Oracle® Communications Session Border Controller

Administrative Security Essentials Guide
Release S-CZ7.2.0

October 2015

ORACLE

Notices

Contents

About this guide

This guide explains support for a Administrative Security License (Admin Security) and Admin Security APC, which provide a suite of applications and tools providing enhanced, more secure system access, monitoring, and management. All functionality described in this guide requires an active Admin Security or Admin Security APC license. Users of the Oracle Communications Session Border Controller s without either Admin Security license can safely ignore this guide.

Specific topics covered in this guide include

- Access
- Audit Log
- IKEv2
- TACACS+
- License Issues

Audience

This guide is written for network administrators and architects, and provides information about the SBC implementation. Supporting, related material is available in the ACLI Configuration Guide. Please refer to that document as needed.

Related Documentation

The following table describes related documentation for the Oracle Communications Session Border Controller.

Document Name	Document Description
Acme Packet 4500 Hardware Installation Guide	Contains information about the components and installation of the AP4500.
Acme Packet 3820 Hardware Installation Guide	Contains information about the components and installation of the AP 3800.
Acme Packet 6300 Hardware Installation Guide	Contains information about the components and installation of the AP 6300.
Acme Packet 6100 Hardware Installation Guide	Contains information about the components and installation of the AP 6100.
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 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.
MIB Reference Guide	Contains information about Management Information Base (MIBs), Acme Packet'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

Document Name	Document Description
	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 accounting.
HDR Resource Guide	Contains information about the Oracle Communications Session Border Controller's Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Administrative Security Essentials	Contains information about the Oracle Communications Session Border Controller's support for its Administrative Security license.
Security Guide	Contains information about security considerations and best practices from a network and application security perspective for the Oracle Communications Session Border Controller family of products.

Revision History

The following table describes updates to this guide.

Date	Description
June, 2014	Initial Release
October 2015	Adds the chapter on TACACS+ in order to complete the list of features that affect Admin Security.
March 2016	Edits the sections on Admin Security and Admin Security ACP to clarify the differences between the features, and to explain the ramifications of uninstalling the licenses

1

Access

Administrative Security

This section describes implications of installing and deleting the Admin Security license and the Admin Security ACP license on an Oracle Communications Session Border Controller (SBC).

These licenses enable the various security enhancements described in this document. In the absence of an Admin Security or Admin Security ACP license, these enhancements are not available.

As with any other license, an **activate-config** command must be executed after license installation for all changes to take effect. Certain ACLI aspects, such as login and password change prompts, change immediately after license installation.

These two licenses relate as follows:

1. Both licenses can exist together or separately on an SBC.
2. Removal of either or both licenses does not make available the protected areas of the system. This ensures that a system cannot be compromised by simply removing the Admin Security license(s).

Note: The Admin Security or the Admin Security ACP feature sets are not intended for all customer use. Consult your Oracle representative to understand the ramifications of enabling these features.

Note: Once the Admin Security or the Admin Security with ACP entitlement is provisioned, it can not be removed from the system in the field; your chassis must be returned to Oracle for replacement.

Admin Security Features for either license:

- telnet access is denied
- FTP access is denied
- history log access is denied
- shell access is denied
- additional password policy features are enabled

Additional Security features available with the Admin Security license:

- EMS (Element Management System) access is blocked
- ACP (Acme Control Protocol) is blocked

Additional Security features available with the Admin Security ACP license

- EMS (Element Management System) access is open

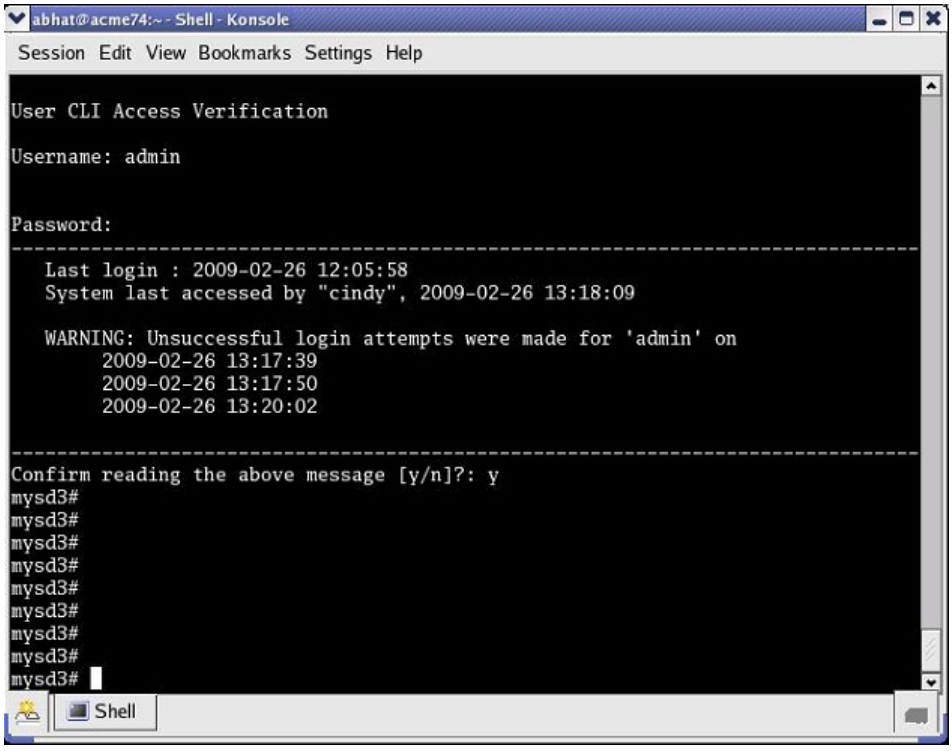
- ACP is open

The below table reflects feature availability under each license scenario.

Case	Admin Security Features	Shell Access	Telnet, FTP, and SSH Keys	History Log File in code/ history	Password-policy	EMS access and ACP Ports
Only Admin Security license is present	enabled	denied	denied	denied	enabled	blocked
Only Admin Security license was deleted	disabled	denied	denied	denied	disabled	open
Only Admin Security ACP license is present	enabled	denied	denied	denied	enabled; **password-security-strength** is available	open
Only Admin Security ACP license was deleted	disabled	denied	denied	denied	disabled	open
Both are present	enabled	denied	denied	denied	enabled; **password-security-strength** is available	open
Both were present and only Admin-Security license was deleted	enabled	denied	denied	denied	enabled; **password-security-strength** is available	open
Both were present and only Admin-Security ACP license was deleted	enabled	denied	denied	denied	enabled; **password-security-strength** is not available	blocked
Both were present then both were deleted	disabled	denied	denied	denied	disabled	open

Login Banner

Upon successful user authentication/authorization, the Oracle SBC 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 SBC.

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

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).

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:

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# admin-security
ACMEPACKET(admin-security)# login-config
ACMEPACKET(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.

```
ACMEPACKET(login-config)# concurrent-session limit 4
ACMEPACKET(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.

```
ACMEPACKET(login-config)# max-login-attempts 5
ACMEPACKET(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.

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

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

   Allowable values are integers within the range 30 through 300, with a default value of 60 seconds.

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

```
ACMEPACKET(login-config)# lockout-interval 30
ACMEPACKET(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.

```
ACMEPACKET(login-config)# send-alarm disabled
ACMEPACKET(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.

```
ACMEPACKET(login-config)# login-auth-mode two-factor
ACMEPACKET(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.

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

A sample login policy configuration appears below:

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

Password Policy

Both the Admin Security and Admin Security ACP licenses support the creation of a password policy that enhances the authentication process by imposing requirements for:

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

The Admin Security license restricts access to the ACP ports and mandates the following password length/strength requirements.

- user password must contain at least 9 characters
- admin password must contain at least 15 characters

  The Admin Security and Admin Security ACP licenses both work to increase the security of the Oracle Communications Session Border Controller (SBC). If a device already has an Admin Security license installed, you can add an Admin Security ACP license later in certain high-security environments. Both licenses may co-exist on a single device, or either license may be on the device alone. An Admin Security ACP license performs the same functions as an Admin Security license, but also allows access to the ACP ports blocked by an Admin Security license.
- passwords must contain at least 2 lower case alphabetic characters
- passwords must contain at least 2 upper case alphabetic characters
- passwords must contain at least 2 numeric characters
- passwords must contain at least 2 special characters
- passwords must differ from the prior password by at least 4 characters
- passwords cannot contain, repeat, or reverse the user name
- passwords cannot contain three consecutive identical characters

The Admin Security ACP license imposes the same password length/strength requirements as above except for the minimum length requirement, and also maintains or reopens access to the ACP ports.

With the enabling of the password-strength command as part of the Admin Security ACP license, you also impose these requirements:

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

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

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.

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

   The password-policy configuration element properties (with the introduction of the Admin Security or Admin Security ACP license) 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 ACP license 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 1 through 65535, with a default value of 90 (days).

```
ACMEPACKET(password-policy)# expiry-interval 60
ACMEPACKET(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).

```
ACMEPACKET(password-policy)# password-change-interval 18
ACMEPACKET(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.

```
ACMEPACKET(password-policy)# expiry-notify-period 10
ACMEPACKET(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).

```
ACMEPACKET(password-policy)# grace-period 1
ACMEPACKET(password-policy)# grace-logins 1
ACMEPACKET(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 10, with a default value of 3 (retained passwords).

   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.

```
ACMEPACKET(password-policy)# password-history-count 10
ACMEPACKET(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 license, this command can be safely ignored.

   password-policy-strength may be enabled when the Admin Security ACP license is enabled. This license includes all the password security features contained in the Admin Security license and also adds password strength requirements beyond those imposed by the Admin Security license. 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 license, retain the default value (disabled). With the Admin Security ACP license installed, use enabled to add the new password requirements as listed above; use disabled to retain only the password requirements defined by the Admin Security license.

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

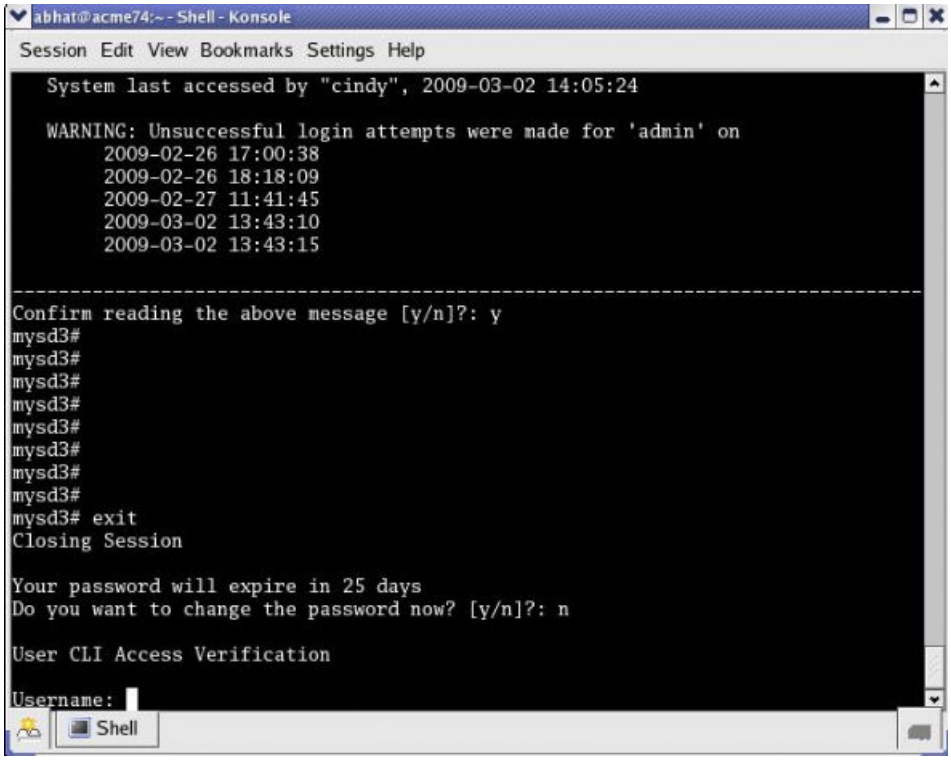
9. Use done, exit and verify-config to complete password policy.

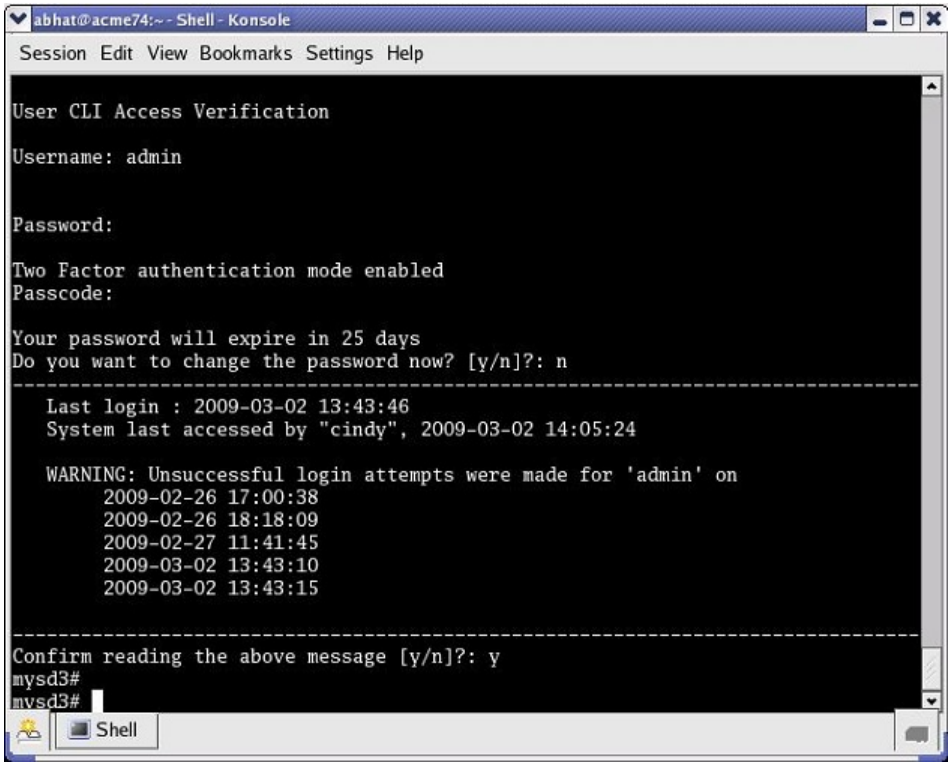
RADIUS Passwords

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 password policy are applicable to RADIUS passwords.

