configuration.

Debugging IKEv2 IPsec Tunnel Establishment

The Oracle Communications Session Border Controller provides details of all IKE endpoints that establish IKEv2/IPsec tunnels. Logging can also be enabled by IP address and
userid.

In a typical deployment scenario, the IP address can be the public address of a NAT device that communicates with the Oracle Communications Session Border Controller; the user-id can be the user-id of a femtocell or an IKE endpoint residing behind the NAT. The user-id can be an EAP identity exchanged during EAP authentication, or the identity contained in the IDi payload of the initial IKE_AUTH message. Typically the identity in the IDi payload is an IP address, an FQDN, or an address as defined in RFC 822, *Standard for the Format of ARPA Internet Text Messages*.

Enabling/Disabling Targeted Debugging

Targeted debugging is enabled by the **security ike debug-logging peer-ip-userid** ACLI command which takes a single string argument in the form ipAddress:userID. For example:

```
ORACLE# security ike debug-logging peer-ip-userid
172.16.20.1:12EDE12626719
ORACLE#
```

With endpoint-specific logging enabled, the log.iked, log.authd, and log.secured files are populated with data pertinent to the target endpoint only and exclude date for all other endpoints. Logging is based on an exact match of the IP address and user-id provided by the argument string.

> **Note:**
>
> This command is expensive and should be used to debug one or two endpoints at a time. The operating system imposes a hard limit of no more than 5 simultaneous targeted debugging sessions.

Use the no form of the command to stop an existing targeted debugging session

```
ORACLE# security ike debug-logging peer-ip-userid
172.16.20.1:12EDE12626719 no
ORACLE#
```

Use the **show security ike peer-endpoint-logging** ACLI command to display a list of configured debug-logging sessions

```
ORACLE# show security ike peer-endpoint-logging
ORACLE#
IPaddress : Userid
==============
172.16.20.1:12EDE12626719
ORACLE#
```

High Availability Caveat

Since the security ike debug-logging peer-ip-userid command is expensive, this implementation intentionally does NOT synchronize log data on the active and standby HA devices. Consequently, in the event of a switchover from the active to the standby, no log data is available on the newly active device. To enable debug-logging on the new active device, the user should verify tunnel establishment, and then use security ike debug-logging peer-ip-userid command on the currently active member of the HA pair.

Configure an IKEv2 Interface

Global values set in the **ike-config** configuration element can be overridden by values set at the **ike-interface** level.

1. Access the **ike-interface** configuration element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# security
   ORACLE(security)# ike
   ORACLE(ike)# ike-interface
   ORACLE(ike-interface)#
   ```

2. **state**—Enable the IKEv2 interface.
3. **ike-version**—Set this attribute to **2**.

4. **address**—Specify the IPv4 or IPv6 address of the interface.

```
ORACLE(ike-interface)# address 10.0.0.10
```

5. **realm-id**—Specify the realm that contains the IP address assigned to this IKEv2 interface.

```
ORACLE(ike-interface)# realm-id access-10
```

6. **ike-mode**—Specify whether the OCSBC will act as a responder or initiator.
7. **sd-authentication-method**—Select the interface-specific method used by IKEv2 peers to authenticate to each other.
   - shared-password—Use a pre-shared-secret to authenticate the remote IKEv2 peer.
   - certificate—Use an X.509 certificate to authenticate the remote IKEv2 peer.

> **Note:**
>
> **sd-authentication-method** can be safely ignored, if authentication utilizes any of the methods described in EAP-based Authentication.

```
ORACLE(ike-interface)# sd-authentication-method shared-password
```

8. **shared-password**—If using the shared-password authentication method, set the shared password.
9. **certificate-profile-id-list**—If using the certificate authentication method, identify the **ike-certificate-profile** configuration element that contains identification and validation credentials required for certificate-based IKEv2 authentication.
10. **multiple-authentication**—Enable or disable multiple authentication as defined in RFC 4739 on this IKEv2 interface.

    The default is **disabled**.
11. **v2-ike-life-seconds**—(Optional) Specify the lifetime (in seconds) for the IKEv2 SAs supported by this IKEv2 interface.

    The default is 86400 (24 hours).
    - Min: 1
    - Max: 999999999
12. **v2-ipsec-life-seconds**—(Optional) Specify the lifetime (in seconds) for the IPsec SAs supported by this IKEv2 interface.

    The default is 28800 (8 hours).
    - Min: 1
    - Max: 999999999
13. **v2-rekey**—(Optional) Enable or disable the automatic re-keying of expired IKEv2 or IPsec SAs on this IKEv2 interface.

With automatic re-keying enabled, and with the global **dpd-time-interval** parameter set to a non-zero value, the OCSBC retransmits the re-keying request if it does not receive a response from the remote IPsec peer within the interval specified by the ike-config **dpd-time-interval** parameter. The OCSBC makes a maximum of three retransmission attempts before abandoning the re-keying effort.

14. **dpd-params-name**—Enable the Dead Peer Detection Protocol on this IKEv2 interface.

    The protocol is initially enabled by setting a non-zero value to the **dpd-time-interval** parameter during IKEv2 global configuration process. The protocol is enabled at the local level by assigning an existing dpd-params configuration element to this IKEv2 interface.

    Refer to Dead Peer Detection Protocol Configuration in this chapter for information on configuring dpd-params configuration elements.

    ```
    ORACLE(ike-interface)# dpd-params-name ikeDPD
    ```

15. **cert-status-check**—(Optional) Enable certificate status checking using either Online Certificate Status Profile (OCSP) or a local copy of a Certificate Revocation List.

    The default is **disabled**.

16. **cert-status-profile-list**—(Optional) Assign one or more **cert-status-profile** configuration elements to this IKEv2 interface.

    Each assigned cert-status-profile provides the information needed to access either an OCSP responder or a CRL source.

    > **Note:**
    >
    > Use quotation marks to assign multiple OCSP responders.

    ```
    ORACLE(ike-interface)# cert-status-profile-list
    "VerisignClass3Designate Verisign-1 Thawte-1"
    ```

17. **access-control-name**—(Optional) Assign an existing access control white or black list to this IKEv2 interface.

    This parameter is meaningful only when authentication uses a RADIUS server to implement the EAP-based authentication, and can otherwise be safely ignored. White lists and black lists specify IMSI prefixes or MAC addresses that are allowed through or denied access to the RADIUS authentication server.

    ```
    ORACLE(ike-interface)# access-control-name white_01
    ```

18. **addr-assignment**—(Optional) Specify the method used to assign addresses in response to an IKEv2 Configuration Payload request.

    The Configuration payload supports the exchange of configuration information between IKEv2 peers. Typically, a remote IKEv2 peer initiates the exchange by requesting an IP address on the gateway's protected network. In response, the OCSBC returns a local address for the peer's temporary use.

    Supported values are:

- local—(the default) use local address pool
- radius-only—obtain local address from RADIUS server
- radius-local —try RADIUS server first, then local address pool

```
ORACLE(ike-interface)# addr-assignment local
```

19. **local-address-pool-id-list**—(Optional) Assign one or more existing address pools to the current interface, if **addr-assignment** is local or radius-local.

    Local address pools provide a group of IP address that can be temporarily leased to remote endpoints who request an IP address on a OCSBC subnet, and also specify DNS information sources made available to remote endpoints.

    During the IKE_AUTH exchange, the IKEv2 initiator (the remote endpoint) often requests an internal IP address from an IPsec responder (the OCSBC). Refer to Section 2.19 of RFC 4306, *Internet Key Exchange (IKEv2) Protocol*, for a description of the request process. Procuring such a local IP address ensures that traffic returning to the endpoint is routed to the OCSBC, and then tunneled back to the endpoint. Local address pools provide the source of these addresses available for temporary endpoint assignment.

    After address assignment from the local address pool, the endpoint retains rights to that address for the tunnel lifetime, which is terminated either by an INFORMATIONAL exchange as defined in Section 1.4 of RFC4306, or by expiration of the tunnel SAs as specified by the **v2-ike-life-seconds** and **v2-ipsec-life-seconds** parameters. In either case, a subsequent request for an assigned IP address results, in all likelihood, with the assignment of a new IP address. However, if the remote endpoint is prematurely terminated by, for example an unscheduled reset or re-boot, a subsequent request for an assigned IP address (assuming that SA timers have not expired) results in the assignment of the previously issued IP address.

    ```
    ORACLE(ike-interface)# local-address-pool-id-list ikePool
    ```

20. **eap-protocol**—(Optional) Set the EAP protocol.

    Available values are:

    - eap-md5
    - eap-tls
    - eap-leap
    - eap-sim
    - eap-srp
    - eap-ttls
    - eap-aka
    - eap-peap
    - eap-mschapv2
    - eap-fast
    - eap-psk
    - eap-radius-passthru

21. Type **done** to save your configuration.
22. Configure additional IKEv2 interfaces if required.

IPsec Security Policy Configuration

You first define ike-sainfo elements that identify cryptographic material available for Security Association negotiation, and then define interface-specific IPsec Security Policies.

IPsec SA Configuration

During the IKE_AUTH exchange, cooperating peers use the secure channel previously established by the IKE_SA_INIT exchange to negotiate child IPsec SAs to construct secure end-to-end IPsec tunnels between the peers. IKE_SA_INIT negotiations use the values provided by the ike-sainfo configuration element.

Use the following procedure to create an ike-sainfo configuration element that specifies cryptographic material used for IPsec tunnel establishment. You will later assign this ike-sainfo configuration element to an IPsec Security Policy which defines IPsec services for a specified IKEv2 interface.

1. Access the **ike-sainfo** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# ike
ORACLE(ike)# ike-sainfo
ORACLE(ike-sainfo)#
```

2. **name**—Provide a unique identifier for this ike-sainfo configuration element.

```
ORACLE(ike-sainfo)# name SA-1
```

3. **security-protocol**—Specify the IPsec security (authentication and encryption) protocols supported by this SA.

