

**Oracle® Communications Session
Border Controller**

Administrative Security Essentials Guide

Release S-CX6.3.0

Formerly Net-Net Session Director

October 2013

Copyright ©2013, 2011, Oracle and/or its affiliates. All rights reserved.

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

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

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

About This Guide	v
Overview	v
Audience	v
Supported Platforms.....	v
Related Documentation	v
Document Revision History	vi
1 Access	9
Authentication and Authorization	9
Local Authentication and Authorization	9
Console Login	10
Serial Port Control	10
Initial Login	11
Remote SSH Login with Password	12
Remote SSH Login with Public Key	13
RADIUS Authentication and Authorization	16
RADIUS Authorization Classes	17
RADIUS and SSH	17
RADIUS and Password Policies	17
Two-Factor Authentication	18
Login Banner.....	19
Login Policy.....	20
Password Policy	21
Configuring Password Policy Properties	22
RADIUS Passwords	24
Changing a Password.....	24
Changing the user Password	27
Changing the admin Password	27
Changing a Passcode	27
Changing the admin Passcode	28
SSH and SFTP.....	29
SSH Operations	29

Configuring SSH Properties.....	29
Managing SSH Keys.....	30
SFTP Operations.....	39
2 Audit Log.....	43
Audit Log Format.....	43
Viewing the Audit Log	46
Audit Log Samples.....	46
Configuring the Audit Log.....	49
Configuring SFTP Audit Log Transfer.....	51
Configuring SFTP Servers	51
Audit Log Alarms and Traps	53
3 Internet Key Exchange (IKEv2).....	55
IKEv2 Overview	55
IKEv2 Global Configuration.....	56
DPD Configuration	61
Certificate Profile Configuration.....	63
Certificate Chain Validation.....	64
ACLI verify-config Command.....	64
Hardware Requirements	64
Data Flow Configuration	65
Local Address Pool Configuration	66
wancom0 Management Interface Configuration.....	67
Tunnel Origination Parameters Configuration	70
SNMP Alarm.....	71
Tunnel Management with the ACLI.....	71
Hardware Requirements	72
IKEv2 Security Association Configuration	72
Security Policy Configuration	76
4 Licensing	79
Installation/Deletion Implications	79

About This Guide

Overview

This guide explains support for a Administrative Security License (*Admin Security*), which provides 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 license. Users of SBCs without an Admin Security license can safely ignore this guide.

Specific topics covered in this guide include

- Access
- Audit Log
- IKEv2
- 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.

Supported Platforms

Release Version S-CX6.3.0 is supported on the Acme Packet 4500 and Acme Packet 3800 series platforms.

Related Documentation

The following table lists the members that comprise the documentation set for this release:

Document Name	Document Description
Acme Packet 4500 System Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 4500 system.
Acme Packet 3800 Hardware Installation Guide	Contains information about the components and installation of the Acme Packet 3800 system.
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 SBC.
ACLI Reference Guide	Contains explanations of how to use the ACLI, as an alphabetical listings and descriptions of all ACLI commands and configuration parameters.

Document Name	Document Description
Maintenance and Troubleshooting Guide	Contains information about SBC 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), Enterprise MIBs, general trap information, including specific details about standard traps and enterprise traps, Simple Network Management Protocol (SNMP) GET query information (including standard and enterprise SNMP GET query names, object identifier names and numbers, and descriptions), examples of scalar and table objects.
Accounting Guide	Contains information about the SBC's accounting support, including details about RADIUS accounting.
HDR Resource Guide	Contains information about the SBC's Historical Data Recording (HDR) feature. This guide includes HDR configuration and system-wide statistical information.
Administrative Security Essentials	Contains information about the SBC's support for its Administrative Security license.

Document Revision History

This section contains a revision history for this document.

Date	Revision Number	Description
April 30, 2012	V1.00	Version release covering licensing, features, and functions related to the administrative security.

The Admin Security License restricts local system access to the console (serial) port; telnet access is denied. Remote users are restricted to encrypted Secure Shell (SSH) access. File system access is enabled by Secure Shell File Transfer Protocol (SFTP).