Changing a Password

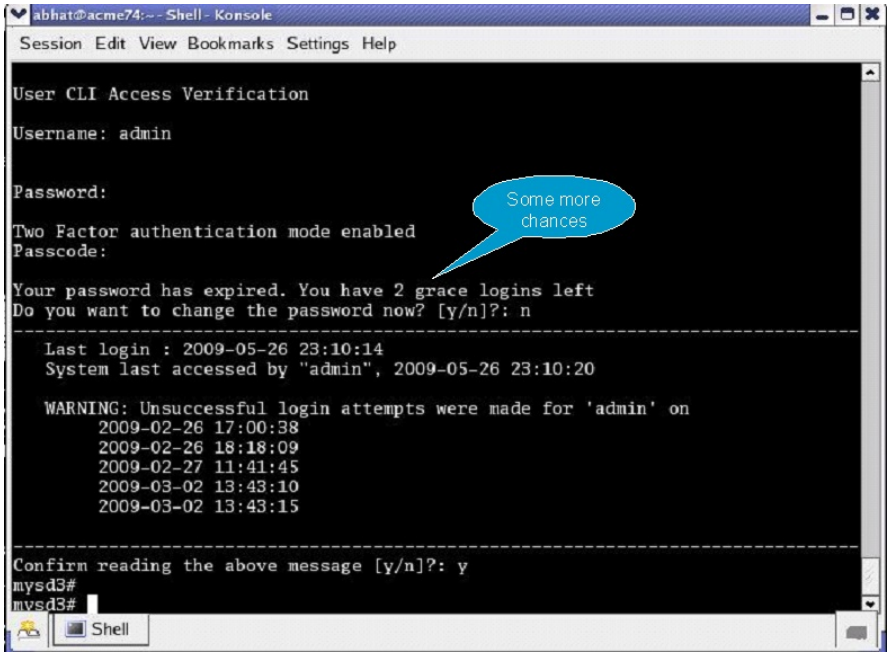
As shown in the following figures, the **password-policy** configuration element provides prior notice of impending password expiration via the login banner display, and with additional notices when ending a login session.

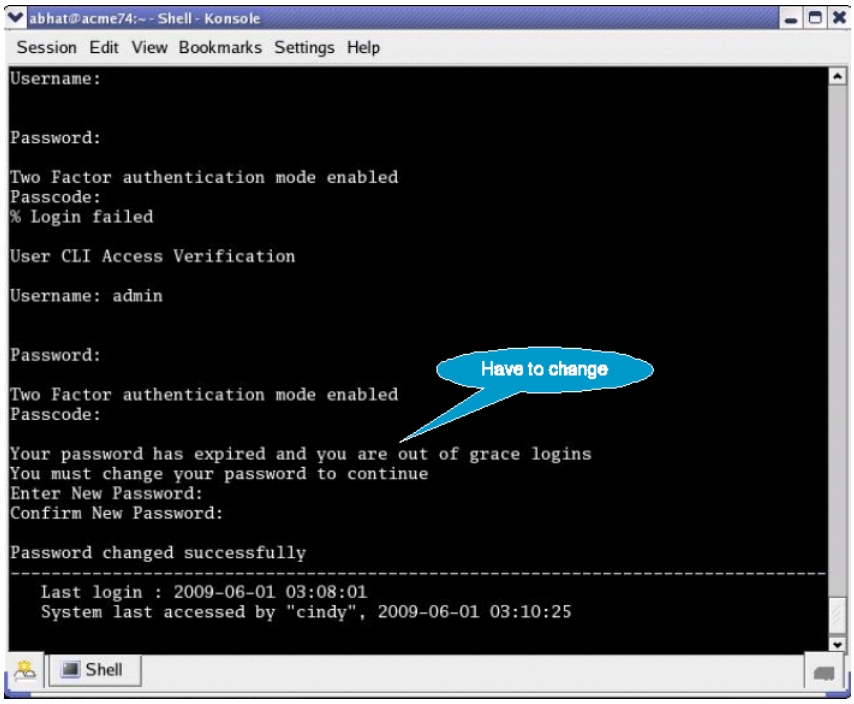




Password Expiration Notices at Login and Logout

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





Grace Login Reminder/Forced Password Change

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.

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

2. Type the new password in response to the Enter new password : prompt.

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

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

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

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.

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

2. Type the new password in response to the Enter new password : prompt.

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

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

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

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.

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

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

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

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

```
ACMEPACKET# 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.

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

Changing the admin Passcode

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

1. Enter secret enable passcode at the prompt.

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

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

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

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

```
ACMEPACKET# 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.

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

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 authentication and authorization of users that is performed locally by the Oracle SBC that is equipped with an active Admin Security license.

The license provides 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 an active Admin Security license, local login to the Oracle SBC 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.

Table 1: Local Login Authentication & Authorization

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

Serial Port Control

With an active Admin Security license, users have the ability to enable or disable access to the serial (console) port. In the absence of this license, 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 license after disabling the serial port, the SBC reverts to its default behavior by providing serial port access.

To turn off access to the serial port:

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

```
ACMEPACKET# console-io disabled
```

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

Initial Login

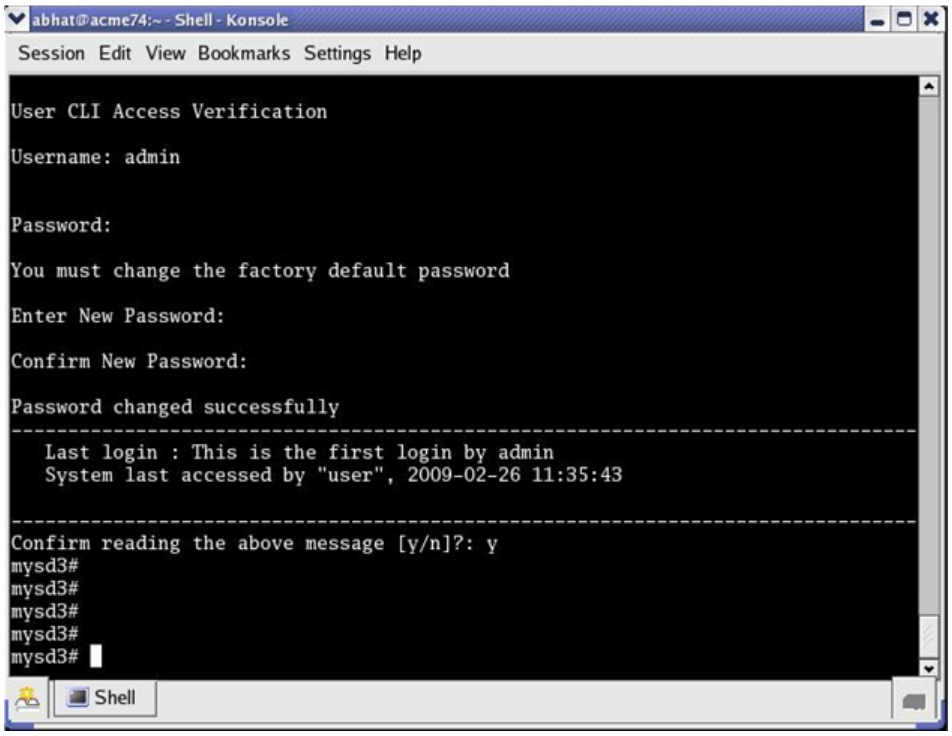
Upon initial login user and admin are required to change the respective password. Initial login is completed only after password change and acknowledgment of the login banner.

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.

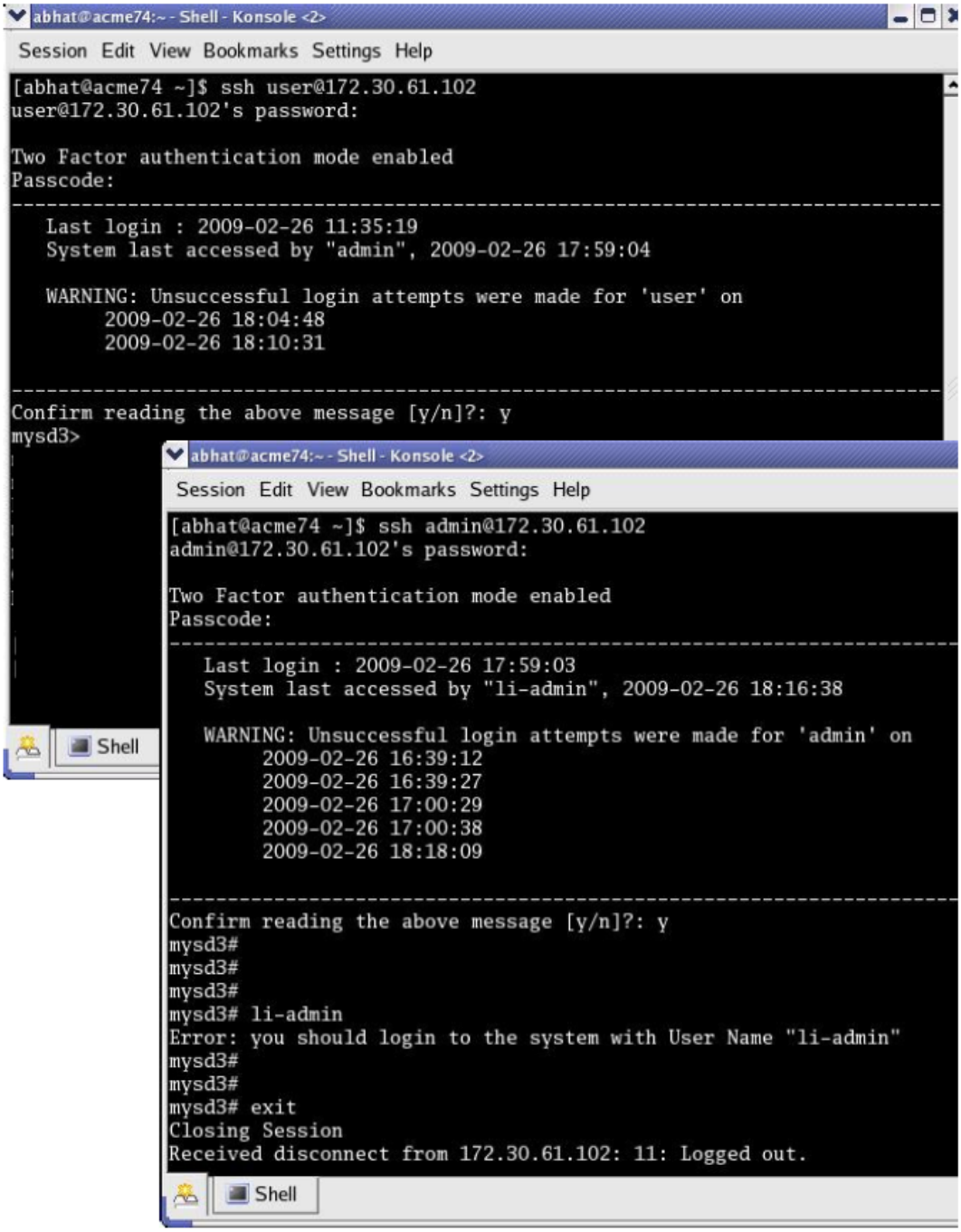
   - user password must contain at least 9 characters
   - admin password must contain at least 15 characters
   - passwords must contain at least 2 lower case alphabetic characters
   - passwords must contain at least 2 upper case alphabetic characters
   - passwords must contain at least 2 numeric characters
   - 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 an active Admin Security license, remote access, via the management interface (also referred to as wancom0), is available using SSH Version 2; telnet access is not allowed under the Admin Security license.

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.

Table 2: Remote Login (SSH/Password) Authentication & Authorization

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

admin	admin mode # 	authenticated locally by SBC via password	authorized locally by SBC 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 an active Admin Security license, 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 SBC performs authentication.

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

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 SBC.

   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

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

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

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

   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.