   The default value is **ah**. Supported values are:

   - ah—Authentication Header. Provides authentication integrity to include the mutual identification of remote peers, non-repudiation of received traffic, detection of data that has been altered in transit, and detection of data that has been replayed, that is copied and then re-injected into the data stream at a later time.
   - esp—Encapsulating Security Payload provides both authentication and privacy services.
   - esp-auth—Supports ESP's optional authentication
   - esp-null—Provides NULL encryption.

> **WARNING:**
>
> This option provides no privacy services and is not recommended for production environments.

4. **auth-algo**—Specify the authentication algorithms supported by this SA.

   Available protocols are:

   - any
   - md5
   - sha1
   - xcbc
   - sha2-256
   - sha2-384
   - sha2-512

5. **encryption-algo**—Specify the encryption algorithms supported by this SA.

   The default is **aes**. Available protocols are:

   - any—Choose any
   - 3des—Triple DES
   - aes—AES with CBC mode
   - aes-ctr—AES with counter mode
   - null—NULL encryption

6. **ipsec-mode**—Specify the IPsec operational mode.

   - tunnel—Provides a secure end-to-end connection between two IP hosts.
   - transport—Provides VPN service where the entire IP packets are encapsulated within an outer IP envelope and delivered from source (an IP host) to destination (generally a secure gateway) across an untrusted internet.

7. **tunnel-local-addr**—If using tunnel mode, specify the IP address of the local IKEv2 interface that terminates the IPsec tunnel.

   ```
   ORACLE(ike-sainfo)# tunnel-local-addr 172.30.89.10
   ```

8. **tunnel-remote-addr**—If using tunnel mode, specify the IP address of the remote IKEv2 peer that terminates the IPsec tunnel.

   Provide the remote IP address or use the default wild-card value (*) to match all IP addresses.

   ```
   ORACLE(ike-sainfo)# tunnel-remote-addr *
   ```

9. Type **done** to save your configuration.
10. If necessary, configure additional IPsec SAs.

Security Policy Configuration

Use the following procedure to define an IPsec Security Policy.

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

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

2. **name**—Identify this IPsec Security Policy.

   ```
   ORACLE(security-policy)# name requireIPsec
   ```

3. **network-interface**—Provide the network interface name of the IKEv2 interface to which this security policy is applied.

   ```
   ORACLE(security-policy)# network-interface M00:0
   ```

4. **priority**—(Optional) Assign a priority to this IPsec Security Policy.
   - Highest priority: 0
   - Lowest priority: 123
5. **action**—Specify the processing of IPsec and non-IPsec traffic streams.
   - allow—Process non-IPsec traffic
   - ipsec—Allow only IPsec traffic
   - srtp—Allow only SRTP traffic
   - srtcp—Allow only SRTCP traffic
6. **direction**—Identity the traffic streams subject to the processing specified by the **action** parameter.

   Available values are:
   - in
   - out
   - both
7. **local-ip-addr-match**—(Optional) Specify the local IP address of the network interface.

   Provide the local IP address or retain the default value, 0.0.0.0, which matches all local IP addresses.

   ```
   ORACLE(security-policy)# local-ip-addr-match 172.30.89.10
   ```

8. **remote-ip-addr-match**—(Optional) Specify the IP address of the remote IKEv2 peer.

Provide the remote IP address or retain the default value, 0.0.0.0, which matches all remote IP addresses.

```
ORACLE(security-policy)# remote-ip-addr-match 0.0.0.0
```

9. **local-port-match**—(Optional) Specify the local ports to which this IPsec Security applies.

   Use 0 to specify all local ports.

   - Min: 1
   - Max: 65535

10. **remote-port-match**—(Optional) Specify the remote ports to which IPsec Security Policy applies.

    Use 0 to specify all remote ports.

    - Min: 1
    - Max: 65535

11. **ike-sainfo-name**—Assign an IPsec data SA to this Security Policy.
12. Type **done** to save your configuration.

Enable Tunnel Pass-Through

Use IPsec Security Policies to enable tunnel pass-through.

Pass-through IPv4 traffic via an IPv4 tunnel

1. Configure IPv4 allow policy for IKE protocol traffic
2. Configure IPv4 ipsec policy for media traffic
3. Configure the IKEv2 IPv4 interface with an IPv4 local address pool, or
4. Configure the RADIUS server to return a Framed-IP-Address and/or Framed-IP-Netmask attribute

Pass-through IPv6 traffic via an IPv6 tunnel

1. Configure IPv6 allow policy for IKE protocol traffic
2. Configure IPv6 ipsec policy for media traffic
3. Configure the IKEv2 IPv6 interface with an IPv6local address pool, or
4. Configure the RADIUS server to return a Framed-IPv6-Prefix or Framed-IPv6-Pool attribute

Pass-through IPv4 traffic via an IPv6 tunnel

1. Configure IPv6 allow policy for IKE protocol traffic
2. Configure IPv4 ipsec policy for media traffic
3. Configure the IKEv2 IPv6 interface with an IPv4 local address pool, or
4. Configure the RADIUS server to return a Framed-IP Address and/or Framed-IP-Netmask attribute

Pass-through IPv6 traffic via an IPv4 tunnel

1. Configure IPv4 allow policy for IKE protocol traffic
2. Configure IPv6 ipsec policy for media traffic
3. Configure the IKEv2 IPv4 interface with an IPv6local address pool, or
4. Configure the RADIUS server to return a Framed-IPv6-Prefix or Framed-IPv6-Pool attribute

IPSec SA Rekey on Sequence Number Overflow

The Oracle Communications Session Border Controller establishes a new IPSec security association (SA) when the counter for the outbound 32-bit Sequence Number (SN) or the 64-bit Extended Sequence Number (ESN) overflows.

The SN or ESN counter is incremented for every outbound packet. These counters can overflow when the OCSBC is handling packet intensive services such as video streaming or long duration calls. In accordance with RFCs 4303 and 7296, the OCSBC establishes new security associations, as part of rekeying, before the SN or ESN counters can roll over. It does this through the use of two parameters in the **ipsec-global-config** configuration element: **rekey-on-sn-overflow**, the default for which is **enabled**, and **sn-rekey-threshold**, which identifies the threshold for rekeying security associations as a percentage of the counter capacity and for which the default is **95**.

There are four ACLI commands you can use to monitor SN and ESN counter overflows:

show datapath etc-stats ppms ipsec

Issuing this command shows, along with other existing IPSec PPM-related statistics, the total number of times SN overflow occurred. The four pertinent parameters are:

- **ob-sn-threshold-overflows** — This counter is incremented when the SN for an outbound SA for a tunnel exceeds the user-configured threshold value.
- **ob-sn-32bit-overflows** — This counter is incremented when the lower 32-bits of the outbound ESN (when ESN is enabled) overflows.
- **standby-ob-sn-overflows** — This counter is incremented when the SN or ESN for an outbound SA for a tunnel overflows the threshold value installed on the standby node during SA installation or update on the standby system.
- **ib-sn-32bit-overflows** — This counter is incremented when the lower 32 bits of the inbound ESN (when ESN is enabled) overflows.

show datapath netlink show

Issuing this command shows the total number of SN overflow notifications received by the netlink layer on the host processor. The four newly-added parameters are the same as those in **show datapath etc-stats ppms ipsec**.

show sa stats ike

Issuing this command shows the number of times an SN overflow triggered a request for an IPsec rekey to acquire a new SA, as well as the number of times rekey requests succeeded and failed.

show security ike statistics

Issuing this command shows, with the parameter **RekeyOnSNoverflow** the number of times an SN overflow triggered an IPsec rekey.