The following sections describe

- Authentication and authorization
- Login policy
- Password policy
- SSH
- SFTP

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 Acme Packet Net-Net 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 Acme Packet Net-Net 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

<u>User Name</u>	<u>Logins into/prompt</u>	<u>Authentication</u>	<u>Authorization</u>
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 Net-Net SBC reverts to its default behavior by providing serial port access.

To turn off access to the serial port:

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

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 acknowledgement 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.

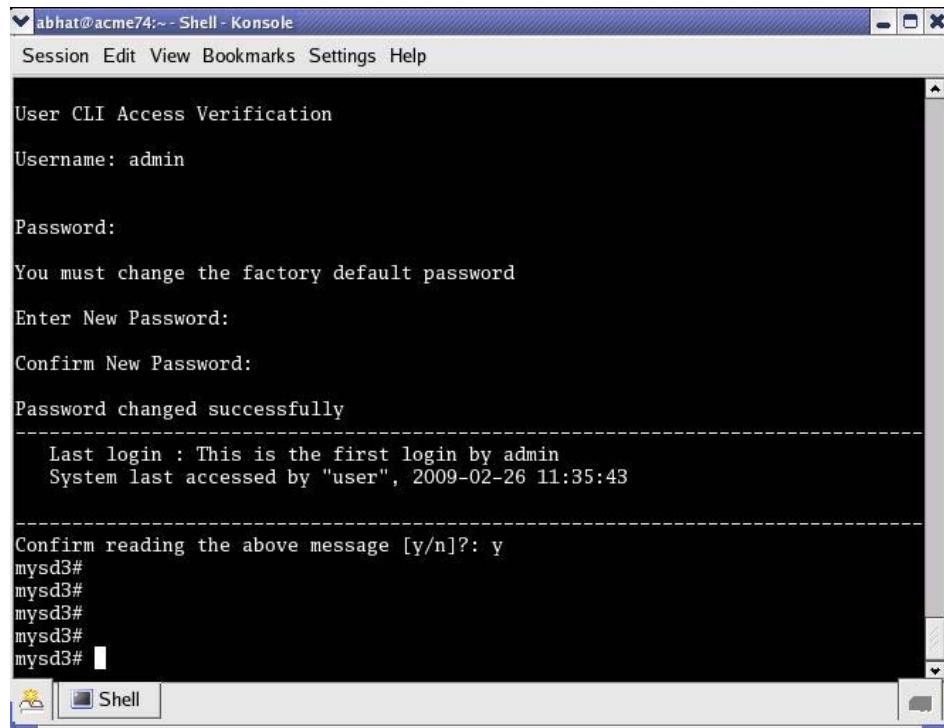


Figure 1: Initial admin Login (Console Access)