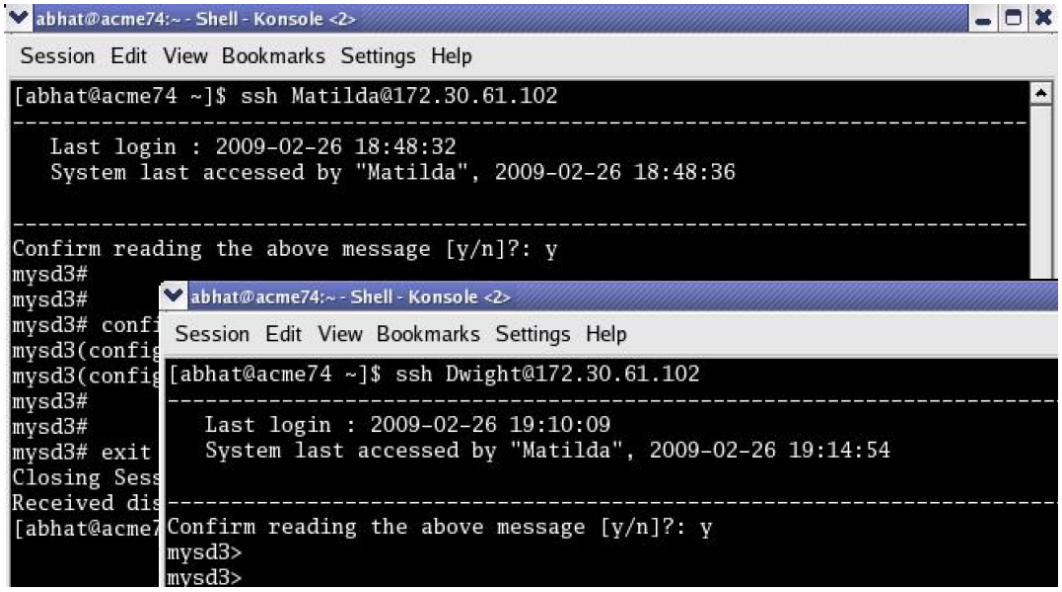
   ```
   ACMEPACKET# ssh-pub-key import authorized-key Matilda admin

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

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

```
SSH public key imported successfully....
WARNING: Configuration changed, run "save-config" command to save it
and run "activate-config" to activate the changes
ACMEPACKET# 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'.
ACMEPACKET# activate-config
Activate-Config received, processing.
waiting for request to finish
SD is not QOS-capable
Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete
ACMEPACKET#
```

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.

Table 3: Remote Login (SSH/Public Key) Authentication & Authorization

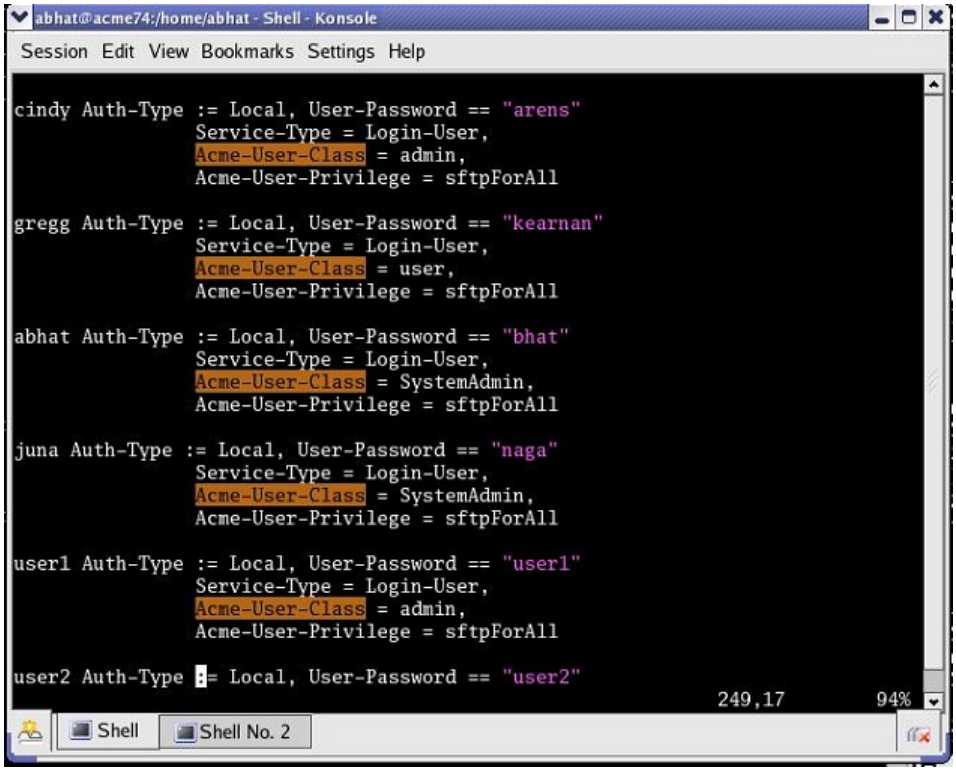
User Name	Logins into/prompt	Authentication	Authorization
not relevant	user mode > or admin mode #	authenticated locally by SBC via SSH public key	authorized locally by SBC authorization determined by authorizationClass command argument (user or admin) inherits access/ privilege defined by the specified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

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 SBC refer to RADIUS Authentication in the 3000 and 4000 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 SBC sends a RADIUS Access Request message to the RADIUS server. The request message contains, among other things, the username:password requesting access to SBC resources. Upon receiving the request, the RADIUS server checks its user file for the username:password pair. If its finds a congruent match, the requestor is authenticated.

Successful authentication generates a Access Accept message to the SBC; the message also contains the contents of two Oracle Vendor Specific Attributes (VSAs). Acme-User-Class specifies the configuration privileges accorded the authenticated user. Acme-User-Privilege specifies the log file access accorded to the authenticated user. Together

these two VSAs provide the authorization function. Consequently, the RADIUS server functions as an authentication and authorization decision point, while the SBC functions as an enforcement point.

RADIUS Authorization Classes

The RADIUS authorization classes, as specified by the Acme-User-Class VSA, 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 ACMEPACKET>

SystemAdmin (RADIUS Acme-User-Class = SystemAdmin)

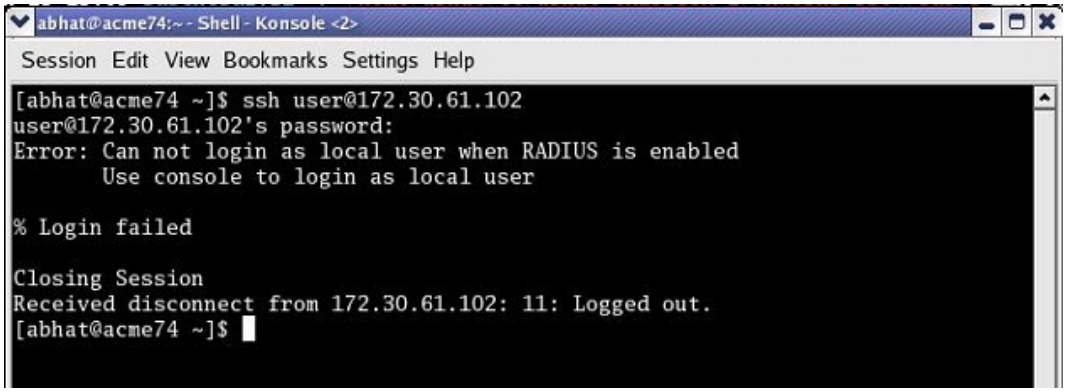
- provides read-write access for system configuration (not including cryptographic keys and certificates)
- The login prompt for this user is ACMEPACKET$

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 ACMEPACKET#

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)

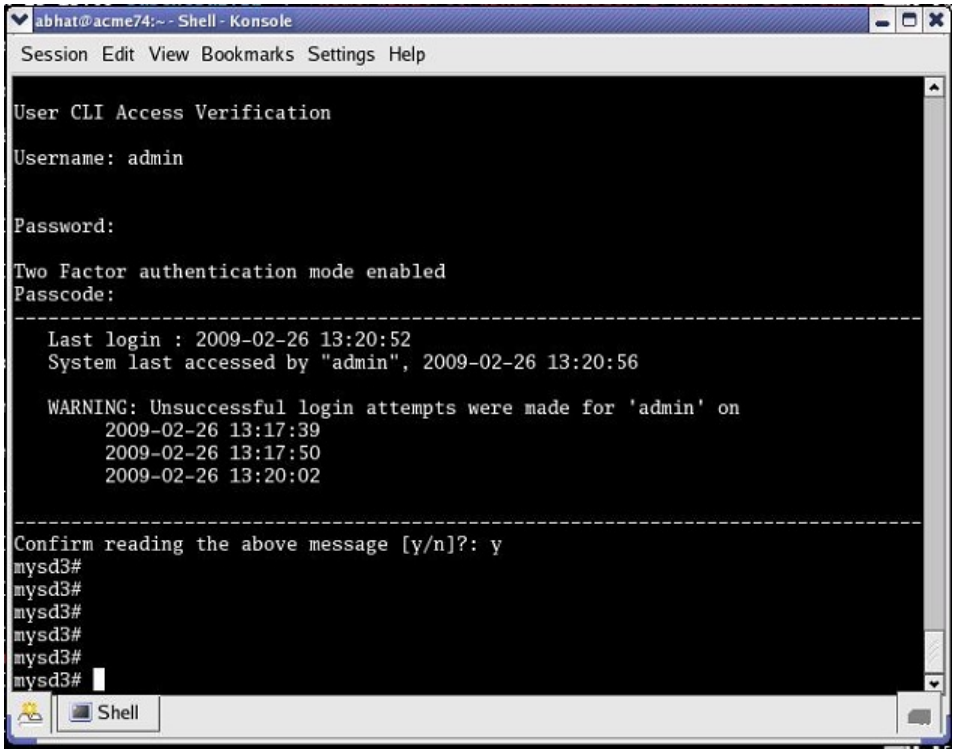
If you want to enable user and admin access via SSH with RADIUS configured, you must explicitly define users on the RADIUS server with appropriate Acme-User-Class.

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.

Two-Factor Authentication

Two-factor authentication, which adds an additional level of security, is available in support of local and SSH password authentication..



Two-Level Authentication

When enabled, two-factor authentication requires the authentication of a second passcode following the successful authentication of the initial password. Passcodes are subject to the length/strength requirements specified by the password policy; however they are not subject to other policy elements such as history or lifetime.

Two-factor authentication is not supported by RADIUS servers.

SSH and SFTP

With an active Admin Security license, 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 Oracle SBC, is defined by a series of five RFCs.

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

RFCs 4252 and 4253 are most relevant to SBC operations.

The transport layer protocol (RFC 4253) provides algorithm negotiation and key exchange. The key exchange includes server authentication and results in a cryptographically secured connection that provides integrity, confidentiality and optional compression. Forward security is provided through a Diffie-Hellman key agreement. This key agreement results in a shared session key. The rest of the session is encrypted using a symmetric cipher, currently 128-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 SBC: traditional password authentication and public-key authentication.

Configuring SSH Properties

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

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

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# admin-security
ACMEPACKET(admin-security)# ssh-config
ACMEPACKET(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.

```
ACMEPACKET(ssh-config)# rekey-interval 20
ACMEPACKET(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.

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

   A sample SSH configuration appears below:

```
ACMEPACKET(ssh-config)# rekey-interval 20
ACMEPACKET(ssh-config)# done
ACMEPACKET(ssh-config)# exit
ACMEPACKET(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 SBC.

   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.

   ```
   ACMEPACKET# 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.

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

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

   ---- BEGIN SSH2 PUBLIC KEY ----
   Comment: "2048-bit RSA, converted from OpenSSH by klee@acme54"
   AAAAB3NzaC1yc2EAAAABIwAAAQEA7OBf08jJe7MSMgerjDTgZpbPblrX4n17LQJgPC7clL
   cDGEtKSiVt5MjcSav3v6AEN2pYZihOxd2Zzismpoo019kkJ56s/IjGstEzqXMKHKUr9mBV
   qvqIEOTqbowEi5sz2AP31GUjQTCKZRF1XOQx8A44vHZCum93/jfNRsnWQ1mhHmaZMmT2LS
   hOr4J/Nlp+vpsvpdrolV6Ftz5eiVfgocxrDrjNcVtsAMyLBpDdL6e9XebQzGSS92TPuKP/
   yqzLJ2G5NVFhxdw5i+FvdHz1vBdvB505y2QPj/iz1u3TA/3O7tyntBOb7beDyIrg64Azc8
   G7E3AGiH49LnBtlQf/aw==
   ---- END SSH2 PUBLIC KEY ----
   ;
   SSH public key imported successfully....
   WARNING: Configuration changed, run "save-config" command to save it
   and run "activate-config" to activate the changes
   ACMEPACKET# 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'.
   ACMEPACKET# activate-config
   Activate-Config received, processing.
   waiting for request to finish
   SD is not QOS-capable
   Request to 'ACTIVATE-CONFIG' has Finished,
   Activate Complete
   ACMEPACKET#
   ```

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 SBC.

   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

   ```
   ACMEPACKET# 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.

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

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

   ---- BEGIN SSH2 PUBLIC KEY ----
   Comment: "1024-bit RSA, converted from OpenSSH by abhat@acme74"
   AAAAB3NzaC1yc2EAAAABIwAAAIEAxcYTV595VqdHy12P+mIZBlpeOZx9sX/mSAFihDJYdL
   qJIWdiZuSmny8HZIxTIC6na62iD25mlEdyLhlYOuknkYBCU7UsLwmx4dLDyHTbrQHz3b1q
   3Tb8auz97/J1p4pw39PT42CoRODzPBrXJV+OglNE/83C1y0SSJ8BjC9LEwE=
   ---- END SSH2 PUBLIC KEY ----;
   SSH public key imported successfully....
   WARNING: Configuration changed, run "save-config" command to save it
   and run "activate-config" to activate the changes
   ACMEPACKET# 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'.
ACMEPACKET# activate-config
Activate-Config received, processing.
waiting for request to finish
SD is not QOS-capable
Request to 'ACTIVATE-CONFIG' has Finished,
Activate Complete
ACMEPACKET#
```

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.

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

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

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

ACMEPACKET(public-key)#
```

   creates a public key record named tashtego.

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

```
ACMEPACKET(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
ACMEPACKET(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.

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

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.

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

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.

   ```
   ACMEPACKET# save-config
   ...
   ...
   ...
   ACMEPACKET# activate-config
   ...
   ...
   ...
   ```

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

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

   selection: 3
   ACMEPACKET(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
   ACMEPACKET(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 SBC with the ssh-pub-key command; you cannot display information about keys generated by the ssh-pub-key command.

```
ACMEPACKET# 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
ACMEPACKET#
```

   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

```
ACMEPACKET# 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
ACMEPACKET#
```

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

```
ACMEPACKET# 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:

AAAAB3NzaC1yc2EAAAABIwAAAQEA7OBf08jJe7MSMgerjDTgZpbPblrX4n17LQJgPC7clLcDGEtK
SiVt5MjcSav3v6AEN2pYZihOxd2Zzismpoo019kkJ56s/
IjGstEzqXMKHKUr9mBVqvqIEOTqbowEi5sz2AP31GUjQTCKZRF1XOQx8A44vHZCum93/
jfNRsnWQ1mhHmaZMmT2LShOr4J/Nlp
```

```
+vpsvpdrolV6Ftz5eiVfgocxrDrjNcVtsAMyLBpDdL6e9XebQzGSS92TPuKP/
yqzLJ2G5NVFhxdw5i+FvdHz1vBdvB505y2QPj/iz1u3TA/
3O7tyntBOb7beDyIrg64Azc8G7E3AGiH49LnBtlQf/aw==

modulus: (256)
ECE05FD3C8C97BB3123207AB8C34E06696CF6E5AD7E27D7B2D02603C2EDC94B703184B4A4A25
6DE4C8DC49ABF7BFA004376A5866284EC5DD99CE2B26A68A34D7D924279EACFC88C6B2D133A9
730A1CA52BF66055AAFA8810E4EA6E8C048B9B33D803F7D4652341308A6511755CE431F00E38
BC7642BA6F77FE37CD46C9D64359A11E66993264F62D284EAF827F365A7EBE9B2FA5DAE8955E
85B73E5E8957E0A1CC6B0EB8CD715B6C00CC8B0690DD2FA7BD5DE6D0CC6492F764CFB8A3FFCA
ACCB2761B9355161C5DC398BE16F747CF5BC176F079D39CB640F8FF8B3D6EDD303FDCEEEDCA7
B4139BEDB783C88AE0EB803373C1BB137006887E3D2E706D9507FF6B
exponent: (1)
23

ACMEPACKET#
```

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.