IPSec SA Rekey on Sequence Number Overflow Configuration

1. Access the **ipsec-global-config** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# ipsec
ORACLE(ipsec)# ipsec-global-config
ORACLE(ipsec-global-config)#
```

2. Select the **ipsec-global-config** object to edit.

```
ORACLE(ipsec-global-config)# select
ORACLE( ipsec-global-config)#
```

3. **rekey-on-sn-overflow** — Identifies whether to enable IPSec rekey on sequence number (SN) or extended sequence number (ESN) overflow. Rekey initiation is independent of the value of the parameter **v2-rekey** in the **ike-interface** configuration element. Allowable values are **enabled** and **disabled**. The default is **enabled**.
4. **sn-rekey-threshold** — Identifies the threshold for triggering an IPSec security association (SA) rekey on SN or ESN overflow as a percentage of the SN (32-bit) or ESN (64-bit) number space. The allowable range is **80** to **100** and the default is **95**.
5. Type **done** to save your configuration.

Pre-Populated ARP Table

In certain topologies remote IPsec endpoints can require access to core network hosts reachable through a Oracle Communications Session Border Controller core interface. In these instances, the OCSBC receives the tunneled packet, and masks the received IP destination address against its own local addresses to determines if direct delivery is possible. If so, the OCSBC issues an ARP request to obtain the physical destination address.

This process can be expedited by pre-populating the interface-specific ARP table with a list of commonly accessed core network host reachable by that interface. With the ARP table pre-populated with IP addresses, the ARP process issues ARP requests at 5 second intervals until a response is received. Once the pre-populated IP address has been resolved, periodic ARP refreshes are used to maintain the currency of the resolution.

Pre-Populate An Interface-Specific ARP Table

1. Access the **network-interface** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# system
ORACLE(system)# network-interface
ORACLE(network-interface)
```

2. Select the **network-interface** object to edit.

```
ORACLE(network-interface)# select
<name>:<sub-port-id>:
1: wancom0:0 ip=10.0.0.2 gw=10.0.4.1

selection: 1
ORACLE(network-interface)#
```

3. **add-neighbor-ip**—Add the initial IP address to the core-interface-specific ARP table.

```
ORACLE(network-interface)# add-neighbor-ip 10.0.0.101
```

4. If necessary, add an additional IP address to the core-interface-specific ARP table.

   You can add a maximum of ten IP addresses to a single network interface.

5. Use the **show** command to examine the pre-populated ARP table, referred to as the neighbor list.

```
ORACLE(network-interface)# show
network-interface
        ...
        neighbor-list                   10.0.0.101
                                        10.0.0.102
                                        10.0.0.103
                                        10.0.0.104
                                        10.0.0.105
                                        10.0.0.106
                                        10.0.0.107
                                        10.0.0.108
                                        10.0.0.109
                                        10.0.0.110
        ...
```

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

Configure Dead Peer Detection

Dead Peer Detection is enabled by setting the dpd-time-interval parameter to a non-zero value. DPD exchanges are asynchronous, consisting of a simple R-U-THERE and an ACK.

1. Access the **dpd-params** configuration element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# security
   ORACLE(security)# ike
   ORACLE(ike)# dpd-params
   ORACLE(dpd-params)#
   ```

2. **name**—Provide a unique identifier for this dpd-params instance.

   ```
   ORACLE(dpd-params)# name ikeDPD
   ```

3. **max-loop**—Specify the maximum number DPD peers whose liveliness is examined every **dpd-interval** period.

   Periodic liveliness is tested by the Oracle Communications Session Border Controller issuing an R-U-THERE message to each peer in the current group. If the peer acknowledges receipt of the message, it is confirmed as alive. If the peer fails to respond, its status is determined by the max-retrans and max-attempts parameter values.

   - Min: 1
   - Max: 999999999

4. **max-retrans**—Specify the maximum number of times that the OCSBC, acting as a DPD initiator, retransmits an unacknowledged R-U-THERE message while performing periodic liveliness tests.

   The default is 3.

   - Min: 1
   - Max: 4

5. **max-attempts**—Specify the number of failed liveliness tests required to declare a peer as dead and take down the IKE tunnel.

   The default is 1.

   - Min: 1
   - Max: 4

6. **max-endpoints**—Specify the maximum number of simultaneous DPD protocol negotiations supported when the CPU is not under load, as specified by **max-cpu-limit**.

   The default is 25.

   - Min: 1
   - Max: 15000

   If CPU workload surpasses the threshold set by max-cpu-limit, this value is over-ridden by load-max-endpoints.

7. **max-cpu-limit**—Specify a threshold value (expressed as a percentage of CPU capacity) at which DPD protocol operations are minimized to conserve CPU resources.

   The default is 60.

- Min: 0
- Max: 100

8. **load-max-loop**—Specify the maximum number of endpoints examined every **dpd-time-interval** when the CPU is under load, as specified by **max-cpu-limit**.

   The default is 40.

   - Min: 1
   - Max: 999999999

   Ensure that the configured value is less than the value assigned to **max-loop**.

9. **load-max-endpoints**—Specify the maximum number of simultaneous DPD Protocol negotiations supported when the CPU is under load, as specified by **max-cpu-limit**.

   The default is 5.

   - Min: 1
   - Max: 15000

   Ensure that the configured value is less than the value assigned to **max-endpoints**.

10. Type **done** to save your configuration.
11. If necessary, configure additional dpd-params configuration elements.
12. Access the **ike-interface** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# ike
ORACLE(ike)# ike-interface
ORACLE(ike-interface)#
```

13. **dpd-params-name**—Enable Dead Peer Detection on this IKEv2 interface.

```
ORACLE(ike-interface)# dpd-params-name ikeDPD
```

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

Certificate Revocation Lists

A Certificate Revocation List (CRL) contains a list of the serial numbers of certificates that have been revoked by the issuing Certification Authority (CA). Such issuing authorities update CRLs periodically, and make the updates lists available to subscribers. CRL updates can be deliver in either PEM (Privacy Enhanced Email) or DER (Distinguished Encoding Rules) format. PEM is base-64 encoded ASCII that provides BEGIN CERTIFICATE and END CERTIFICATE statements; DIR is a binary rendering of the PEM format. Both formats (PEM and DIR) are supported by the Oracle Communications Session Border Controller.

When authentication of remote IKEv2 peers is certificate-based, you can enable CRL usage on IKEv2 interfaces to verify certificate status.

CRL-Based Certificate Verification

This section provides instruction on using the ACLI to configure periodic retrieval of CRLs.

Configuration of CRL-based certificate verification is a three-step process.

1. Specify the information and cryptological resources required to access one or more CRL sources.
2. If not already done, enable CRL usage on an IKEv2 interface.
3. Associate one or more CRLs with an IKEv2 interface.

Configure CRL Certificate Verification

The **cert-status-profile** element is a container for the information required to access a specific CRL source.

1. Access the **cert-status-profile** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# cert-status-profile
ORACLE(cert-status-profile)#
```

2. **name**—Provide a unique name for this profile.
3. **type**—Select the certificate revocation check method.

   Available values are:

   - OCSP
   - CRL
4. Specify either the IP address or the hostname of the CRL source.
   - **ip-address**—Specify the IP address of the CRL source.
   - **host-name**—Specify the hostname of the CRL source

   > **Note:**
   >
   > If values are provided for both attributes, the OCSBC uses the IP address and ignores the **host-name** value.

5. **crl-list**—Specify the source filepath(s) to one or more requested CRLs.

   For example:

```
ORACLE(cert-status-profile)# crl-list /crl/v2/tc_class_3_ca_II.crl
```

6. **realm-id**—Specifies the realm used to request and receive CRLs.

   In the absence of an explicitly configured value, the OCSBC provides a default value of wancom0, specifying CRL-related transmissions across the wancom0 management interface.

> **Note:**
>
> If the CRL source is identified by its FQDN, the realm identified by **realm-id** must be DNS-enabled.

7. **responder-cert**—Identify the certificate used to validate the received CRL (the public key of the CRL source).

   Provide the name of the certificate configuration element that contains the certificate used to validate the signed CRL.

8. **retry-count**—Specify the maximum number of times to retry an CRL source in the event of connection failure.

   The default is 1.

   - Min: 0
   - Max: 10

9. **dead-time**—Specify the quarantine period imposed on an unavailable CRL source.

   The default is 0.

   - Min: 0
   - Max: 3600

10. **crl-update-interval**—Specify the interim in seconds between CRL updates.

    The default is 86400.

    - Min: 600
    - Max: 2600000

    CRLs are stored in the /code/crls directory. Outdated, invalid CRLs are over-written with the each newly-obtained current CRL.

11. Type **done** to save your configuration.
12. If necessary, configure additional **cert-status-profile** configuration elements.

SNMP Traps

An SNMP trap is thrown, and a major alarm generated, if the Oracle Communications Session Border Controller is unable to retrieve a CRL from the server. This trap includes the server's FQDN, assuming that the FQDN has been identified during the configuration process, the server's IP address, the reason for the failure, and the time of the last successful CRL retrieval, with the time expressed as the number of seconds since midnight January 1, 1970.

A second SNMP trap is thrown when the OCSBC successfully retrieves a CRL. This trap includes the server's FQDN, assuming that the FQDN has been identified during the configuration process, and the server's IP address. The issue of this trap also clears any associated major alarm.

Configuring Manual CRL Updates

The ACLI provides the ability to perform an immediate manual refresh of one or more CRLs.

Use the following command to refresh a single CRL.