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

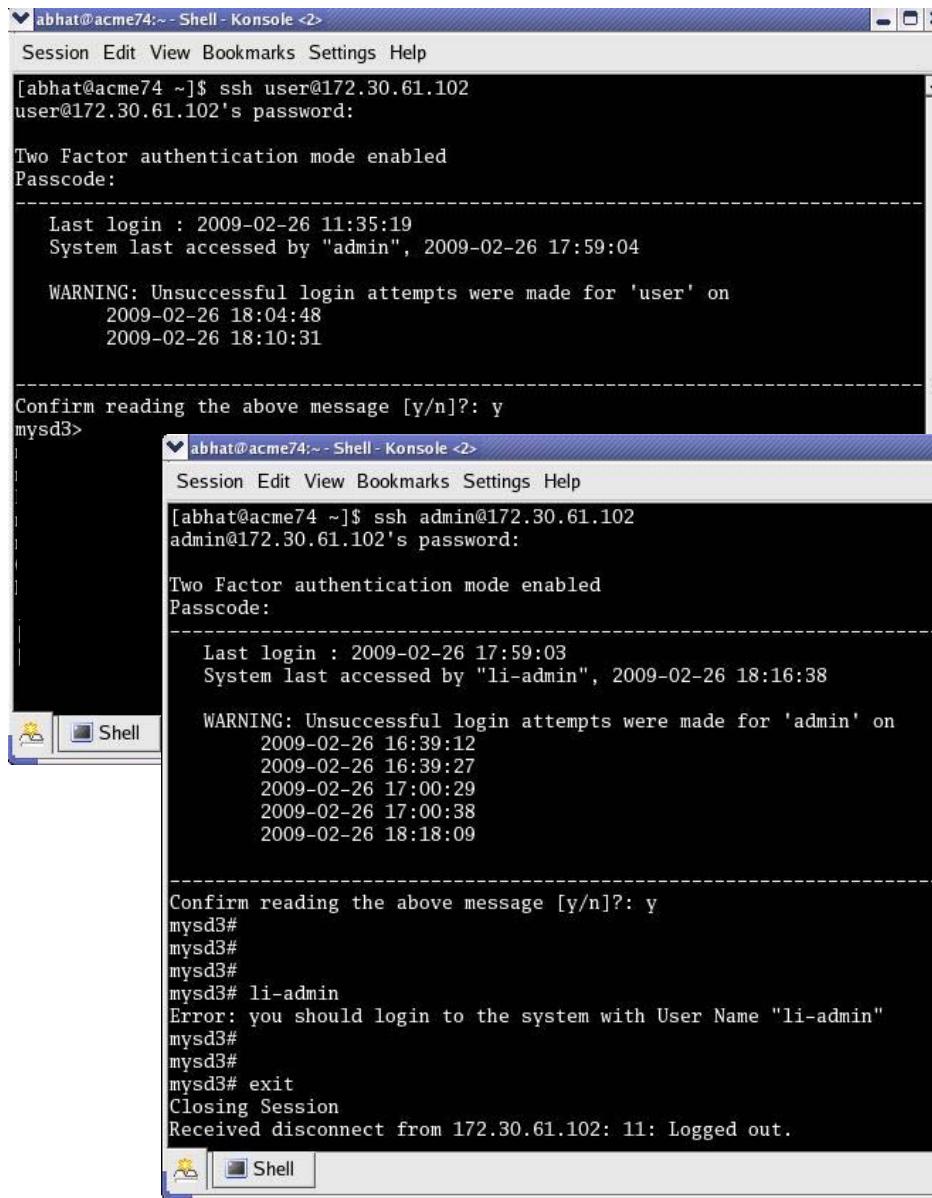
- passwords cannot contain, repeat, or reverse the user name
- passwords cannot contain three consecutive identical characters

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*)



The screenshot shows two terminal windows side-by-side. Both windows are titled 'abhat@acme74:~ - Shell - Konsole <2>'. The top window shows a login for 'user' at 172.30.61.102. It prompts for a password and displays a two-factor authentication message. It then shows a warning about unsuccessful login attempts for 'user' on 2009-02-26 at 18:04:48 and 18:10:31. The bottom window shows a login for 'admin' at 172.30.61.102. It also prompts for a password and displays a two-factor authentication message. It shows a warning about unsuccessful login attempts for 'admin' on 2009-02-26 at 16:39:12, 16:39:27, 17:00:29, 17:00:38, and 18:18:09. Both windows then prompt the user to confirm reading the banner message, with 'y' entered in the bottom window. Finally, the bottom window shows the user attempting to log in as 'li-admin', which is rejected, and then exiting the session.

```

[abhat@acme74 ~]$ ssh user@172.30.61.102
user@172.30.61.102's password:
Two Factor authentication mode enabled
Passcode:
-----
Last login : 2009-02-26 11:35:19
System last accessed by "admin", 2009-02-26 17:59:04

WARNING: Unsuccessful login attempts were made for 'user' on
2009-02-26 18:04:48
2009-02-26 18:10:31

-----
Confirm reading the above message [y/n]?: y
mysd3>
[abhat@acme74 ~]$ ssh admin@172.30.61.102
admin@172.30.61.102's password:
Two Factor authentication mode enabled
Passcode:
-----
Last login : 2009-02-26 17:59:03
System last accessed by "li-admin", 2009-02-26 18:16:38

WARNING: Unsuccessful login attempts were made for 'admin' on
2009-02-26 16:39:12
2009-02-26 16:39:27
2009-02-26 17:00:29
2009-02-26 17:00:38
2009-02-26 18:18:09

-----
Confirm reading the above message [y/n]?: y
mysd3#
mysd3#
mysd3#
mysd3# li-admin
Error: you should login to the system with User Name "li-admin"
mysd3#
mysd3#
mysd3# exit
Closing Session
Received disconnect from 172.30.61.102: 11: Logged out.