```
ACMEPACKET# 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:

AAAAB3NzaC1kc3MAAACBAPY8ZOHY2yFSJA6XYC9HRwNHxaehvx5wOJ0rzZdzoSOXxbETW6ToHv8D
1UJ/z+zHo9Fiko5XybZnDIaBDHtblQ
+Yp7StxyltHnXF1YLfKD1G4T6JYrdHYI14Om1eg9e4NnCRleaqoZPF3UGfZia6bXrGTQf3gJq2e7
Yisk/gF
+1VAAAAFQDb8D5cvwHWTZDPfX0D2s9Rd7NBvQAAAIEAlN92+Bb7D4KLYk3IwRbXblwXdkPggA4pf
dtW9vGfJ0/RHd+NjB4eo1D+0dix6tXwYGN7PKS5R/FXPNwxHPapcj9uL1Jn2AWQ2dsknf+i/
FAAvioUPkmdMc0zuWoSOEsSNhVDtX3WdvVcGcBq9cetzrtOKWOocJmJ80qadxTRHtUAAACBAN7CY
+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQeSXG1vO
+JsvphVMBJc9HSn24VYtYtsMu74qXviYjziVucWKjjKEb11juqnF0GDlB3VVmxHLmxnAz643WK42
Z7dLM5sY29ouezv4Xz2PuMch5VGPP+CDqzCM4loWgV

p: (128)
F63C64E1D8DB2152240E97602F47470347C5A7A1BF1E70389D2BCD9773A12397C5B1135BA4E8
1EFF03D5427FCFECC7A3D162928E57C9B6670C86810C7B5B950F98A7B4ADC7296D1E75C5D582
DF283D46E13E8962B747608D783A6D5E83D7B836709195E6AAA193C5DD419F6626BA6D7AC64D
07F7809AB67BB622B24FE017ED55
q: (20)
DBF03E5CBF01D64D90CF7D7D03DACF5177B341BD
g: (128)
94DF76F816FB0F828B624DC8C116D76E5C177643E0800E297DDB56F6F19F274FD11DDF8D8C1E
1EA350FED1D8B1EAD5F060637B3CA4B947F1573CDC311CF6A9723F6E2F5267D80590D9DB249D
FFA2FC5000BE2A143E499D31CD33B96A12384B12361543B57DD676F55C19C06AF5C7ADCEBB4E
2963A8709989F34A9A7714D11ED5
pub_key: (128)
```

```
DEC263E28ABF5807A51CC5C1D426EC72BD6DBD4B028D8AC1AA179DA74581EA6D34141E4971B5
BCEF89B2FA6154C04973D1D29F6E1562D62DB0CBBBE2A5EF8988F3895B9C58A8E32846F5D63B
AA9C5D060E50775559B11CB9B19C0CFAE3758AE3667B74B339B18DBDA2E7B3BF85F3D8FB8C72
1E5518F3FE083AB308CE25A16815

ACMEPACKET#
```

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

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

displays detailed information for all SSH imported keys.

SFTP Operations

SFTP is an interactive file transfer program, similar to FTP, which 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.

Command	Description
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.

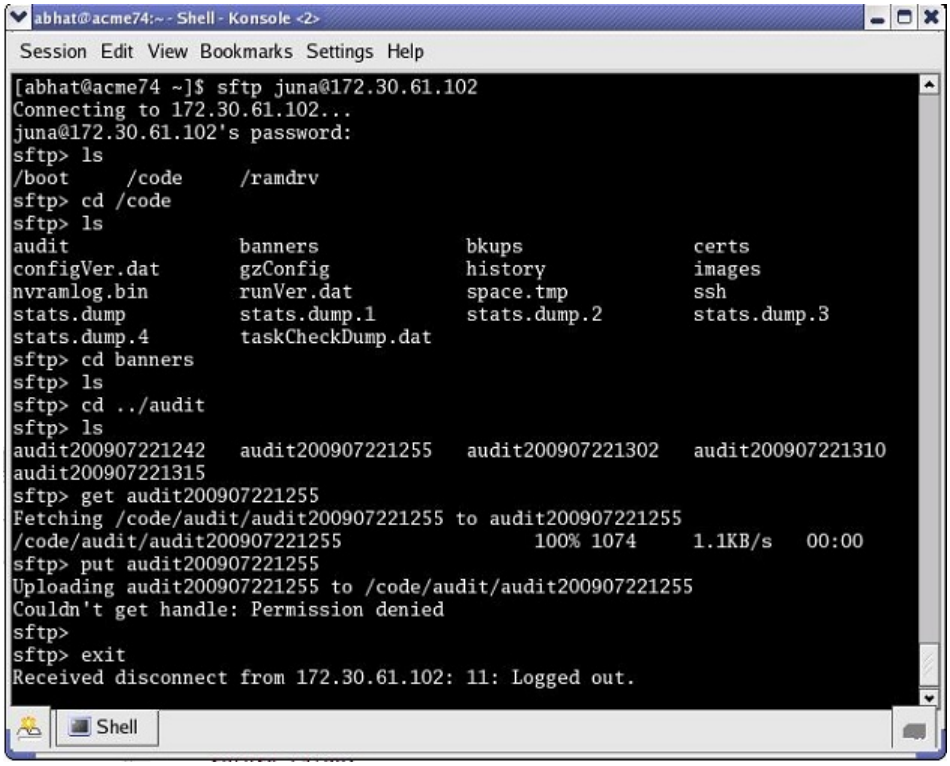
Command	Description
lmkdir path	Create local directory specified by path.
ln oldpath newpath	Create a symbolic link from oldpath to newpath.
lpwd	Print local working directory.
ls [path]	Display remote directory listing of either path or current directory if path is not specified.
lumask umask	Set local 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.

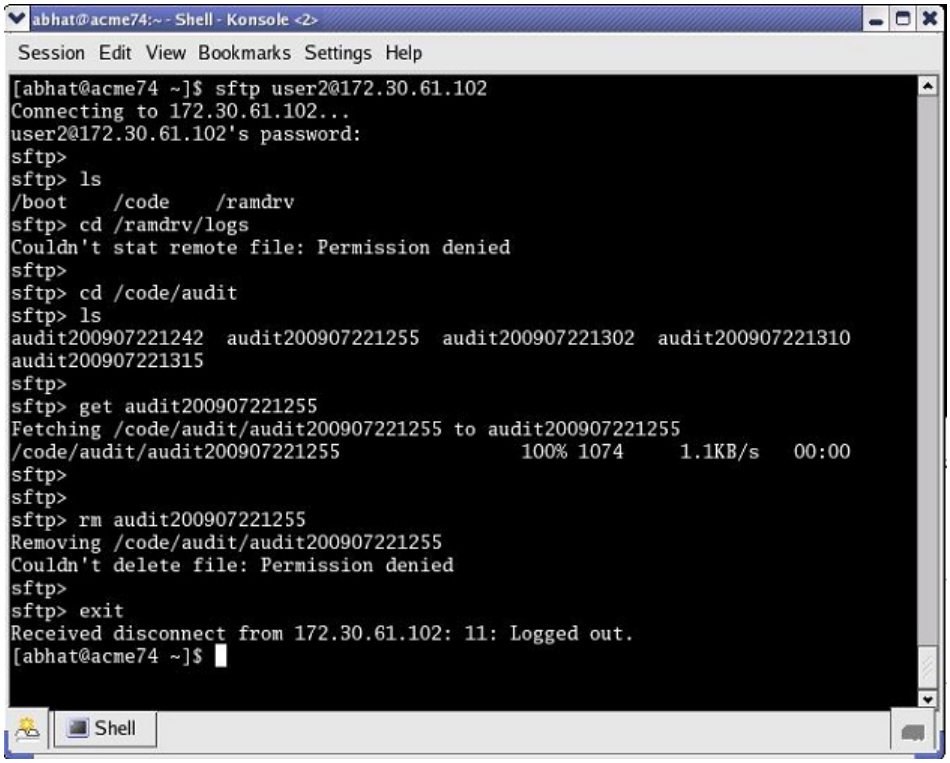
Note: Command availability is subject to Oracle authorization/privilege classes.

Some SFTP commands are available to only certain users; some commands are available to no users.

The following figure which shows two sample SFTP sessions illustrates some facets of SFTP authentication and authorization.

juna presents an SSH public key as an authentication credential, and after successful authentication/authorization, is granted admin privileges. user presents a password as an authentication credential, and after successful authentication/authorization, is granted user privileges.





SFTP Authentication/Authorization

Note juna's inability to access the put command (which moves a file from the local system to the SBC), and user's inability to access a sub-directory under /opt, or to delete an audit log.

The following table summarizes SFTP authentication and authorization.

Table 4: SFTP Authentication & Authorization

User Name	Logins into/prompt	Authentication	Authorization
user	not relevant	authenticated locally by SBC via password	authorized locally by SBC assigned to user class inherits access/privilege defined by that class
admin	not relevant	authenticated locally by SBC via password	authorized locally by SBC assigned to admin class inherits access/privilege defined by that class
or			
not relevant	not relevant	authenticated locally by SBC via SSH public key	authorized locally by SBC authorization determined by authorizationClass command argument (user or admin) inherits access/privilege defined by the specified class

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

2

Audit Log

Overview

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

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

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

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

   audit<timestamp>

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

   audit200903051630

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

Audit Log Format

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

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

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

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

Category takes the values: security | configuration | system

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

Result takes the values: successful | unsuccessful

Resource identifies the configuration element accessed by the user

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

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

The following chart summarizes 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'`

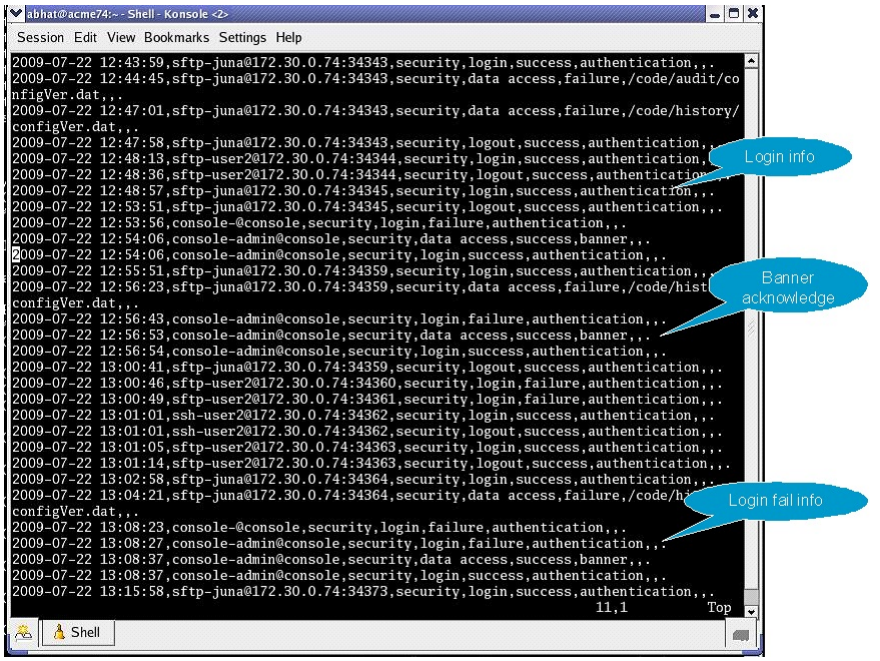
	lastModifiedDate=’2009-03-05 15:45:01> </sshPubKeyRecord
Modify	a) any action that modifies a configuration property 2009-03-05 15:48:01,acliConsole-admin@console,configuration,modify,success,public-key, Previous= <?xml version=’1.0’ standalone=’yes’?> <sshPubKeyRecord name=’dummy’ comment=’’ keyType=’2’ 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

Viewing the Audit Log

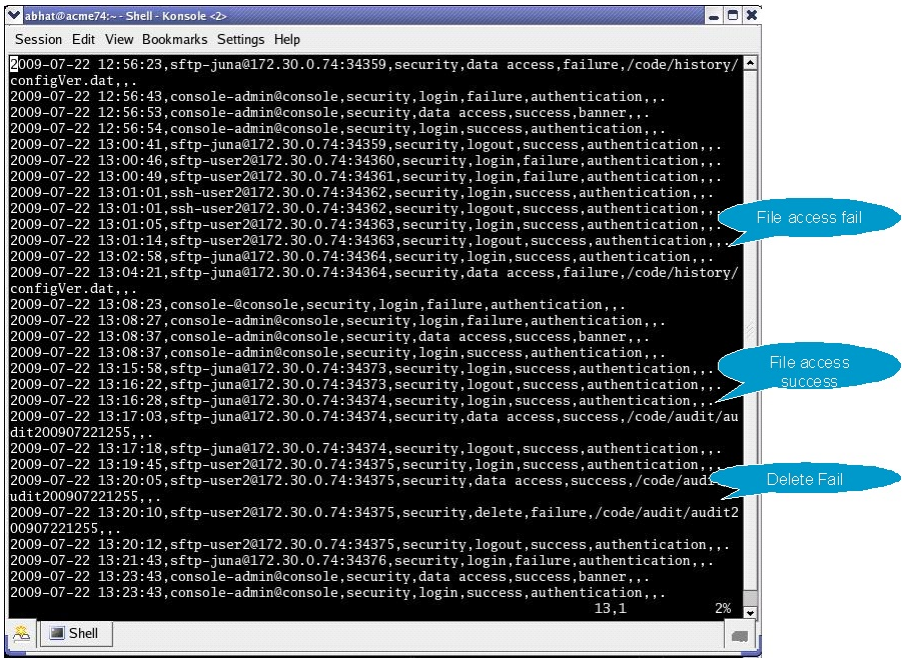
The audit log can be displayed only after transfer to an SFTP server, either by (1) automatic transfer triggered by a timer, or space-based threshold as previously described; or by (2) manual SFTP transfer accomplished by the admin user.

Audit Log Samples

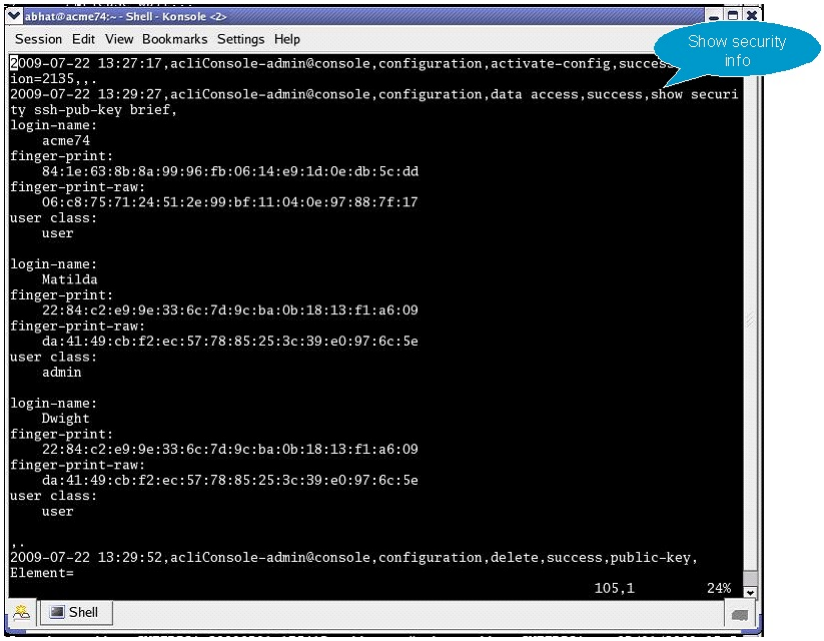
The follow screen captures provide samples of specific audit log entries.



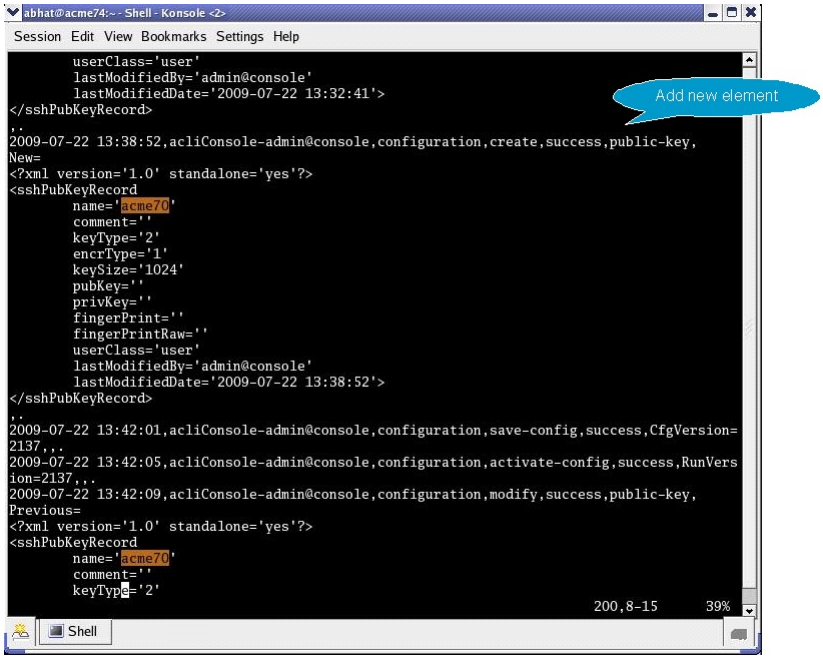
Login Reporting



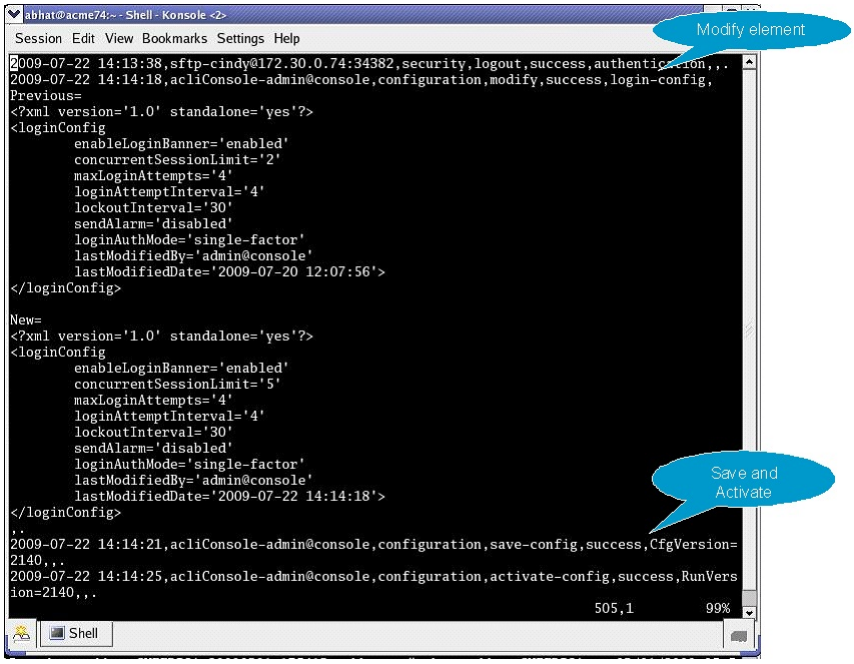
File Access Reporting



show security Reporting



Create Element Reporting



Modify Element/Activate Reporting

Configuring 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. From admin mode, use the following command path to access the audit-logging configuration element:

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

audit-logging configuration element properties are shown below with their default values