```
ORACLE# load-crl local-file <fileName>
```

where <fileName> is a remote filepath specified by the crl-list attribute.

Use the following command to refresh all CRLs.

```
ORACLE# load-crl local-file all
```

Use the following command to refresh all CRLs from a specific CRL source.

```
ORACLE# load-crl cert-status-profile <certStatusProfileName>
```

where <certStatusProfileName> references the certificate-status-profile configuration element that contains the CRL source IP address or FQDN.

Use the following command to refresh all CRLs.

```
ORACLE# load-crl cert-status-profile all
```

Online Certificate Status Protocol

The Online Certificate Status Protocol (OCSP) enables users to determine the revocation state of a specific certificate. Because OCSP ensures access to the freshest CRL, it can provide a more timely source of revocation information than is possible with dynamically or manually loaded CRLs. Guaranteed access to the most recent CRL, however, comes at the expense of increased traffic: a single request/response exchange for each revocation check.

If the OCSP responder returns a status of good, the certificate is accepted and authentication succeeds. If the OCSP responder returns a status other than good, the certificate is rejected and authentication fails.

Certificate status is reported as

- good—which indicates a positive response to the status inquiry. At a minimum, a positive response indicates that the certificate is not revoked, but does not necessarily mean that the certificate was ever issued or that the time at which the response was produced is within the certificate's validity interval.
- revoked—which indicates a negative response to the status inquiry. The certificate has been revoked, either permanently or temporarily.
- unknown—which indicates a negative response to the status inquiry. The responder cannot identify the certificate.

When authentication of remote IKEv2 peers is certificate-based, you can enable OCSP on IKEv2 interfaces to verify certificate status.

OCSP-Based Certificate Verification

The following sections provides instruction on using the ACLI to configure OCSP-based certificate verification.

Configuration of OCSP-based certificate verification is a three-step process.

1. Specify the information and cryptological resources required to access one or more OSCP responders.
2. Enable OCSP on an IKEv2 interface.
3. Associate one or more OCSP responders with an IKEv2 interface.

Configure OCSP Certificate Verification

1. Access the **cert-status-profile** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# cert-status-profile
ORACLE(cert-status-profile)#
```

2. **name**—Provide a unique name for this profile.
3. **type**—Select the certificate revocation check method.

   Available values are:

   - OCSP
   - CRL
4. Specify either the IP address or the hostname of the CRL source.
   - **ip-address**—Specify the IP address of the CRL source.
   - **host-name**—Specify the hostname of the CRL source

> **Note:**
>
> If values are provided for both attributes, the OCSBC uses the IP address and ignores the **host-name** value.

5. **realm-id**—Specify the realm used to transmit OCSP requests and receive OCSP responses.

   In the absence of an explicitly configured value, the OCSBC provides a default value of wancom0, specifying OCSP protocol transmissions across the wancom0 management interface.
6. **requester-cert**—Specify the certificate used to sign requests.

   Ignore this attribute if requests are not signed. If a signed request is required by the OCSP responder, provide the name of the certificate configuration element that contains the certificate used to sign OCSP requests.
7. **responder-cert**—Identifies the certificate used to validate signed OCSP response (a public key of the OCSP responder).

> **Note:**
>
> RFC 2560 requires that all OCSP responders digitally sign OCSP responses, and that OCSP requesters validate incoming signatures.

8. **retry-count**—Specify the maximum number of times to retry an CRL source in the event of connection failure.

   The default is 1.

   - Min: 0
   - Max: 10

9. **dead-time**—Specify the quarantine period imposed on an unavailable CRL source.

   The default is 0.

   - Min: 0
   - Max: 3600

10. Type **done** to save your configuration.
11. If necessary, configure additional **cert-status-profile** configuration elements.

SNMP Traps

An SNMP trap is thrown if a configured OSCP responder becomes unreachable.

A second SNMP trap is thrown when connectivity is re-established with a previously unreachable OCSP responder.

Enable Certificate Verification on an IKEv2 Interface

1. Access the **ike-interface** configuration element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# security
   ORACLE(security)# ike
   ORACLE(ike)# ike-interface
   ORACLE(ike-interface)#
   ```

2. Select the **ike-interface** object to edit.

   ```
   ORACLE(ike-interface)# select
   <address>:

   ORACLE(ike-interface)#
   ```

3. **cert-status-check**—Enable certificate status checking on this IKEv2 interface.
4. **cert-status-profile-list**—Assign a CRL source or sources to the IKEv2 interface

> **Note:**
>
> Use quotation marks to assign multiple CRL sources.

```
ORACLE(ike-interface)# cert-status-profile-list "CRL1-VS CRL2-VS
CRL3-VS"
ORACLE(ike-interface)#
```

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

Configuring Access Control

The OCSBC supports IKEv2 access-control white lists that permit authentication only for a provisioned list of IMSI prefixes or MAC addresses. The OCSBC also supports black lists that deny authentication to a provisioned list of IMSI prefixes or MAC addresses.

Configuring White Lists

Use the procedures described in this section only when authentication is performed by the EAP-SIM protocol. This section can be ignored when the Oracle Communications Session Border Controller employs any other authentication method.

EAP-SIM Protocol Overview

The EAP-SIM Protocol is described in RFC 4186, Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identify Modules (EAP-SIM). Originally developed by the 3GPP (3rd Generation Partnership Project), the EAP-SIM protocol provides for mutual authentication between the authenticator (a RADIUS server) and a GSM subscriber.

Within the EAP-SIM framework the GSM subscriber identifies itself with its International Mobile Subscriber Identity (IMSI), a digit string providing a globally unique identity for the subscriber's device. The IMSI is stored on a Subscriber Identity Module (SIM) installed in the GSM phone.

The IMSI is usually a 15-digit string that takes the following form:

```
<MCC><MNC><MSIN>
```

- MCC (Mobile Country Code) prefix — 3 digits that uniquely identify the carrier's residence, not the subscriber's current location
- MNC (Mobile Network Code) prefix — 2 or 3 digits that identify the carrier (the concatenation of the MCC and MNC prefixes provide unambiguous identification of the carrier network)
- MSIN (Mobile Station Identification Number) — the remaining digits identify the specific device within the carrier's network

IMSI/MAC Filtering

With EAP-SIM protocol in use, authentication is accomplished by a RADIUS server. Using the Wm interface, the Oracle Communications Session Border Controller passes the received IMSI identity to the RADIUS server. In order to minimize server processing, the OCSBC provides users with the optional ability to compile IMSI prefix white lists that filter identities presented for RADIUS authentication. White lists are inclusive in that only those identities matching list contents are granted RADIUS access; non-matching identities are summarily rejected by the OCSBC. The white lists contain numeric strings or simple regular expressions that identify blocks of subscribers eligible for access to the RADIUS server.

These strings are interpreted as either an IMSI prefix or as a MAC address. White lists now contain either IMSI or MAC identifiers. Identifiers are constructed using the digits 0 through 9 , any hexadecimal digit, and the ^ wild-card character, which specifies any single base-10 or base-16 digit. Each identifier one or one or more subscribers eligible for authentication.

Sample identifiers are as follows:

- 744 matches the country of Paraguay
- 74401 matches a specific Paraguayan carrier (Hola Paraguay S.A.)
- 7440^ matches all current Paraguayan carriers (74401, 74402, 74404, and 74405)

Configure IMSI/MAC White Lists

The ike-access-control configuration element defines a white list that filters IMSI or MAC identities presented by remote endpoints during the authentication process. Only those identities matching the literal or regular expressions contained within the white list are forwarded via the Wm interface to a RADIUS server for authentication.

1. Access the **ike-access-control** configuration element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# security
   ORACLE(security)# ike
   ORACLE(ike)# ike-access-control
   ORACLE(ike-access-control)#
   ```

2. **name**—Provide a unique identifier.

   ```
   ORACLE(ike-access-control)# name white_01
   ```

3. **state**—Enable access control.
4. **identifier**—Provide one or more MCC or MCC/MNC match patterns for IMSI-based whitelisting.