```

Figure 2: Remote SSH Login

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

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

<u>User Name</u>	<u>Logins into/prompt</u>	<u>Authentication</u>	<u>Authorization</u>
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 Acme Packet Net-Net 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 to 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 authori zed-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

```
ragnarok# ssh-pub-key import authori zed-key Dwight
ragnarok#
```

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

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

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.

```
ragnarok# 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"  
AAAAB3NzaC1yc2EAAAABIwAAEAXcYTV595VqdHy12P+mI ZBI pe0Zx9sX/mSAFihDJYdL  
qJlWdiZuSmny8HZI xTlC6na62i D25ml EdyLhI Y0uknkYBCU7UsLwmx4dLDyHTbrQHz3b1q  
3Tb8auz97/J1p4pw39PT42CoR0DzPBrXJV+0gI NE/83C1y0SSJ8Bj C9LEwE=
```

---- 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

```
ragnarok# 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'.

```
ragnarok# activate-config
```

Activate-Config received, processing.

waiting for request to finish

SD is not QOS-capable

Request to 'ACTIVATE-CONFIG' has finished,

Activate Complete

```
ragnarok#
```

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.

```

abhat@acme74:~ - Shell - Konsole <2>
Session Edit View Bookmarks Settings Help
[abhat@acme74 ~]$ ssh Matilda@172.30.61.102
Last login : 2009-02-26 18:48:32
System last accessed by "Matilda", 2009-02-26 18:48:36

Confirm reading the above message [y/n]? y
mysd3#
mysd3# config
mysd3(config)
mysd3(config) [abhat@acme74 ~]$ ssh Dwight@172.30.61.102
mysd3#
mysd3# Last login : 2009-02-26 19:10:09
mysd3# exit
Closing Session
Received disconnect from 172.30.61.102 port 22: Normal termination of session.
[abhat@acme74 ~]$ Confirm reading the above message [y/n]? y
mysd3>
mysd3>

```

Figure 3: SSH with Public Key Login

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

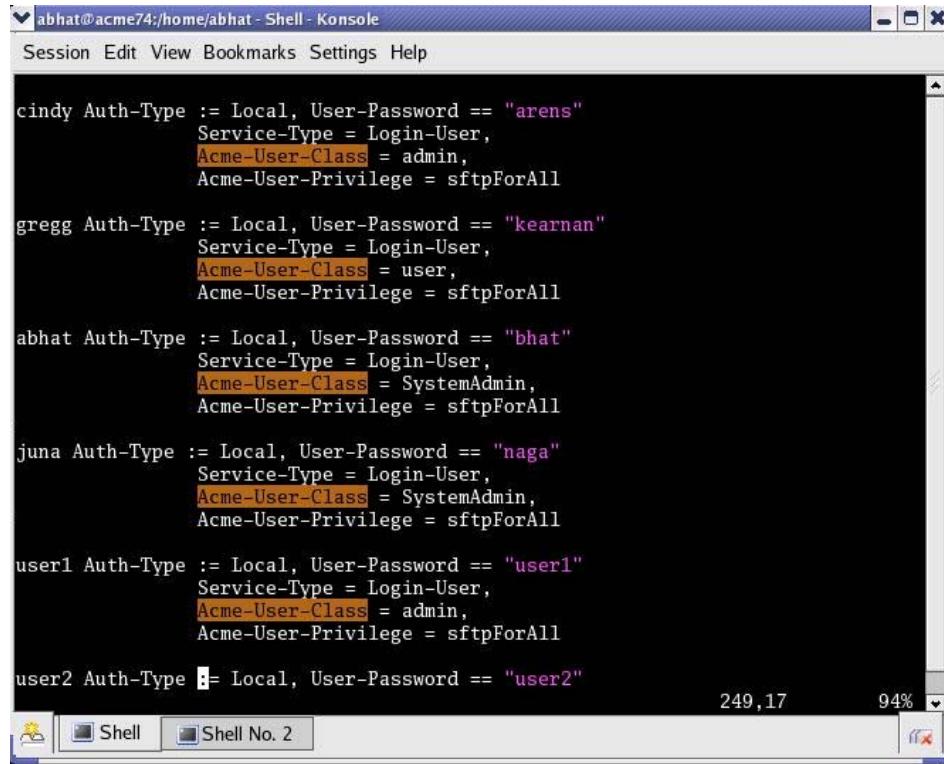
User Name	Logins into/prompt	Authentication	Authorization
	user mode		authorized locally by SBC
	>		authorization determined by
not relevant	or	authenticated locally by SBC via SSH public key	authorizationClass command argument (user or admin) inherits access/privilege defined by the specified class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

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 *Net-Net 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.