```
admin-state             disabled
detail-level            brief
file-transfer-time      720
max-storage-space       32
percentage-full         75
max-file-size           5
storage-path            /code/audit
```

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

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

```
ACMEPACKET(audit-logging)# admin-state enable
ACMEPACKET(audit-logging)#
```

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.

```
ACMEPACKET(audit-logging)# detail-level verbose
ACMEPACKET(audit-logging)#
```

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

   Allowable values are integers within the range 0 through 65535.

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

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

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

```
ACMEPACKET(audit-logging)# file-transfer-time 1
ACMEPACKET(audit-logging)#
```

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

   Allowable values are integers within the range 1 through 32.

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

   ```
   ACMEPACKET(audit-logging)# max-storage-space 8
   ACMEPACKET(audit-logging)#
   ```

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

   Allowable values are integers within the range 0 through 99.

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

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

   ```
   ACMEPACKET(audit-logging)# percentage-full 0
   ACMEPACKET(audit-logging)#
   ```

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

   Allowable values are integers within the range 0 through 10.

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

   Retain the default value (5 Megabytes), or provide an alternate value to trigger absolute-size-based-transfer. With absolute-size-based-transfer enabled, automatic upload of the audit file to an SFTP server or servers is triggered when audit log size exceeds the value **max-file-size**. At that time the audit log is transferred and an alarm alerting the recipient to the transfer is generated. Assuming the file transfer succeeds, the audit log is deleted. If the file transfer fails, the audit log is retained until it exceeds the value specified by **max-storage-space.**

   ```
   ACMEPACKET(audit-logging)# max-file-size 0
   ACMEPACKET(audit-logging)#
   ```

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

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

```
ACMEPACKET(audit-logging)# storage-path code/mgmt
ACMEPACKET(audit-logging)#
```

A sample audit log configuration appears below:

```
ACMEPACKET(admin-security)# admin-state enabled
ACMEPACKET(admin-security)# file-transfer-time 1
ACMEPACKET(admin-security)# percentage-full 0
ACMEPACKET(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.

Configuring 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 SBC. The host key identifies the server as a trusted entity when the SBC 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 SBC 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. From audit-logging mode, use the **push-receiver** command to access the configuration element:

   ```
   ACMEPACKET(audit-logging)# push-receiver
   ACMEPACKET(push-receiver)#
   ```

   push-receiver configuration element properties are shown below with their default values

   ```
   server              none
   port                22
   remote-path         "" (empty string)
   filename-prefix     "" (empty string)
   username            "" (empty string)
   auth-type           password
   password            "" (empty string)
   public-key          "" (empty string)
   ```

2. **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,

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

3. **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.

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

4. **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,

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

5. **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,

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

6. **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,

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

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

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

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

8. **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,

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

9. **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,

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

   A sample SFTP server configuration appears below:

   ```
   ACMEPACKET(push-receiver)# 192.0.2.100
   ACMEPACKET(push-receiver)# remote-path /home/acme
   ACMEPACKET(push-receiver)# filename-prefix auvik
   ACMEPACKET(push-receiver)# username acme
   ACMEPACKET(push-receiver)# auth-type public-key
   ACMEPACKET(push-receiver)# public-key acme01
   ACMEPACKET(push-receiver)# 192.0.2.125
   ACMEPACKET(push-receiver)# remote-path /security/auditLogs
   ACMEPACKET(push-receiver)# filename-prefix auvik
   ACMEPACKET(push-receiver)# username acme
   ACMEPACKET(push-receiver)# auth-type password
   ACMEPACKET(push-receiver)# password ************
   ```

   This configuration identifies two SFTP servers as audit log recipients.

The first server (192.0.2.100) requires SSH public key authentication. acme01 aliases the certificate presented to the server by the Oracle Oracle Communications Session Border Controller (SBC) in its SFTP client role.

The second server (192.0.2.125) requires SSH password authentication.

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.

3

Internet Key Exchange (IKEv2)

Release S-C6.2.0 provides support for Version 2 of the Internet Key Exchange Protocol (IKEv2) as defined in RFC 4306, Internet Key Exchange (IKEv2) Protocol, and for the related Dead Peer Detection (DPD) protocol as defined in RFC 3706, A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers.

IKEv2 operations are initially restricted to the wancom0 management interface of Net-Net SBC platforms, IKEv2 supports the establishment of up to ten IPsec tunnels across the interface, making it possible to encrypt all management traffic. Such traffic includes, but is not limited to:

- administrative logins
- CDR storage
- SFTP of accounting records
- syslogs
- RADIUS authentication
- SNMP traps and gets
- XML configuration
- audit log conveyance

wancom0 IKEv2 protocol operations can support either responder or initiator mode, meaning that the wancom0 IKEv2 protocol instance can receive and respond to tunnel signalling from a remote peer, or can initiate tunnel signalling to a remote peer. In initiator mode, certain IPsec tunnels can be automatically re-established after system restart or boot.

IKEv2 Overview

IKEv2 is used for the generation and exchange of cryptographic material between two IKEv2 peers. Peers use the exchanged material to establish IPsec tunnels.

All IKEv2 messages are request/response pairs. It is the responsibility of the IKEv2 requester to retransmit the request in the absence of a timely response.

IKEv2 has an initial handshake, which usually consists of two request/response pairs. The first request/response pair negotiates cryptographic algorithms and performs a Diffie-Hellman exchange. The second request/response pair (which is encrypted and integrity protected with keys based on the Diffie-Hellman exchange) reveals peer identities and provides for a certificate-based or shared-secret-based integrity check. The initial exchange results in the creation of an IKE Security Association (SA) which is required for the establishment of IPsec tunnels between the remote peers

After the initial handshake, additional requests can be initiated by either peer, and consist of informational messages or requests to establish IPsec tunnels. Informational messages convey such things as null messages for detecting peer aliveness, or information on the deletion of SAs.

The exchange to establish an IPsec tunnel consists of an optional Diffie-Hellman exchange (if perfect forward secrecy is required), nonces (so that a unique key for the IPsec tunnel is established), and negotiation of traffic selector values which indicate the addresses, ports, and protocol types to be transmitted through the tunnel.

IKEv2 Configuration Steps

IKEv2 configuration consists of the following steps, some of which are optional.

1. Configure IKEv2 global parameters.
2. Optionally, enable and configure the DPD Protocol.
3. If IKEv2 peer authentication is certificate-based, configure certificate profiles.
4. If configuration payload requests for IP addresses are handled locally, configure one or more local address pools.
5. Configure the wancom0 management interface for IKEv2 operations.
6. Configure IKEv2 SAs.
7. Assign the IKEv2 SA to an IPsec Security Policy.
8. Configure IPsec tunnels across the wancom0 interface.

IKEv2 Global Configuration

Use the following procedure to perform IKEv2 global configuration.

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

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

2. Use the **ike-version** parameter to specify IKEv2.

```
ACMEPACKET(ike-config)# ike-version 2
ACMEPACKET(ike-config)#
```

3. Use the **log-level** parameter to specify the contents of the IKE log.

   Events are listed below in descending order of criticality.

```
emergency (most critical)
critical
major
minor
warning
notice
info       (least critical — the 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)
```

   In the absence of an explicitly configured value, the default value of info is used.

```
ACMEPACKET(ike-config)# log-level warning
ACMEPACKET(ike-config)#
```

4. Use the optional **udp-port** parameter to specify the port monitored for IKE protocol traffic.

   In the absence of an explicitly configured value, the default port number of 500 is used.

```
ACMEPACKET(ike-config)# udp-port 5000
ACMEPACKET(ike-config)#
```

5. Use the optional **sd-authentication-method** to select the default method used to authenticate the IKEv2 SA.

   Two authentication methods are supported.

   shared-password—(the default) uses a PSK (pre-shared key) to authenticate the remote IKEv2 peer.

   certificate—uses an X.509 certificate to authenticate the remote IKEv2 peer.

   This global default can be over-ridden at the interface level.

```
ACMEPACKET(ike-config)# sd-authentication-method certificate
ACMEPACKET(ike-config)#
```

6. If **sd-authentication-method** is shared-password, use the **shared-password** parameter to specify the default PSK required for password-based IKEv2 authentication.