   This identifier, a literal string, matches the Russian Federation.

   ```
   ORACLE(ike-access-control)# identifier 250
   ```

This identifier, which uses the wildcard symbol (^) signifying any single digit within the range 0 through 9, matches the continental United States.

```
ORACLE(ike-access-control)# identifier 31^
```

This identifier, a double-quote delimited list of prefixes separated by spaces, matches T-Mobile United States networks.

```
ORACLE(ike-access-control)# identifier "26201 26206"
```

This identifier, a double-quote delimited list of prefixes separated by spaces, matches Verizon Wireless United States networks.

```
ORACLE(ike-access-control)# identifier "310004 310012"
```

For MAC-based whitelisting, the following double-quote delimited list identifies three specific MAC addresses.

```
ORACLE(ike-access-control)# identifier "0123456789AB 6789912345BF
DA2345918290"
```

> **Note:**
>
> Do not configure an empty white list. Assigning an empty white list to an IKEv2 interface results in authentication failure for all presented identities.

5. Type **done** to save your configuration.
6. If necessary, configure additional **ike-access-control** configuration elements.

Configure Black Lists

A black list is provisioned with a femtocell's EAP identity, taking the form `<MAC ID>@cellID.serviceProvider.com` and denying authentication for such femtocells trying to establish IKE/IPsec tunnels. Black lists are only applicable for femtocell clients doing EAP authentication to the OCSBC and are not applicable for clients doing password-based or certificate-based authentication.

1. Access the **ike-access-control** configuration element.

```
ORACLE# configure terminal
ORACLE(configure)# security
ORACLE(security)# ike
ORACLE(ike)# ike-access-control
ORACLE(ike-access-control)#
```

2. **name**—Provide a unique identifier.

```
ORACLE(ike-access-control)# name black_01
```

3. **state**—Enable access control.
4. **blacklisted-dentifiers**—Provide one or more MAC-based match patterns for MAC-address-based black lists.

   The following double-quote delimited list identifies three specific MAC addresses whose authentication is summarily rejected.

   ```
   ORACLE(ike-access-control)# blacklisted-dentifiers "0123456789AB
   6789912345BF DA2345918290"
   ```

   This identifier, which uses the wildcard symbol (^) signifying any single hexadecimal digit, specifies two ranges of contiguous MAC addresses.

   ```
   ORACLE(ike-access-control)# blacklisted-dentifiers "0123456789A^,
   ^123456789AB"
   ```

   For IMSI-based black lists, this example uses a double-quote delimited list of prefixes separated by spaces, to match Verizon Wireless United States networks.

   ```
   ORACLE(ike-access-control)# blacklisted-dentifiers "310004 310012"
   ```

   > **Note:**
   >
   > Do not configure an empty black list. Assigning an empty black list to an IKEv2 interface results in authentication eligibility for all presented identities.

5. Tyoe **done** to save your configuration.
6. If necessary, configure additional **ike-access-control** configuration elements.

Assign a White List or Black List to an IKEv2 Interface

1. Access the **ike-interface** configuration element.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# security
   ORACLE(security)# ike
   ORACLE(ike)# ike-interface
   ORACLE(ike-interface)#
   ```

2. Select the **ike-interface** object to edit.

   ```
   ORACLE(ike-interface)# select
   <address>:

   ORACLE(ike-interface)#
   ```

3. **access-control-name**—Identify the white list or black list assigned to the current interface.

```
ORACLE(ike-interface)# access-control-name white_01
```

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

White List/Black List Interaction

White lists and black lists may or may not be assigned to the IKEv2 interfaces. The following rules are used to support implementation of both list types.

1. If neither a white list nor a black list are assigned to an IKEv2 interface, all EAP authentication requests are forwarded to a RADIUS authentication server for final determination.
2. If only a white list is assigned to an IKEv2 interface, the incoming EAP identity is be checked against that white list. If the EAP identity is contained in the white list, the authentication request is forwarded to a RADIUS authentication server for final determination. If the EAP identity is absent, authentication is denied.
3. If only a black list is assigned to an IKEv2 interface, the incoming EAP identity is checked against that black list. If the EAP identity is contained in the black list, authentication is denied. If the EAP identity is absent, the authentication request is forwarded to a RADIUS authentication server for final determination..
4. If both a white list and a black list are assigned to an IKEv2 interface, the OCSBC checks both the white and the black list for incoming EAP identity.

   If the EAP identity is contained in the white list, and absent from the black list, the authentication request is forwarded to a RADIUS authentication server for final determination.

   If the EAP identity is contained in the black list and absent from the white list, authentication is rejected.

   If the EAP identity is present in both the lists, the black list takes priority. Consequently, authentication is rejected. This situation will have been previously reported by the **verify-config** ACLI command.

   If the EAP identity is absent from both the lists, the whate list takes priority. Consequently, since the EAP identity is not contained in the white list the authentication is denied.

Viewing Security IKE Statistics

Via the **show security ike statistics** ACLI command, you can view statistics derived from the IKEAuthIDError and BlacklistIKEAuthIDError counters, containing the number of authentication denials due to both white and black list filtering.

For detailed information on the **show security ike statistics** ACLI command, see "Appendix B: ACLI Quick Reference" of this guide.

Threshold Crossing Alert Configuration

Threshold Crossing Alerts (TCAs) monitor specific MIB variables or counters, and generate SNMP traps when object values cross defined thresholds. Three types of TCAs are supported:

- IKE Failed Authentication (monitors IKE negotiation counters)
- IPsec Tunnel Removal (monitors IPsec tunnel counters)
- Dead Peer Detections (monitors DPD protocol counters)

Threshold levels, listed in order of increasing importance are clear, minor, major, and critical. Each threshold level is user-configurable and is accompanied by a associated reset-counter, also user-configurable, which prevents the issue of extraneous SNMP traps when a counter is bouncing across threshold values.

A threshold crossing event occurs when the associated counter value rises above the next-highest threshold value, or when the associated counter value falls below the next-lowest reset-threshold value. An SNMP trap, raising the alert level, is generated as soon as the counter value exceeds the next-highest threshold. An SNMP trap, lowering the alert level, occurs only during a check period when the TCA examines all counter values. Such check periods occur at 100 second intervals.

The following scenario illustrates TCA operations. The sample TCA, ike-tca-group, monitors the count of dead IKEv2 peers. Threshold and reset values are shown. A minor alarm threshold and its associated reset threshold have not been configured.

```
nameike-tca-group
tca-typeike-dpd
critical100
reset-critical90
major80
reset-major50
minor0
reset-minor0
```

t=time

t=0 ike-dpd counter= 30 ike-dpd alert level=clear

t=1 ike-dpd counter= 60 ike-dpd alert level=clear

t=2 ike-dpd counter= 80 ike-dpd alert level=major trap sent

t=3 ike-dpd counter= 95 ike-dpd alert level=major

t=4 ike-dpd counter=100 ike-dpd alert level=critical trap sent

t=5 ike-dpd counter=120 ike-dpd alert level=critical

t=6 ike-dpd counter= 99 ike-dpd alert level=critical

t=7 ike-dpd counter= 90 ike-dpd alert level=major trap sent

t=8 ike-dpd counter= 60 ike-dpd alert level=major

t=9 ike-dpd counter= 0 ike-dpd alert level=clear trap sent

Use the following procedure to configure TCAs.

1. From superuser mode, use the following command sequence to access threshold-crossing-alert-group configuration mode. While in this mode, you configure threshold-crossing-alert-group configuration elements.

   ```
   ORACLE# configure terminal
   ORACLE(configure)# system
   ORACLE(system)# threshold-crossing-alert-group
   ORACLE(threshold-crossing-alert-group)#
   ```

2. Use the **name** parameter to provide a unique identifier for this threshold-crossing-alert-group instance.

   name enables the creation of multiple threshold-crossing-alert-group instances.

   ```
   ORACLE(threshold-crossing-alert-group)# name ikeTCA
   ORACLE(threshold-crossing-alert-group)#
   ```

3. Use the **threshold-crossing-alert** parameter to enter threshold-crossing-alert configuration mode. While in this mode, you create specific TCA types and associated values.

   ```
   ORACLE(threshold-crossing-alert-group)# threshold-crossing-alert
   ORACLE(threshold-crossing-alert)#
   ```

4. Use the **type** parameter to specify the TCA type.

   Supported values are:

   - ike-failed-auth — (the default) tracks authentication failures
   - ipsec-tunnel-removal — tracks the destruction of IPsec tunnels
   - ike-dpd — tracks the detection of dead DPD peers

   ```
   ORACLE(threshold-crossing-alert)# type ike-dpd
   ORACLE(threshold-crossing-alert)#
   ```

5. Use the **critical** parameter to specify the critical threshold level.

   The default value (0) indicates that the threshold is not configured.

   ```
   ORACLE(threshold-crossing-alert)# critical 100
   ORACLE(threshold-crossing-alert)#
   ```

6. Use the **reset-critical** parameter to specify the value at which the critical level is replaced with the next lowest configured threshold level (major, minor, or clear, depending on configuration values).

   The default value (0) indicates that the threshold is not configured.

   ```
   ORACLE(threshold-crossing-alert)# reset-critical 90
   ORACLE(threshold-crossing-alert)#
   ```

7. Use the **major** parameter to specify the major threshold level.

The default value (0) indicates that the threshold is not configured.