```
cindy Auth-Type := Local, User-Password == "ahrens"
  Service-Type = Login-User,
  Acme-User-Class = admin,
  Acme-User-Privilege = sftpForAll

gregg Auth-Type := Local, User-Password == "kearnan"
  Service-Type = Login-User,
  Acme-User-Class = user,
  Acme-User-Privilege = sftpForAll

abhat Auth-Type := Local, User-Password == "bhat"
  Service-Type = Login-User,
  Acme-User-Class = SystemAdmin,
  Acme-User-Privilege = sftpForAll

juna Auth-Type := Local, User-Password == "naga"
  Service-Type = Login-User,
  Acme-User-Class = SystemAdmin,
  Acme-User-Privilege = sftpForAll

user1 Auth-Type := Local, User-Password == "user1"
  Service-Type = Login-User,
  Acme-User-Class = admin,
  Acme-User-Privilege = sftpForAll

user2 Auth-Type := Local, User-Password == "user2"
```

Figure 4: RADIUS Users File

Upon receiving a login request, the Net-Net SBC send 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 Acme Packet 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.

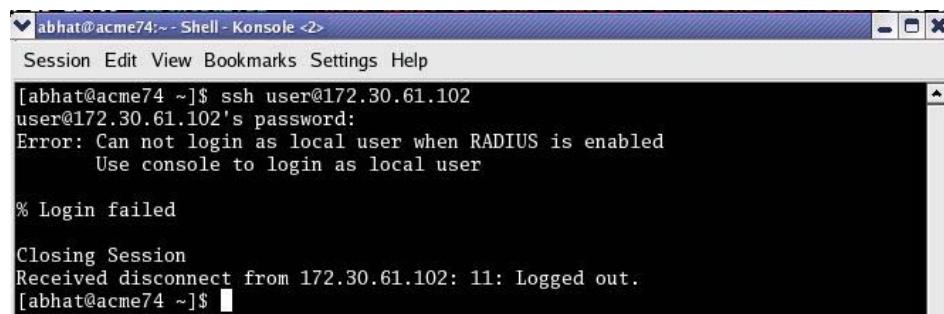
RADIUS Authorization Classes

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.



The screenshot shows a terminal window titled "abhat@acme74:~ - Shell - Konsole <2>". The window has a menu bar with "Session", "Edit", "View", "Bookmarks", "Settings", and "Help". The terminal content is as follows:

```
[abhat@acme74 ~]$ ssh user@172.30.61.102
user@172.30.61.102's password:
Error: Can not login as local user when RADIUS is enabled
Use console to login as local user