   The PSK is a string of ACSII printable characters no longer than 255 characters (not displayed by the ACLI).

   This global default can be over-ridden at the interface level.

```
ACMEPACKET(ike-config)# shared-password !yetAnotherPaSSword1of87354
ACMEPACKET(ike-config)#
```

7. If **sd-authentication-method** is certificate, use the **certificate-profile-id** to identify the default **ike-certificate-profile** configuration element that contains identification and validation credentials required for certificate-based IKEv2 authentication.

   Provide the name of an existing ike-certificate-profile configuration element.

   This global default can be over-ridden at the interface level.

```
ACMEPACKET(ike-config)# certificate-profile-id valCred-IKEv2
ACMEPACKET(ike-config)#
```

8. Use the optional **dpd-time-interval** parameter to specify the maximum period of inactivity before the DPD protocol is initiated on a specific endpoint.

   Allowable values are within the range 1 through 999999999 (seconds) with a default of 0.

   The default value, 0, disables the DPD protocol; setting this parameter to a non-zero value globally enables the protocol and sets the inactivity timer.

```
ACMEPACKET(ike-config)# dpd-time-interval 20
ACMEPACKET(ike-config)#
```

9. Use the optional **v2-ike-life-seconds** parameter to specify the default lifetime (in seconds) for the IKEv2 SA.

   Allowable values are within the range 1 through 999999999 (seconds) with a default of 86400 (24 hours).

   This global default can be over-ridden at the interface level.

```
ACMEPACKET(ike-config)# v2-ike-life-seconds 43200
ACMEPACKET(ike-config)#
```

10. Use the optional **v2-ipsec-life-seconds** parameter to specify the default lifetime (in seconds) for the IPsec SA.

    Allowable values are within the range 1 through 999999999 (seconds) with a default of 28800 (8 hours).

    This global default can be over-ridden at the interface level.

```
ACMEPACKET(ike-config)# v2-ipsec-life-seconds 14400
ACMEPACKET(ike-config)#
```

11. Retain the default value for the optional **eap-protocol** parameter.

    The default, and only currently-supported value, eap-radius-passthru, specifies the use of a RADIUS server for Extensible Authentication Protocol (EAP) processing. The SG shuttles incoming and outgoing EAP messages between the remote IKEv2 peer and the RADIUS server.

```
ACMEPACKET(ike-config)# eap-protocol eap-radius-passthru
ACMEPACKET(ike-config)#
```

12. Use the optional **eap-bypass-identity** parameter to specify whether or not to bypass the EAP (Extensible Authentication Protocol) identity phase.

    EAP, defined in RFC 3748, Extensible Authentication Protocol (EAP), provides an authentication framework widely used in wired and wireless networks.

    An Identity exchange is optional within the EAP protocol exchange. Therefore, it is possible to omit the Identity exchange entirely, or to use a method-specific identity exchange once a protected channel has been established.

    However, where roaming is supported, it may be necessary to locate the appropriate backend authentication server before the authentication conversation can proceed. The realm portion of the Network Access Identifier (NAI) is typically included within the EAP-Response/Identity to enable the routing of the authentication exchange to the appropriate authentication server. Therefore, while the peer-name portion of the NAI may be omitted in the EAP-Response/Identity where proxies or relays are present, the realm portion may be required.

    Identify bypass is disabled by default — thus requiring an identity exchange.

```
ACMEPACKET(ike-config)# eap-bypass-identity enabled
ACMEPACKET(ike-config)#
```

13. Use the optional **addr-assignment** parameter to specify the default 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 protected network. In response, IKEv2 returns a local address for use by the requesting peer.

    This parameter specifies the source of the returned IP address.

    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

    This global default can be over-ridden at the interface level.

```
ACMEPACKET(ike-config)# addr-assignment radius-only
ACMEPACKET(ike-config)#
```

14. Use the **overload-threshold**, **overload-interval**, **overload-action**, **overload-critical-threshold**, and **overload-critical-interval** parameters to configure system response to an overload state.

    Use the optional **overload-threshold** parameter to specify the percentage of CPU usage that triggers an overload state.

    Values are within the range 1 through 100 (percent) with a default of 100, which effectively disables overload processing.

```
ACMEPACKET(ike-config)# overload-threshold 60
ACMEPACKET(ike-config)#
```

    Use the optional **overload-interval** parameter to specify the interval (in seconds) between CPU load measurements when in the overload state.

    Values are within the range 1 through 60 (seconds) with a default of 1.

```
ACMEPACKET(ike-config)# overload-interval 3
ACMEPACKET(ike-config)#
```

    Use the optional **overload-action** parameter to specify response to an overload state. The overload state is reached when CPU usage exceeds the percentage threshold specified by the **overload-threshold** parameter.

    By default, no preventive action is taken in response to an overload, You can, however, use this parameter to implement a call rejection algorithm in response to the overload. With the algorithm enabled, the CPU uses the following calculation to reject/drop some number of incoming calls:

    DropRate = (currentLoad - overloadThreshold) / (100 -overloadThreshold)

Thus, assuming a current CPU load of 70% and an overload threshold of 60%, the SG drops 1 of out every 4 incoming calls until the load falls below the threshold value.

Use **none** to retain default behavior (no action); use **drop-new-connection** to implement call rejection.

```
ACMEPACKET(ike-config)# overload-action drop-new-connection
ACMEPACKET(ike-config)#
```

Use the optional **overload-critical-threshold** parameter to specify the percentage of CPU usage that triggers a critical overload state.

When this threshold is exceeded, the SBC drops all incoming calls until the load drops below the critical threshold level, at which point it may drop selective calls depending on the value of the **overload-threshold** parameter.

Values are within the range 1 through 100 (percent) with a default of 100, which effectively disables overload processing.

Ensure that this threshold value is greater than the value assigned to **overload-threshold**.

```
ACMEPACKET(ike-config)# overload-critical-threshold 75
ACMEPACKET(ike-config)#
```

Use the optional **overload-critical-interval** parameter to specify the interval (in seconds) between CPU load measurements when in the critical overload state.

Values are within the range 1 through 60 (seconds) with a default of 1.

```
ACMEPACKET(ike-config)# overload-criticalinterval 2
ACMEPACKET(ike-config)#
```

15. Use the **red-port**, **red-max-trans**, **red-sync-start-time**, and **red-sync-comp-time** parameters to configure redundancy.

    Oracle SBCs can be deployed in pairs to deliver high availability (HA). Two SBCs operating in this way are called an HA node.

    Two SBCs work together in an HA node, one in active mode and one in standby mode.

    - The active SBC checks itself for internal process and IP connectivity issues. If it detects that it is experiencing certain faults, it will hand over its role as the active system to the standby SBC in the node.
    - The standby SBC is the backup system, which maintains a synchronous configuration with the active node. The standby SBC monitors the status of the active system so that, if needed, it can assume the active role without the active system having to instruct it to do so.

      Refer to High Availability Nodes in the 3000 and 4000 ACLI Configuration Guide for information on cabling and configuring HA nodes.

      Use the **red-port** parameter to specify the port number monitored for IKEv2 synchronization messages.

      The default value (0) effectively disables redundant high-availability configurations. Select a port value other than 0 (for example, 1995) to enable high-availability operations.

      ```
      ACMEPACKET(ike-config)# red-port 1995
      ACMEPACKET(ike-config)#
      ```

      Use the **red-max-trans** parameter to specify the maximum number of retained IKEv2 synchronization messages.

      Values are within the range 0 through 999999999 (messages) with a default of 10000.

      ```
      ACMEPACKET(ike-config)# red-trans 7500
      ACMEPACKET(ike-config)#
      ```

16. Use the **red-sync-start-time** parameter to specify the interval, in milliseconds, between health checks performed by the active node to confirm that it still retains this role.

    If the active role is verified, the timer is reset. If, for any reason, the health check is deficient, the active transitions to the standby role, and the previous standby assumes the active role.

    Supported values are integers within the range 0 through 999999999, with a default value of 5000 (5 seconds).

Values are within the range 0 through 999999999 (milliseconds) with a default of 500.

```
ACMEPACKET(ike-config)# red-sync-start-time 2500
ACMEPACKET(ike-config)#
```

Use the **red-sync-comp-time** parameter to specify the interval between standby initiated probes that confirm the availability of the active node.

Values are within the range 0 through 999999999 (milliseconds) with a default of 500.

```
ACMEPACKET(ike-config)# red-sync-comp-time 750
ACMEPACKET(ike-config)#
```

17. Use **done**, **exit**, and **verify-config** to complete configuration of IKEv2 global parameters.

DPD Configuration

IKEv2 peers can lose connectivity unexpectedly, perhaps as a result of routing problems, or reboot of one of the peers. Neither IKEv2 nor IPsec offers an efficient and scalable method to respond to connectivity loss. Consequently established SAs can remain in place until their configured lifetimes eventually expire. Such behavior results in mis-management of system resources and the presence of black holes where packets are tunneled to oblivion.

With DPD, each peer's state is largely independent of the other's. A peer is free to request proof of connectivity when it needed — there are no mandatory, periodic exchanges as would be required by a detection method based on keepalive or heartbeat messages. DPD asynchronous exchanges require fewer messages and achieve greater scalability.

If there is ongoing valid IPSec traffic between peers, there is little need to check connectivity. After a period of inactivity, however, connectivity is questionable. Verification of connectivity is only urgently necessary if there is traffic to be sent. For example, if one peer has IPsec traffic to send after the period of idleness, it need to know if its remote peer is still alive. At this point, peer A can initiate the DPD exchange.

If you enabled the DPD protocol with the **dpd-time-interval** parameter, use the following procedure to create a DPD template, an operational set of DPD parameters, that you subsequently assign to the wancom0 management interface.

This section can be safely ignored if you did not enable DPD.

DPD Configuration Steps

1. From superuser mode, use the following command sequence to access dpd-params configuration mode. While in this mode, you configure DPD templates.

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

2. Use the required **name** parameter to provide a unique identifier for this dpd-params instance.

   name enables the creation of multiple dpd-params instances.

```
ACMEPACKET(dpd-params)# name dpdTemplate-1
ACMEPACKET(dpd-params)#
```

3. Use the **max-loop** parameter to specify the maximum number DPD peers examined every **dpd-interval**, which value is established during IKE global configuration.

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

   Allowable values are within the range 1 through 999999999 (endpoints) with a default of 100.

```
ACMEPACKET(dpd-params)# max-loop 80
ACMEPACKET(dpd-params)#
```

4. Use the **max-endpoints** parameter to specify the maximum number of simultaneous DPD protocol negotiations supported when the CPU is not under load (as specified by the **max-cpu-limit** property).

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

   Allowable values are within the range 1 through 999999999 (endpoints) with a default of 25.

   ```
   ACMEPACKET(dpd-params)# max-endpoints 20
   ACMEPACKET(dpd-params)#
   ```

5. Use the **max-cpu-limit** parameter to specify a threshold value (expressed as a percentage of CPU capacity) at which DPD protocol operations are minimized to conserve CPU resources.

   Allowable values are within the range 0, which effectively disables DPD operations, through 100 (percent) with a default of 60.

   ```
   ACMEPACKET(dpd-params)# max-cpu-limit 50
   ACMEPACKET(dpd-params)#
   ```

6. Use the **load-max-loop** parameter to specify the maximum number of endpoints examined every **dpd-time-interval** when the CPU is under load, as specified by the **max-cpu-limit** parameter.

   Allowable values are within the range 1 through 999999999 (endpoints) with a default of 40. Ensure that the configured value is less than the value assigned to **max-loop**.

   ```
   ACMEPACKET(dpd-params)# load-max-loop 30
   ACMEPACKET(dpd-params)#
   ```

7. Use the **load-max-endpoints** parameter to specify the maximum number of simultaneous DPD Protocol negotiations supported when the CPU is under load, as specified by the **max-cpu-limit** property.

   Allowable values are within the range 1 through 999999999 (endpoints) with a default of 5. Ensure that the configured value is less than the value assigned to **max-endpoints**.

   ```
   ACMEPACKET(dpd-params)# load-max-endpoints 3
   ACMEPACKET(dpd-params)#
   ```

8. Use **done**, **exit**, and **verify-config** to complete configuration of the DPD template instance.
9. If necessary, repeat Steps 1 through 8 to configure additional DPD templates.

Certificate Profile Configuration

If authentication between IKEv2 peers is certificate based, use the following procedure to create one or more certificate profiles that provide identification and validation credentials for a specific wancom0 IKEv2 identity.

This section can be safely ignored if authentication is based upon a PSK.

1. From superuser mode, use the following command sequence to access ike-certificate-profile configuration mode. While in this mode, you configure certificate profiles.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# security
   ACMEPACKET(security)# ike
   ACMEPACKET(ike)# ike-certificate-profile
   ACMEPACKET(ike-certificate-profile)#
   ```

2. Use the required **identity** parameter to specify the IKEv2 entity that uses the authentication and validation credentials provided by this ike-certificate-profile instance.

   Identify the subject of this ike-certificate-profile by either an IP address or fully-qualified domain name (FQDN).

   identity enables the creation of multiple ike-certificate-profile instances.

   ```
   ACMEPACKET(ike-certificate-profile)# identity jojo.net
   ACMEPACKET(ike-certificate-profile)#
   ```

3. Use the required **end-entity-certificate** parameter to supply the unique name of a certificate-record configuration element referencing the identification credential (specifically, an X509.v3 certificate) offered by a local IKEv2 entity to verify its asserted identity.

```
ACMEPACKET(ike-certificate-profile)# end-entity-certificate ACME-1a
ACMEPACKET(ike-certificate-profile)#
```

4. Use the required **trusted-ca-certificates** parameter to compile a list or one or more certificate-record configuration elements referencing trusted Certification Authority (CA) certificates used to authenticate a remote IKEv2 peer

   Provide a comma separated list of existing CA **certificate-record** configuration elements.

```
ACMEPACKET(ike-certificate-profile)# trusted-ca-certificates verisignClass3-
a,verisignClass3-b,baltimore,thawte-a
ACMEPACKET(ike-certificate-profile)#
```

5. Use the optional **verify-depth** parameter to specify the maximum number of chained certificates that will be processed while authenticating the IKEv2 peer.

   Provide an integer within the range 1 through 10 (the default).