```
ORACLE(threshold-crossing-alert)# major 80
ORACLE(threshold-crossing-alert)#
```

8. Use the **reset-major** parameter to specify the value at which the major level is replaced with the next lowest configured threshold level (minor or clear, depending on configuration values).

   The default value (0) indicates that the threshold is not configured.

   ```
   ORACLE(threshold-crossing-alert)# reset-major 50
   ORACLE(threshold-crossing-alert)#
   ```

9. Use the **minor** parameter to specify the minor threshold level.

   The default value (0) indicates that the threshold is not configured.

   ```
   ORACLE(threshold-crossing-alert)# minor 0
   ORACLE(threshold-crossing-alert)#
   ```

10. Use the **reset-minor** parameter to specify the value at which the minor level is replaced with the next lowest configured threshold level (clear).

    The default value (0) indicates that the threshold is not configured.

    ```
    ORACLE(threshold-crossing-alert)# reset-minor 0
    ORACLE(threshold-crossing-alert)#
    ```

11. If required, repeat Steps 4 through 10 to add other TCA types to the current threshold-crossing-alert-group configuration element.

    The threshold-crossing-alert-group configuration element can contain a maximum of three individual threshold-crossing-alerts, one of each supported type.

12. Use **done**, **exit**, and **verify-config** to complete configuration of the threshold-crossing-alert-group configuration element.

13. If necessary, repeat Steps 1 through 12 to configure additional threshold-crossing-alert-group configuration elements.

14. From superuser mode, use the following command sequence to access ike-config configuration mode. While in this mode, you configure IKEv2 interface parameters.

    ```
    ORACLE# configure terminal
    ORACLE(configure)# security
    ORACLE(security)# ike
    ORACLE(ike)# ike-interface
    ORACLE(ike-interface)#
    ```

15. Use the optional **threshold-crossing-alert-group-name** parameter to assign an existing threshold-crossing-alert-group configuration element to this IKEv2 interface.

    ```
    ORACLE(ike-interface)# threshold-crossing-alert-group-name ikeTCA
    ORACLE(ike-interface)#
    ```

16. Use **done**, **exit**, and **verify-config** to complete configuration of the TCA.

IKEv2 Interface Management

The following two sections provide details on available counters that gather usage and error data related to IKEv2/IPsec operations on the Oracle Communications Session Border Controller.

The first section, IKEv2 Protocol Operations, describes a series of 32-bit counters that report interface-specific data on various protocol transactions. Protocol operations counter values are available with SNMP, through the ACLI **show security ike statistics** command, and can also be obtained by subscription to the ike_stats HDR group.

The second section, IKEv2 Negotiation Errors, describes a series of 32-bit counters that report interface-specific errors encountered during IKEv2 negotiations. Negotiation errors counter values are also available with SNMP, through the ACLI **show security ike statistics** command, and can also be obtained by subscription to the ike-stats HDR group.

The third section, RADIUS Protocol Operations, describes a series of 32-bit counters that report RADIUS-server-specific data. RADIUS protocol operations counter values are also available with SNMP, through the ACLI **show radius** command, and can also be obtained by subscription to the radius-stats HDR group.

The final section, Diameter Protocol Operations, describes a series of 32-bit counters that report Diameter-server-specific data. Diameter protocol operations counter values are also available with SNMP, and can also be obtained by subscription to the diameter-stats HDR group.

IKEv2 Protocol Operations

The SNMP MIB is formed by appending the value in the SNMP MIB Ending column to 1.3.6.1.4.1.9148.3.9.1.9.X (apSecurityIkeInterfaceInfoTable), where X specifies the interface index. For example, the SNMP MIB for the Current Child SA Pairs is 1.3.6.1.4.1.9148.3.9.1.9.X.33, where X specifies the interface index.

> **Note:**
>
> The range for all 32-bit counters is 0 to 4294967295.

Name	Description	Type	SNMP MIB Ending
Current Child SA Pairs	The number of current child IPsec SA pairs on the interface. As each IPsec tunnel requires two unidirectional SAs, this number equals the current number of tunnels on the interface. Note that this count is available through both an ACLI show command and an SNMP GET operation.	gauge	.33

Name	Description	Type	SNMP MIB Ending
Maximum Child SA Pairs	The largest number of child IPsec SA pairs on the interface since this counter was last reset. As each IPsec tunnel requires a single SA pair, this value equates to the largest number of tunnels on the interface.	gauge	
Last Reset Timestamp	The time that this interface was last reset -- expressed as a UNIX timestamp containing the number of seconds since January 1, 1970.	UNIX timestamp	
Child SA Request	The number of requests to add a child SA pair that were received on the interface. These requests include IPsec SA rekey requests.	counter	.1
Child SA Success	The number of requests to add a child SA pair that were successfully completed on the interface. These successes include new children created by IPsec SA rekeys.	counter	.2
Child SA Failure	The number of requests to add a child SA pair that were not successfully completed on the interface. These failures include unsuccessful IPsec SA rekeys.	counter	.3
Child SA Delete Requests	The number of requests to delete a child SA pair that were received on the interface. These requests include deletion requests associated with IPsec SA rekeys.	counter	.4
Child SA Delete Success	The number of requests to delete a child SA pair that were successfully completed on the interface. These successes include children deleted by IPsec SA rekeys.	counter	.5
Child SA Delete Failure	The number of requests to delete a child SA pair that were not successfully completed on the interface. These failures include unsuccessful deletions associated IPsec SA rekeys.	counter	.6
Child SA Rekey	The number of child IPsec rekey exchanges transacted on the interface.	counter	.7
Initial Child SA Establishment	The number of initial child SA pair establishments, in other words, the number of successful IKE_AUTH exchanges transacted on the interface.	counter	.8
DPD Received Port Change	The number of DPD messages received on the interface that contained a port change from the previously received message. The port change indicates that the IKEv2 has moved to another port, or that an intervening NAT device has changed port mapping. These actions do not impact SA functions.	counter	.9
DPD Received IP Change	The number of DPD messages received on the interface that contained an IP address change from the previously received message.	counter	.10

Name	Description	Type	SNMP MIB Ending
DPD Response Received	The number of DPD ACK responses received on the interface. An ACK is sent by an IKEv2 peer in response to an R-U-THERE issued by the Oracle Communications Session Border Controller. A successful R-U-THERE/ACK exchange establishes availability on the remote IKEv2 peer.	counter	.11
DPD Response Not Received	The number of R-U-THERE messages transmitted on the interface that were not acknowledged within the DPD allowed interval.	counter	.12
DPD Received	The number of all DPD protocol messages received on the interface.	counter	.13
DPD Retransmitted	The number of R-U-THERE messages that were re-transmitted because the original R-U-THERE message was not acknowledged.	counter	.14
DPD Sent	The number of R-U-THERE messages that were sent across the interface, to include retransmitals.	counter	.15
IKE SA Packets Sent	The number of IKEv2 SA packets sent across the interface.	counter	.16
IKE SA Packets Received	The number of IKEv2 SA packets received across the interface.	counter	.17
IKE SA Packets Dropped	The number of IKEv2 SA packets dropped by the interface.	counter	.18
Authentication Failures	The number of authentication failures that occurred after the purported identity of the remote IKEv2 peer was ascertained.	counter	.19