% Login failed

Closing Session
Received disconnect from 172.30.61.102: 11: Logged out.
[abhat@acme74 ~]$
```

Figure 5: Local User Login with SSH (RADIUS Enabled)

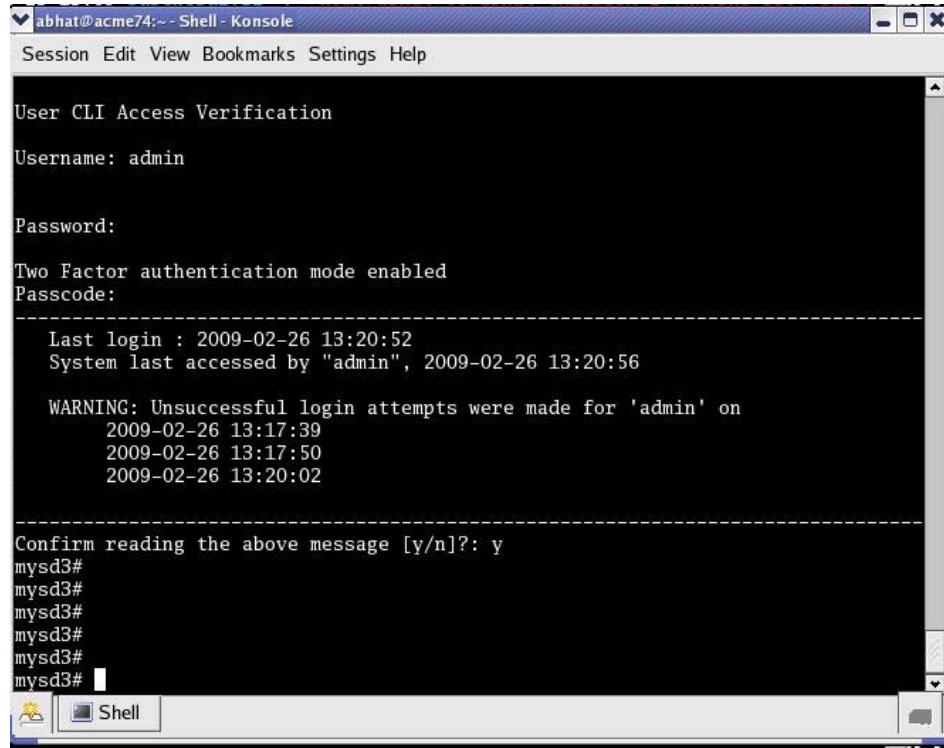
If you want to enable user and admin access via SSH with RADIUS configured, you must explicitly define users on the RADIUS server with appropriate *Acme-User-Class*.

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..



The screenshot shows a terminal window titled 'abhat@acme74:~ - Shell - Konsole'. The window displays the following text:

```
User CLI Access Verification
Username: admin

Password:
Two Factor authentication mode enabled
Passcode:
-----
Last login : 2009-02-26 13:20:52
System last accessed by "admin", 2009-02-26 13:20:56

WARNING: Unsuccessful login attempts were made for 'admin' on
2009-02-26 13:17:39
2009-02-26 13:17:50
2009-02-26 13:20:02