```
ACMEPACKET(ike-certificate-profile)# verify-depth 10
ACMEPACKET(ike-certificate-profile)#
```

6. Use **done**, **exit**, and **verify-config** to complete configuration of the ike-certificate-profile instance.
7. If necessary (for instance if you require individual certificates for each IPsec tunnel instance, repeat Steps 1 through 6 to configure additional ike-certificate-profile instances.

Certificate Chain Validation

Release S-C6.2.0 enhances the preparation of certificate chains when the remote peer (acting as the IPsec tunnel initiator) authenticates a wancom0 IPsec tunnel.

The SBC authenticates to the remote peer with a certificate chain starting with a certificate specific to the wancom0 tunnel instance (that is, the certificate referenced by the **end-entity-certificate** parameter), that certificate's immediate Certification Authority (CA) certificate, then the next intermediate CA certificate, and so on until it either reaches a configured maximum number of certificates (specified by the **verify-depth** parameter), or until it ends with a root CA certificate (a self-signed certificate in which the Issuer and Subject are the same). If the length of the certificate chain is constrained by the maximum limit, the SBC presents a partial certificate chain to the initiating peer, who can accept or reject it.

When in mutual-authentication mode, in which the server authenticates the TLS client, the requires a similar certificate chain from the client starting with the client's entity (end) certificate and containing a CA certificate trusted by the server before the configured maximum chain length is exceeded. The trusted CA certificate need not be a root CA, nor does it need to be the last certificate in the chain.

ACLI verify-config Command

The **verify-config** command has been enhanced to confirm that the entity (end) certificate specified by the **end-entity-certificate** parameter can to chained back to a trusted certificate (specified by **trusted-ca-certificates** parameter) within the chain length constraints imposed by the **verify-depth** parameter.

Hardware Requirements

Certificate chain validation requires the presence of an IPsec NIU and an SSM (Signaling Security Module) or SSM2.

Data Flow Configuration

If the Oracle SBC assigns local addresses in response to IKEv2 Configuration Payload requests, you must configure data-flows that you subsequently assign to a specific local address pool.

This section can be safely ignored if a RADIUS server provides address assignment services.

1. From superuser mode, use the following command sequence to access local-address-pool configuration mode. While in this mode, you configure bandwidth profiles.

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

2. Use the required **name** parameter to provide a unique identifier for this data-flow instance.

   name enables the creation of multiple data-flow instances.

```
ACMEPACKET(data-flow)# name omar
ACMEPACKET(data-flow)#
```

3. Use the required **realm-id** parameter to identify the realm that supports this data-flow instance.

```
ACMEPACKET(data-flow)# realm-id access-1
ACMEPACKET(data-flow)#
```

4. Use the optional **group-size** parameter to 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 SBC 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 integers within the range 1 through 255.

   For maximum efficiency, this value should be set to a power of 2.

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

5. Use the optional **upstream-rate** parameter to specify the allocated upstream bandwidth.

   Allowable values are integers within the range 0 (the default) through 999,999,999.

   The default value (0) allocates all available bandwidth.

```
ACMEPACKET(data-flow)# upstream-rate 560000000
ACMEPACKET(data-flow)#
```

6. Use the optional **downstream-rate** parameter to specify the allocated downstream bandwidth.

   Allowable values are integers within the range 0 (the default) through 999,999,999.

   The default value (0) allocates all available bandwidth.

```
ACMEPACKET(data-flow)# downstream-rate 280000000
ACMEPACKET(data-flow)#
```

7. Use **done**, **exit**, and **verify-config** to complete configuration of the data-flow instance.
8. If necessary, repeat Steps 1 through 7 to configure additional data-flow instances.

Local Address Pool Configuration

If the Oracle SBC assigns local addresses in response to IKEv2 Configuration Payload requests, you must configure local-address-pool instances that define realm-specific ranges of assignable IPv4 addresses.

This section can be safely ignored if a RADIUS server provides address assignment services.

1. From superuser mode, use the following command sequence to access local-address-pool configuration mode. While in this mode, you configure ranges of contiguous IP addresses.

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

2. Use the required **name** parameter to provide a unique identifier for this local-address-pool instance.

   name enables the creation of multiple local-address-pool instances.

```
ACMEPACKET(local-address-pool)# name phelps
ACMEPACKET(local-address-pool)#
```

3. Use the required **dns-realm-id** parameter to identify the DNS realm to which this local-address-pool instance is assigned.

```
ACMEPACKET(local-address-pool)# dns-realm-id access-1
ACMEPACKET(local-address-pool)#
```

4. Use the required **data-flow** parameter to identify the data-flow assigned to this local-address-pool instance.

```
ACMEPACKET(local-address-pool)# data-flow dFlow-1
ACMEPACKET(local-address-pool)#
```

5. Use **address-range** to move to address-range configuration mode.

```
ACMEPACKET(local-address-pool)# address-range
ACMEPACKET(address-range)#
```

6. Use **network-address** in conjunction with **subnet-mask** to define a contiguous pool of IPv4 addresses.

   The following sequence defines a range of 62 addresses from 192.168.0.1 through 192.168.0.62.

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

7. Use **done** and **exit** to complete configuration of the address-range instance.
8. Use **done**, **exit**, and **verify-config** to complete configuration of the local-address-pool instance.
9. If necessary, repeat Steps 1 through 8 to configure additional local-address-pool instances.

wancom0 Management Interface Configuration

Use the following procedure to configure the wancom0 management interface for IKEv2 operations.

1. Obtain the IP address of wancom0 management interface.

   If necessary, use the following command sequence to access the boot parameters which contain the wancom0 address.

   Press Enter to scroll through the boot parameters.

   The inet on ethernet (e) parameter contains the wancom0 IP address

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# bootparam
'.' = clear field; '-' = go to previous field; q = quit
bootdevice     : wancom0
processor number:    : 0
```

```
host name    : goose
file name    : nnSC620b1.gz
inet on ethernet (e)    : 172.30.55.127
...
...
```

2. From configuration mode, use the following command sequence to access ike-interface configuration mode.

```
ACMEPACKET(configure)# security
ACMEPACKET(security)# ike
ACMEPACKET(ipsec)# ike-interface
ACMEPACKET(ike-interface)#
```

3. Use the **address** parameter to specify the wancom0 address.

```
ACMEPACKET(ike-interface)# address 172.30.55.127
ACMEPACKET(ike-interface)#
```

4. Use the **realm-id** parameter to specify the realm that contains the IP address assigned to this IKEv2 interface.

```
ACMEPACKET(ike-interface)# realm-id MGMT
ACMEPACKET(ike-interface)#
```

5. Use the **ike-mode** parameter to specify the operational mode, either responder (the default) or initiator.

```
ACMEPACKET(ike-interface)# ike-mode initiator
ACMEPACKET(ike-interface)#
```

6. Use the optional interface-specific **sd-authentication-method** parameter to select the method used to authenticate the IKEv2 SA.

   By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

   Two authentication methods are supported.

   shared-password — (the default) uses a PSK that is used to calculate a hash over a block of data.

   certificate — uses an X.509 certificate to digitally sign a block of data.

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

7. If **sd-authentication-method** is shared-password, use the **shared-password** parameter to specify an interface-specific PSK required for password-based IKEv2 authentication.

   By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

```
ACMEPACKET(ike-interface)# shared-password 123ffGGH65900tnojbt=+
ACMEPACKET(ike-interface)#
```

8. If **sd-authentication-method** is certificate, use the **certificate-profile-id** parameter to identify an interface-specific ike-certificate-profile instance that contains identification and validation credentials required for certificate-based IKEv2 authentication.

   By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

```
ACMEPACKET(ike-interface)# certificate-profile-id jojo.net
ACMEPACKET(ike-interface)#
```

9. If DPD has been enabled at the global level, use the **dpd-params-name** parameter to assign a DPD template, an operational set of DPD parameters, to the wancom0 interface.

   If DPD has not been enabled, this parameter can be safely ignored.

```
ACMEPACKET(ike-interface)# dpd-params-name olivier
ACMEPACKET(ike-interface)#
```

10. Use the optional interface-specific **v2-ike-life-seconds** parameter to specify the lifetime (in seconds) for the IKEv2 SAs supported by the wancom0 interface.

By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

Allowable values are within the range 1 through 999999999 (seconds) with a default of 86400 (24 hours).

```
ACMEPACKET(ike-interface)# v2-ike-life-seconds 21600
ACMEPACKET(ike-interface)#
```

11. Use the optional interface-specific **v2-ipsec-life-seconds** parameter to specify the lifetime (in seconds) for the IPsec SAs supported by the wancom0 interface.

    By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

    Allowable values are within the range 1 through 999999999 (seconds) with a default of 28800 (8 hours).

    ```
    ACMEPACKET(ike-interface)# v2-ipsec-life-seconds 7200
    ACMEPACKET(ike-interface)#
    ```

12. Retain the default value for the optional **eap-protocol** parameter.

    The default, and only currently-supported value, eap-radius-passthru, specifies the use of a RADIUS server for Extensible Authentication Protocol (EAP) processing. The SG shuttles incoming and outgoing EAP messages between the remote IKEv2 peer and the RADIUS server.

    ```
    ACMEPACKET(ike-interface)# eap-protocol eap-radius-passthru
    ACMEPACKET(ike-interface)#
    ```

13. Use the optional interface-specific **addr-assignment** parameter to 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, an IRAC (IPsec Remote Access Client) initiates the exchange by requesting an IP address on the gateway's protected network. In response, the gateway, referred to as an IRAS (IPsec Remote Access Server), returns a local address for the IRAC's use.

    By default, this parameter inherits the value set at the IKEv2 global level. The global level can be over-ridden at the interface level.

    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

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

14. Use **done**, **exit**, and **verify-config** to complete initial wancom0 configuration.

Tunnel Origination Parameters Configuration

If you have set the IKEv2 mode to initiator, and want to enable the automatic re-establishment of IPsec tunnels on the wancom0 interface during system restart or boot, you must next configure a tunnel-orig-params configuration element, which contains the information necessary to re-establish IPsec tunnels.

Use the following procedure to configure a tunnel-orig-params configuration element.

1. From superuser mode, use the following command sequence to access tunnel-orig-params configuration mode. While in this mode, you define remote tunnel endpoints.

   ```
   ACMEPACKET# configure terminal
   ACMEPACKET(configure)# security
   ACMEPACKET(security)# ike
   ACMEPACKET(ike)# tunnel-orig-params
   ACMEPACKET(tunnel-orig-params)#
   ```

2. Use the **name** parameter to identify this instance of the tunnel-orig-params configuration element.

```
ACMEPACKET(tunnel-orig-params)# name syslog
ACMEPACKET(tunnel-orig-params)#
```

3. Use the **remote-addr** parameter to identify the remote IKEv2 peer at the remote end of the IPsec tunnel.

```
ACMEPACKET(tunnel-orig-params)# remote-addr 192.168.34.90
ACMEPACKET(tunnel-orig-params)#
```

4. Use the **retry-limit** parameter to specify the maximum number of tunnel initiation attempts.

   Allowable values are within the range 1 through 5, with a default value of 3.

```
ACMEPACKET(tunnel-orig-params)# retry-limit 5
ACMEPACKET(tunnel-orig-params)#
```

5. Use the **retry-time** parameter to specify the interval (in seconds) between tunnel initiation attempts.

   Allowable values are within the range 5 through 60 (seconds), with a default value of 10.

```
ACMEPACKET(tunnel-orig-params)# retry-time 24
ACMEPACKET(tunnel-orig-params)#
```

6. Use **done**, **exit**, and **verify-config** to complete configuration of this instance of a tunnel-orig-params configuration element.
7. If necessary, repeat Steps 1 through 9 to configure additional tunnel-orig-params instances.

Assign tunnel-orig-params to wancom0

Use the following procedure, which assigns one or more tunnel-orig-params to the wancom0 interface, to complete wancom0 configuration.

1. From super mode, use the following command sequence to access the wancom0 interface.

```
ACMEPACKET# configure terminal
ACMEPACKET(configure)# security
ACMEPACKET(security)# ike
ACMEPACKET(ipsec)# ike-interface
ACMEPACKET(ike-interface)# select
<address>:
172.30.1.150
172.30.1.151
172.30.55.127
selection: 3
ACMEPACKET(ike-interface)#
```

2. Use the **tunnel-orig-name-list** parameter to assign one or more tunnel-orig-params instances (up to a maximum of 10) to the wancom0 interface.

   Each instance specifies the remote end of a single IPsec tunnel.

   Identify tunnel-orig-params instances by name; enclose multiple entries with quotation marks’

```
ACMEPACKET(ike-interface)# tunnel-orig-name-list syslog SFTPserver SNMP-1
SNMP-2 auditLog keyStore
ACMEPACKET(ike-interface)#
```

3. Use **done**, **exit**, and **verify-config** to complete configuration of the wancom0 interface.

SNMP Alarm

If any or all of the tunnels designated by a tunnel-orig-params configuration element fail to establish after the first attempt, the SBC makes **retry-limit** attempts to establish the tunnel(s) with an interval of **retry-time** seconds between each initiation attempt.

If the tunnels fail to establish after the retry limit is reached, the SBC issues an apSecurityTunnelFailureNotification with a supported value of initiator-timeout assigned to the apSecurityFailureCause field.

After issuing the alarm the SBC makes no further attempts to initiate tunnels until the next reboot or restart.

Tunnel Management with the ACLI

The ACLI provides commands to re-initiate or to delete a specific wancom0 IPsec tunnels.

To initiate tunnels:

```
ACMEPACKET# security ike initiate-tunnel <wancom0-IP-address>
```

Initiates the same sequence for establishing wancom0 IKEv2 initiator tunnels as occurs during system boot.

To delete a specific tunnel:

```
ACMEPACKET# security ipsec delete tunnel <remote-IP-address> <spi>
```

- remote-IP-address is the address of the IKEv2 peer at the remote end of the tunnel
- spi is the security parameter index (SPI) — part of the SA negotiated by the endpoint peers.