IKE Message Errors	The number of otherwise uncharacterized IKEv2 message errors.	counter	.20
Authentication ID Errors	The number of errors that occurred during the identification stage of the authentication process.	counter	.21
Certificate Status Requests	The number of certificate status requests sent across the interface to an OCSP responder.	counter	.22
Certificate Status Success	The total number of OCSP successes, that is the number of OCSP requests that generated a good status from an OCSP responder.	counter	.23
Certificate Status Fail	The total number of OCSP failures, to include unacknowledged OCSP requests and those requests that generated a revoked or unknown response from an OCSP responder.	counter	.24
DDoS Sent	The number of suspicious, and possibly malicious, endpoints reported by the interface-specific DDoS process (if configured as described in the IKEv2 DDoS Protection section of the Oracle Communications Session Border Controller Essentials guide).	counter	.25
DDoS Received	The number of suspicious, and possibly malicious, endpoints reported by statically provisioned deny lists (as described in SIP Signaling Services and Security chapters of the ACLI Configuration Guide).	counter	.26

Name	Description	Type	SNMP MIB Ending
IKE Message Retransmissions	The total number of IKEv2 message re-transmissions.	counter	.27
SA Init Messages Received	The total number of IKEv2 message re-transmissions.	counter	.28
SA Init Message Sent	The total number of IKEv2 message re-transmissions.	counter	.29
SA Establishment Attempts	The total number of IKEv2 message re-transmissions.	counter	.30
SA Establishment Success	The total number of IKEv2 SA successfully established on the IKEv2 interface.	counter	.31
Tunnel Rate	Specifies the tunnel establishment rate, in terms of tunnels created per second. Note that this count is available through both an ACLI show command and an SNMP GET operation.	gauge	.32

IKEv2 Negotiation Errors

The SNMP MIB is formed by appending the value in the SNMP MIB Ending column to 1.3.6.1.4.1.9148.3.9.1.3.X (apSecurityIkeInterfaceStatsEntry), where X specifies the interface index. For example, the SNMP MIB for the CPU Overload Errors is 1.3.6.1.4.1.9148.3.9.1.3.X.3, where X specifies the interface index.

Name	Description	SNMP MIB Ending
CPU Overload Errors	The number of IKEv2 requests that were rejected because of CPU load constraints.	.3
Init Cookie Errors	The number of all IKEv2 exchanges that failed because of faulty Security Parameter Index (SPI) values. SPIs provide a local SA identifier and are exchanged between IKEv2 peers in the common IKEv2 header and in Notify Payloads.	.4
Auth Errors	The number of failed IKE_AUTH exchanges, regardless of the specific reason for failure.	.5
EAP Access Request Errors	The number of authentication failures that occur ed during the EAP access phase.	.6
EAP Access Challenge Errors	The number of authentication failures that occur ed during the EAP challenge phase.	.7
TS Errors	The number of CREATE_CHILD_SA exchanges that failed because of faulty TS payload contents, or failure on the part of the remote peers to negotiate the offered traffic selectors.	.8
CP Errors	The number of IKE_AUTH and/or CREATE_CHILD_SA exchanges that failed because of faulty, unsupported, or unknown Configuration Payload contents.	.9
IKE Errors	The number of IKE_SA_INIT and/or CREATE_CHILD_SA exchanges that failed because of faulty, unsupported, or unknown Key Exchange Payload contents.	.10

Name	Description	SNMP MIB Ending
Proposal Errors	The number of failed negotiations that resulted from the inability to reconcile crytographic proposals contained in the Security Association Payloads exchanged by IKEv2 peers. Security Association Payloads are exchanged during the IKE_SA_INIT, IKE_AUTH, and CREATE_CHILD_SA stages.	.11
Syntax Errors	The number of failed negotiations, of any type, resulting from otherwise uncharacterized errors.	.12
Critical Payload Errors	The number of failed negotiations that resulted from the presence of a Critical flag in a payload that could not be parsed, or was not supported. IKEv2 adds a critical flag to each payload header for further flexibility for forward compatibility. If the critical flag is set and the payload type is unrecognized, the message must be rejected and the response to the IKE request containing that payload MUST include a Notify payload UNSUPPORTED_CRITICAL_PAYLOAD, indicating an unsupported critical payload was included. If the critical flag is not set and the payload type is unsupported, that payload must be ignored.	.13

RADIUS Protocol Operations

The SNMP MIB is formed by appending the value in the SNMP MIB Ending column to 1.3.6.1.4.1.9148.3.18.1.1.1 (aapRadiusServerStatsEntry). For example, the SNMP MIB for the Server Roundtrip Time is 1.3.6.1.4.1.9148.3.18.1.1.1.3.

Name	Description	SNMP MIB Ending
Server Roundtrip Time	Contains the average round trip time for a response from this RADIUS server.	.3
Server Malformed Access Response	Contains the number of malformed access responses received on this RADIUS server.	.4
Server Access Requests	Contains the number of access requests received on this RADIUS server.	.5
Server Disconnect Requests	Contains the number of disconnect requests received on this RADIUS server.	.6
Server Disconnect ACKS	Contains the number of acknowledged disconnects on this RADIUS server.	.7
Server Disconnect NACKS	Contains the number of unacknowledged disconnects on this RADIUS server.	.8
Server Bad Authenticators	Contains the number of authentication rejections on this RADIUS server.	.9
Server Access Retransmissions	Contains the number of access retransmitals on this RADIUS server.	.10
Server Access Accepts	Contains the number of successful authentications on this RADIUS server.	.11

Name	Description	SNMP MIB Ending
Server Timeouts	Contains the number of Response timeouts on this RADIUS server.	.12
Server Access Rejects	Contains the number of unsuccessful authentications on this RADIUS server.	.13
Server Unknown PDUTypes	Contains the number or unknown/unreadable PDUs received by this RADIUS server.	.14
Server Access Challenges	Contains the number of Access Challenges on this RADIUS server.	.15

Diameter Protocol Operations

The SNMP MIB is formed by appending the value in the SNMP MIB Ending column to 1.3.6.1.4.1.9148.3.13.1.1.2.2.X (apDiamInterfaceStatsTable), where X specifies the diameter server index. For example, the SNMP MIB for the Diameter Messages Sent is 1.3.6.1.4.1.9148.3.13.1.1.2.2.X.3, where X specifies the diameter server index.

Name	Description	SNMP MIB Ending
Diameter Messages Sent	Contains the number of messages sent by this Diameter server.	.3
Diameter Messages Sent Failed	Contains the number of unacknowledged messages sent by this Diameter server.	.4
Diameter Messages Resent	Contains the number of messages re-transmitted to this Diameter server.	.5
Diameter Messages Received	Contains the number of messages received by this Diameter server.	.6
Diameter Messages Processed	Contains the number of messages processed by this Diameter server.	.7
Diameter Connection Timeouts	Contains the number of connection timeouts on the Diameter server.	.8
Diameter BadState Drops	Contains the number of packets dropped because of faulty state on the Diameter server.	.9
Diameter BadType Drops	Contains the number of packets dropped because of faulty type on the Diameter server.	.10
Diameter BadID Drops	Contains the number of packets dropped because of faulty ID on the Diameter server.	.11
Diameter AuthFail Drops	Contains the number of failed authentications on the Diameter server.	.12
Diameter Invalid Peer Messages	Contains the number of client messages that could not be parsed on the Diameter server.	.13

ACLI Show Commands

ACLI **show** commands

- display and reset IKEv2 performance and error counters
- display IKEv2 SA data
- display IKEv2 TCA data

Performance and Error Counters

Three ACLI commands display and reset IKEv2 performance and error counters.

Use the **show security** command to display performance and error counters for a specified IKEv2 interface, or for all IKEv2 interfaces.