Use the **show security ipsec sad wancom0 brief** command to display the SPI

Hardware Requirements

IPsec tunnel establishment on the wancom0 management interface requires the presence of an IPsec NIU and an SSM2.

IKEv2 Security Association Configuration

Use the following procedure to create an IKEv2 SA that identifies cryptographic material available for IPsec tunnel establishment. You will later assign this IKEv2 SA to an IPsec Security Policy.

1. From superuser mode, use the following command sequence to access ike-sainfo configuration mode. While in this mode, you configure global IKEv2 SAs.

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

2. Use the required **name** parameter to provide a unique identifier for this ike-sainfo instance.

   name enables the creation of multiple ike-sainfo instances.

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

3. Use the **security-protocol** parameter to specify the IPsec security (authentication and encryption) protocols supported by this SA.

   The following security protocols are available.

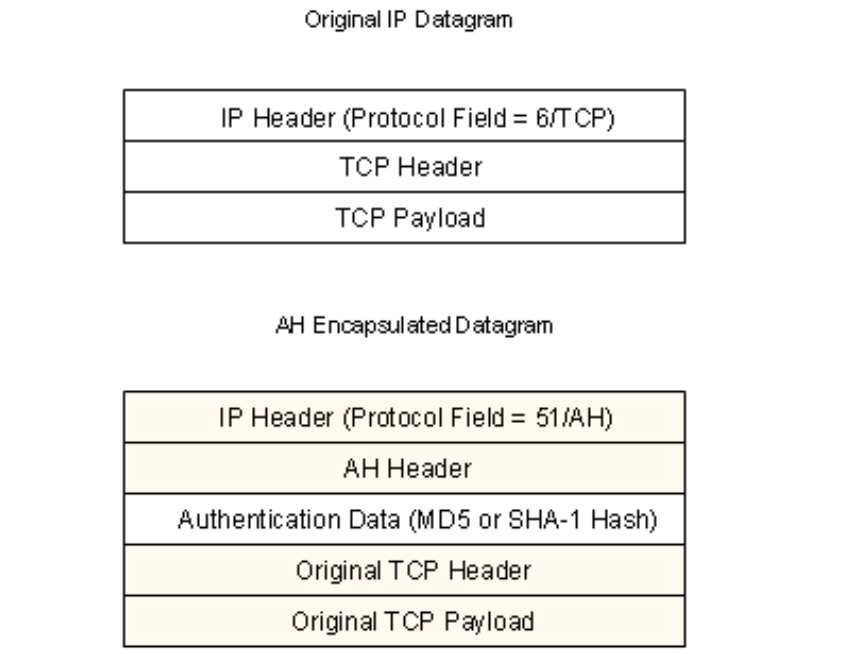
   Authentication Header (AH) — the default value — as defined by RFC 4302, IP Authentication Header, which provides authentication integrity to include the mutual identification of remote peers, non-repudiation of received traffic, detection of data that has been altered in transit, and detection of data that has been replayed, that is copied and then re-injected into the data stream at a later time. Authentication services utilize the authentication algorithm specified by the **auth-algo** parameter.

   Encapsulating Security Payload (ESP) as defined by RFC 4303, IP Encapsulating Security Payload, which provides both authentication and privacy services. Privacy services utilize the encryption algorithm specified by the **encryption-algo** parameter.

   ESP-AUTH (also RFC 4303-based), which supports ESP's optional authentication.

   ESP-NULL (also RFC 4303-based) which proves NULL encryption as described in RFC 2410, The NULL Encryption Algorithm and Its Use With IPsec. This option provides no privacy services, and is not recommended for production environments.

   Refer to the following figures for additional details.



AH Transport Mode -- missing picture

Original IP Datagram

IP Header (Protocol Field = 6/TCP)
TCP Header
TCP Payload

AH Encapsulated Datagram

New IP Header (Protocol Field = 51/AH)
AH Header
Authentication Data (MD5 or SHA-1 Hash)
Original IP Header
Original TCP Header
Original TCP Payload

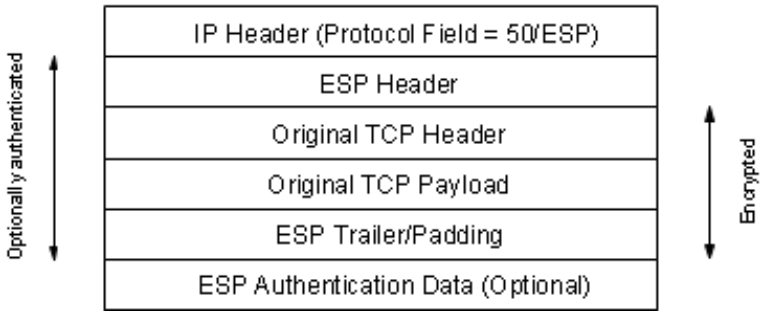
Authenticated data, note that TOS, Flags, Fragmentation, TTL, and Header Checksum fields of the IP Header are not covered by the authentication calculation.

AH Tunnel Mode

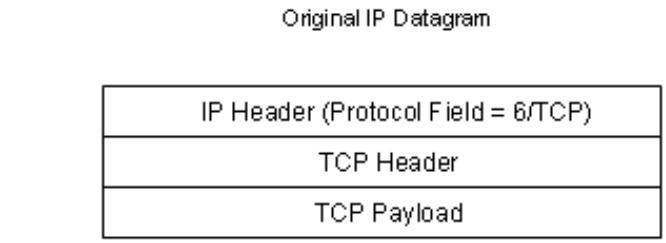
Original IP Datagram

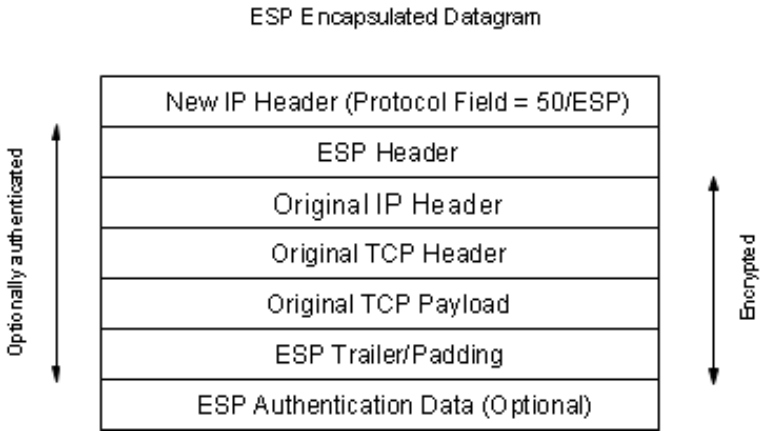
IP Header (Protocol Field = 6/TCP)
TCP Header
TCP Payload

ESP Encapsulated Datagram



ESP Transport Mode





ESP Tunnel Mode

```
ACMEPACKET(ike-sainfo)# security-protocol esp
ACMEPACKET(ike-sainfo)#
```

4. Use the **auth-algo** parameter to specify the authentication algorithms supported by this SA.

   The following authentication protocols are available

   - Message Digest Algorithm 5 (md5) — as defined by RFC 1321, The MD5 Message-Digest Algorithm.
   - Secure Hash Algorithm (sha) — as defined by FIPS PUB 180-1, Secure Hash Standard.
   - any (the default) — supports both MD5 and SHA-1.

   ```
   ACMEPACKET(ike-sainfo)# auth-algo md5
   ACMEPACKET(ike-sainfo)#
   ```

5. Use the **encryption-algo** parameter to specify the encryption algorithms supported by this SA.

   The following encryption protocols are available

   - Triple DES (3des) — as defined by ANSI X.9.52 1998, Triple Data Encryption Algorithm Modes of Operation.
   - Advanced Encryption Standard (aes) — FIPS PUB 197, Advanced Encryption Standard.
   - NULL Encryption (null) — as described in RFC 2410, The NULL Encryption Algorithm and Its Use With IPsec. This option provides no privacy services, and is not recommended for production environments.
   - any (the default) — supports all listed encryption protocols.

   ```
   ACMEPACKET(ike-sainfo)# encryption-algo aes
   ACMEPACKET(ike-sainfo)#
   ```

6. Use the **ipsec-mode** parameter to specify the IPSec operational mode.

   Transport mode (the default) provides a secure end-to-end connection between two IP hosts. Transport mode encapsulates the IP payload.

   Tunnel mode provides VPN service where 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.

Refer to the previous figures for encapsulation details.

```
ACMEPACKET(ike-sainfo)# ipsec-mode tunnel
ACMEPACKET(ike-sainfo)#
```

7. If **ipsec-mode** is tunnel, use the required **tunnel-local-addr** parameter to specify the IP address of the local IKEv2 interface that terminates the IPsec tunnel.

   This parameter can safely be ignored if **ipsec-mode** is transport.

```
ACMEPACKET(ike-sainfo)# tunnel-local-addr 192.169.204.14
ACMEPACKET(ike-sainfo)#
```

8. If **ipsec-mode** is tunnel, use the **tunnel-remote-addr** parameter to 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.

   This parameter can safely be ignored if **ipsec-mode** is transport.

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

9. Use **done**, **exit**, and **verify-config** to complete configuration of IKEv2 SA.
10. If necessary, repeat Steps 1 through 9 to configure additional IKEv2 SAs.

Security Policy Configuration

Use the following procedure to assign an IKEv2 SA to an existing Security Policy. Note that the network interface supported by the Security Policy must be the wancom0 management interface

1. From superuser mode, use the following command sequence to access security-policy configuration mode. While in this mode, you configure security policies.

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

2. Use the **ike-sainfo-name** parameter to assign an IKEv2 SA to this Security Policy.

```
ACMEPACKET(security-policy)# ike-sainfo-name SA-1
ACMEPACKET(security-policy)#
```

3. Use **done**, **exit**, and **verify-config** to complete configuration of this Security Policy.

   The following sample security policies support IKEv2 over the wancom0 management interface. The first policy (ikepol) opens port 500, while the second policy (poll) specifies IPsec on all other ports.

```
ACMEPACKET# show running-config security-policy
security-policy

       name                                ikepol
       network-interface                   W00:0
       priority                            0
       local-ip-addr-match                 172.30.55.127
       remote-ip-addr-match                172.30.89.11
       local-port-match                    500
       remote-port-match                   500
       trans-protocol-match                ALL
       direction                           both
       local-ip-mask                       255.255.255.255
       remote-ip-mask                      255.255.255.255
       action                              allow
       ike-sainfo-name
       outbound-sa-fine-grained-mask
              local-ip-mask                255.255.255.255
```

```
                remote-ip-mask              255.255.255.255
                local-port-mask             0
                remote-port-mask            0
                trans-protocol-mask         0
                valid                       enabled
                vlan-mask                   0xFFF
        last-modified-by                    admin@console
        last-modified-date                  2009-11-11 19:06:32

security-policy
        name                                pol1
        network-interface                   W00:0
        priority                            1
        local-ip-addr-match                 172.30.89.10
        remote-ip-addr-match                172.30.89.11
        local-port-match                    0
        remote-port-match                   0
        trans-protocol-match                ALL
        direction                           both
        local-ip-mask                       255.255.255.255
        remote-ip-mask                      255.255.255.255
        action                              ipsec
        ike-sainfo-name                     ikesa1
        outbound-sa-fine-grained-mask
                local-ip-mask               255.255.255.255
                remote-ip-mask              255.255.255.255
                local-port-mask             0
                remote-port-mask            0
                trans-protocol-mask         0
                valid                       enabled
                vlan-mask                   0xFFF
        last-modified-by                    admin@console
        last-modified-date                  2009-11-11 19:07:03
```

4

TACACS+

TACACS+ (Terminal Access Controller Access Control System Plus) is a protocol originally developed by Cisco Systems, and made available to the user community by a draft RFC, *TACACS+ Protocol, Version 1.78* (draft-grant-tacacs-02.txt). TACACS+ provides AAA (Authentication, Authorization, and Accounting) services over a secure TCP connection using Port 49.

TACACS+ Overview

Like DIAMETER and RADIUS, TACACS+ uses a client/server model in which a Network Access Server (NAS) acts in the client role and a TACACS+ equipped device (a daemon in TACACS+ nomenclature) assumes the server role. For purposes of the current implementation, the Oracle Oracle Communications Session Border Controller functions as the TACACS+ client. Unlike RADIUS, which combines authentication and authorization, TACACS+ provides three distinct applications to provide finer grade access control.

Authentication is the process that confirms a user's purported identity. Authentication is most often based on a simple username/password association, but other, and more secure methods, are becoming more common. The following authentication methods are support by the current implementation: simple password, PAP (Protocol Authentication Protocol), and CHAP (Challenge Handshake Authentication Protocol).

Authorization is the process that confirms user privileges. TACACS+ can provide extremely precise control over access to system resources. In the current implementation, TACACS+ controls access to system administrative functions.

TACACS+ provides secure communication between the client and daemon by encrypting all packets. Encryption is based on a shared-secret, a string value known only to the client and daemon. Packets are encrypted in their entirety, save for a common TACACS+ header.

The cleartext header contains, among other fields, a version number, a sequence number. and a session ID. Using a methodology described in Section 5 of the TACACS+ draft RFC, the sender encrypts outbound cleartext messages by repetitively running the MD5 hash algorithm over the concatenation of the session ID, shared-secret, version number, and sequence number values, eventually deriving a virtual one-time-pad of the same length as the message body. The sender encrypts the cleartext message with an XOR (Exclusive OR) operation, using the cleartext message and virtual one-time-pad as inputs.

The message recipient, who possesses the shared-secret, can readily obtain the version number, sequence number, session ID, and message length from the cleartext header. Consequently, the recipient employs the same methodology to derive a virtual one-time-pad identical to that derived by the sender. The recipient decrypts the encrypted message with an XOR operation, using the encrypted message and virtual one-time-pad as inputs.

Details on the TACACS+ functions and configuration can be found in the Oracle Communications Session Border Controller ACLI Configuration Guide.

The TACACS+ implementation is based upon the following internet draft.

draft-grant-tacacs-02.txt, *The TACACS+ Protocol Version 1.78*

Other relevant documents include

RFC 1321, *The MD-5 Message Digest Algorithm*

RFC 1334, *PPP Authentication Protocols* .

RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*

Glossary