```
ORACLE# show security 192.169.204.15
```

with a specified interface, displays performance and error counters for the target interface

```
ORACLE# show security all
```

with all, displays performance and error counters for all IKEv2 interfaces

Use the **reset ike-stats** command to reset (set to 0) performance and error counters for a specified IKEv2 interface, or for all IKEv2 interfaces.

```
ORACLE# reset ike-stats 192.169.204.15
```

with a specified interface, resets performance and error counters for the target interface

```
ORACLE# reset ike-stats all
```

with all, resets performance and error counters for all IKEv2 interfaces

Use the **reset ike-mib** command to reset (set to 0) MIB-based error counters for all IKEv2 interfaces.

```
ORACLE# reset ike-mib
```

re-sets the MIB-based error counters for all IKEv2 interfaces

IKEv2 and Child SAs

Use the **show security** command with optional arguments to display IKEv2 and child SA information to include:

- IP address and port of remote end-point
- intervening NAT device (yes | no)
- local IP address
- tunnel state (up | down)
- initiator cookie
- responder cookie
- remote inner (tunnel) IP address

- incoming/outgoing Security Parameter Indexes (SPI) of the child SA

```
ORACLE# show security sad ike-interface 192.169.204.15
```

with a specified interface address, displays SA information for a single IKEv2 interface

```
ORACLE# show security sad ike-interface all
```

with all, displays SA information for all IKEv2 interfaces

```
ORACLE# show security sad ike-interface all
Displaying the total (4321) number of entries may take long and could
affect system performance.
Continue? [y/n]?: y
Peer: 6.0.0.36:500 (NAT: No) Host: 172.16.101.2 State: Up
    IKE Cookies: 0x23e71b73d5a10c58[I] 0xd2017a6fb84a4fa6[R]
    Child Peer IP: 101.0.0.36:0 Child SPI: 4236760138[I] 1721373661[O]
Peer: 6.0.0.28:500 (NAT: No) Host: 172.16.101.2 State: Up
    IKE Cookies: 0xf64d031d32525730[I] 0xcea2d5ae3c91050f[R]
    Child Peer IP: 101.0.0.28:0 Child SPI: 3632387333[I] 1421117246[O]
Peer: 6.0.0.9:500 (NAT: No) Host: 172.16.101.2 State: Up
    IKE Cookies: 0x84ec95a1cd0a4c5d[I] 0x1b61b385c4e627b4[R]
    Child Peer IP: 101.0.0.9:0 Child SPI: 2432742837[I] 3872387177[O]
Peer: 6.0.0.25:500 (NAT: No) Host: 172.16.101.2 State: Up
    IKE Cookies: 0x541b2651e88c9368[I] 0xdc393a61af6dc909[R]
    Child Peer IP: 101.0.0.25:0 Child SPI: 785656546[I] 148357787[O]
Peer: 6.0.0.27:500 (NAT: No) Host: 172.16.101.2 State: Up
    IKE Cookies: 0x3ba43c5c685e37e6[I] 0x7bfa6f0781dce1a8[R]
    Child Peer IP: 101.0.0.27:0 Child SPI: 767765646[I] 3797275291[O]
Peer: 6.0.0.22:500 (NAT: No) Host: 172.16.101.2 State: Up
    IKE Cookies: 0x925e540ecbd58dbb[I] 0x7e1101371a5a5823[R]
    Child Peer IP: 101.0.0.22:0 Child SPI: 787745714[I] 876969665[O]
Peer: 6.0.0.2:500 (NAT: No) Host: 172.16.101.2 State: Up
    IKE Cookies: 0xda0f568684ba5e2c[I] 0x74c533da2fd29901[R]
    Child Peer IP: 101.0.0.2:0 Child SPI: 3884481109[I] 1862217459[O]
Peer: 6.0.0.7:500 (NAT: No) Host: 172.16.101.2 State: Up
    IKE Cookies: 0x6166bac4438f3ca7[I] 0x71d1049a0f8520f4[R]
    Child Peer IP: 101.0.0.7:0 Child SPI: 2798332266[I] 2789214337[O]
Peer: 6.0.0.15:500 (NAT: No) Host: 172.16.101.2 State: Up
    IKE Cookies: 0x0e060701115069bf[I] 0x2e69adbf15438000[R]
    Child Peer IP: 101.0.0.15:0 Child SPI: 713005957[I] 1985608540[O]
Continue? [y/n]?: y
...
...
```

Use **show security** with the peer address obtained by the previous command to display more detailed information regarding a specific tunnel to include:

- IKE version
- Diffie Hellman group
- the IKE SA hash algorithm
- the IKE SA message authentication code algorithm

- the IKE SA encryption algorithm
- seconds since SA creation
- SA lifetime in seconds
- remaining lifetime in seconds
- IPsec operational mode (tunnel | transport)
- IPsec security protocol (AH |ESP)
- IPsec authentication protocol (SHA1 | MD5 | any)
- IPsec encryption protocol (AES | 3DES | null| any)

```
ORACLE# show security sad ike-interface <ipAddress> peer <ipAddress>
ORACLE# show security sad ike-interface 172.16.101.2 peer 6.0.0.36:500

IKE SA:

    IKE Version : 2
    Tunnel State : Up
    Last Response [Seconds] : 212
    AAA Identity :
    NAT : No

    IP Addresses [IP:Port]
        Peer : 6.0.0.36:500
        Server Instance : 172.16.101.2:500

    Cookies
        Initiator : 0x23e71b73d5a10c58
        Responder : 0xd2017a6fb84a4fa6

    Algorithms
        DH Group : 2
        Hash : HMAC-SHA1
        MAC : SHA1-96
        Cipher : 3DES

    SA Times [Seconds]
        Creation : 141
        Expiry : 86400
        Remaining : 86188

IPSec SA:

    IP Addresses [IP:Port]
        Destination : 101.0.0.36:0
        Source : 172.16.101.2:0

    SPI
        Outbound : 1721373661
        Inbound : 4236760138

    Algorithms
        Mode : TUNNEL
```

```
        Protocol : ESP
        Authentication : SHA1
        Encryption : AES

    Traffic Selectors [Start IP - End IP]
        Destination : 101.0.0.36 - 101.0.0.36
        Source : 172.16.101.2 - 172.16.101.2
```

TCA Counters

An ACLI command is provided to display TCA information.

```
ORACLE# show security ike threshold-crossing-alert <ipAddress> || all
```

with a specified IPv4/IPv6 interface address, displays TCA information for the specified IKEv2 interface, otherwise displays TCA information for all IKEv2 interfaces

```
ORACLE# show security ike threshold-crossing-alert all
ORACLE# show security ike threshold-crossing-alert all
IKE Threshold Crossing Alerts
tca-type: ike-auth-failure
           reset            reset                          reset
critical     critical    major     major       minor      minor
---------- ---------- ---------- ---------- ---------- ----------
       40         30         25         24         12          1
current value:
   Window Total Maximum
        0    0        0
current level: clear

tca-type: ipsec-tunnel-removal
             reset               reset                    reset
critical     critical    major     major       minor      minor
---------- ---------- ---------- ---------- ---------- ----------
        0          0         10          5          0          0
current value:
     Window Total Maximum
          0     0       0
current level: clear
```

TCA Traps

TCAs generate SNMP traps to report crossing of threshold levels, or to clear threshold levels. For a detailed description of these traps, see "TCA Traps" in "Appendix A: MIB SNMP Quick Reference".

Historical Data Records

Various statistical counts are available as comma separated values (CSV) Historical Data Record (HDR) files. HDR files are specified and pushed to an accounting server as described in the Overview chapter of the 4000 C-Series Historical Data Recording (HDR) Resource Guide.

IKEv2 Interface HDR

CSV header fields for IKEv2 Interface HDRs are listed below.

IKEv2 Interface HDR	Type
TimeStamp	Integer
Interface	IP Address
Current Child SA Pairs	Counter
Maximum Child SA Pairs	Counter
Last Reset TimeStamp	Integer
Child SA Requests	Counter
Child SA Success	Counter
Child SA Failure	Counter
Child SA Delete Request	Counter
Child SA Delete Success	Counter
Child SA Delete Failure	Counter
Child SA Rekey	Counter
Initial Child SA Establishment	Counter
DPD Received Port Change	Counter
DPD Received IP Change	Counter
DPD Response Received	Counter
DPD Response Not Received	Counter
DPD Received	Counter
DPD Retransmitted	Counter
DPD Sent	Counter
IKE SA Packets Sent	Counter
IKE SA Packets Received	Counter
IKE SA Packets Dropped	Counter
Authentication Failures	Counter
IKE Message Errors	Counter
Authentication ID Errors	Counter
Certificate Status Requests	Counter
Certificate Status Success	Counter
Certificate Status Fail	Counter
DDoS Sent	Counter
DDoS Received	Counter
IKE Message Retransmissions	Counter
Tunnel Rate	Counter
Child SA Pair	Guage
IKE SA INIT Messages Received	Counter
IKE SA INIT Messages Sent	Counter
IKE SA Establishment Attempts	Counter
IKE SA Establishment Success	Counter
IKE CPU Overload Error	Counter
IKE init Cookie Error	Counter
IKE EapAccessRequestError	Counter
IKE EapAccessChallengeError	Counter

IKEv2 Interface HDR	Type
IKE TS Error	Counter
IKE CP Error	Counter
IKE KE Error	Counter
IKE Proposal Error	Counter
IKE Syntax Error	Counter
IKE Critica; Payload Error	Counter

RADIUS HDR

CSV header fields for RADIUS HDRs are listed below.

IKEv2 Interface HDR	Type
Time Stamp	Integer
RADIUS Sever IP Address	IP Address
RADIUS Server Port	Port Address
Round Trip Time	Counter
Malformed Access Response	Counter
Access Requests	Counter
Disconnect Requests	Counter
Disconnect ACKs	Counter
Bad Authenticators	Counter
Access Retransmissions	Counter
Access Accepts	Counter
Timeouts	Counter
Access Rejects	Counter
Unknown PDU Types	Counter
Access Challenges	Counter

Diameter HDR

CSV header fields for Diameter HDRs are listed below.

IKEv2 Interface HDR	Type
Time Stamp	Integer
Diameter Sever IP Address	IP Address
Diameter Server Port	Port Address
Messages Sent	Counter
Messages Sent Failed	Counter
Messages Resent	Counter
Messages Received	Counter
Messages Processed	Counter
Connection Timeouts	Counter
Bad State Drops	Counter
Bad Type Drops	Counter
Bad ID Drops	Counter

IKEv2 Interface HDR	Type
Auth Failed Drops	Counter
Invalid Peer Messages	Counter