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

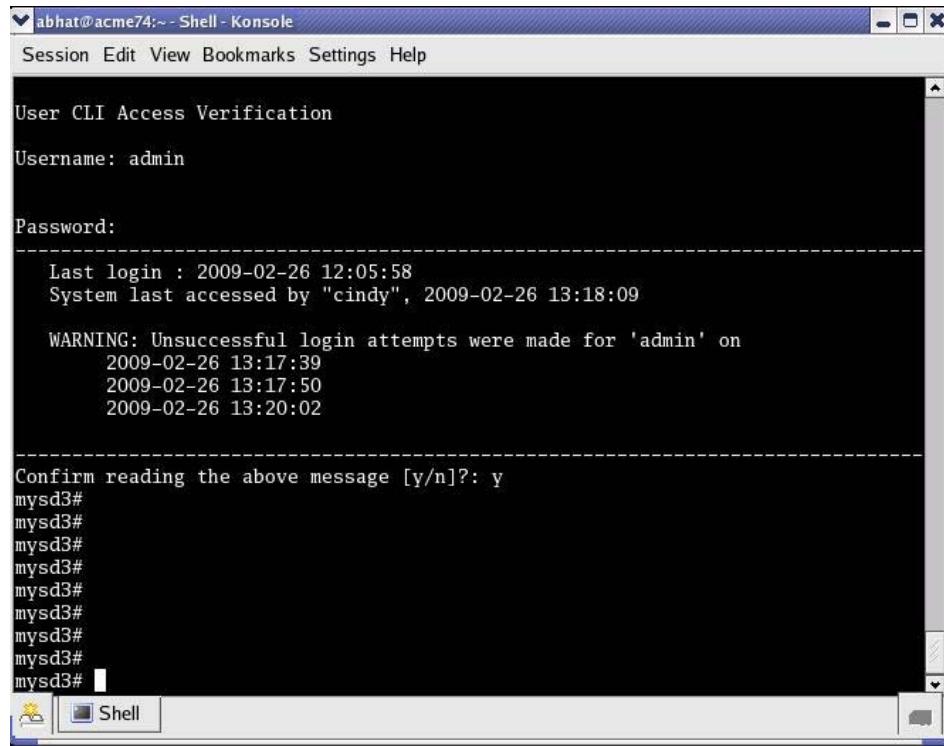
Figure 6: Two-Level Authentication

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.

Login Banner

Upon successful user authentication/authorization, the Acme Packet Net-Net SBC displays the login banner.



The screenshot shows a terminal window titled 'abhat@acme74:~ - Shell - Konsole'. The window contains the following text:

```
User CLI Access Verification
Username: admin

Password:
-----
Last login : 2009-02-26 12:05:58
System last accessed by "cindy", 2009-02-26 13:18:09

WARNING: Unsuccessful login attempts were made for 'admin' on
2009-02-26 13:17:39
2009-02-26 13:17:50
2009-02-26 13:20:02

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

Figure 7: 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:

```
ragnarok# configure terminal > security > admin-security > login-config
ragnarok(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.

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

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

```
ragnarok(login-config)# login-attempt-interval 6
ragnarok(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.

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

```
ragnarok(I oги n-confi g)# send-alarm di sabl ed
ragnarok(I oги n-confi g)#
```
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.

```
ragnarok(I oги n-confi g)# I oги n-auth-mode two-factor
ragnarok(I oги n-confi g)#
```
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.

```
ragnarok(I oги n-confi g)# enabl e-l oги n-banner di sabl e
ragnarok(I oги n-confi g)#
```

A sample login policy configuration appears below:

```
ragnarok(I oги n-confi g)# concurrent-sessi on l imi t 4
ragnarok(I oги n-confi g)# max-l oги n-attempts 5
ragnarok(I oги n-confi g)# l oги n-attempt-i nterval 6
ragnarok(I oги n-confi g)# l ockout-i nterval 30
ragnarok(I oги n-confi g)# done
ragnarok(I oги n-confi g)# exi t
ragnarok(admi n-securi ty)#

```

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

The Admin Security license supports the creation of a Password Policy that enhance the authentication process by imposing password length and strength requirements, password history and re-use requirements, and password expiration requirements.

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 Password Policy.

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

```
ragnarok# configure terminal > security > password-policy  
ragnarok(password-policy)#

```

The **password-policy** configuration element properties are shown below with their default values.

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

2. **min-secure-pwd-length**—is ignored when the *Admin Security* license is installed

The license mandates the following password 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

3. **expiry-interval**—specifies the password or SSH public key lifetime in days

expiry-interval applies to both passwords and SSH public keys used to identify an SSH user client. Password lifetime tracking begins when a password is changed; SSH public key lifetime tracking begins when the key is imported to the Net-Net SBC.

Allowable values are integers within the range 1 through 65535, with a default of 90 days.

Retain the default value, or specify a new password lifetime.

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

```

4. **expiry-notify-period**—specifies the number of days prior to expiration that users begin to receive password or SSH public key expiration reminders

expiry-notify-period applies to both passwords and SSH public keys used to identify an SSH user client.

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

During the notify period, users are reminded of impending password/SSH public expiration at both device login and logout.

Retain the default value, or specify a new notification start date.

```
ragnarok(password-pol i cy)# expi ry-noti fy-peri od 10
ragnarok(password-pol i cy)#

```

5. **grace-period**—in conjunction with **grace-logins**, limits user access after password or SSH public key expiration

grace-period applies to both passwords and SSH public keys used to identify an SSH user client.

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

After password or SSH public key expiration, users are granted some number of logins (specified by the **grace-logins** property) for some number of days (specified by this property). Once the number of logins has been exceeded, or once the grace time period has elapsed, the user is forced to change his or her password, or to obtain a new SSH public key.

Retain the default value, or specify a new grace period.

```
ragnarok(password-pol i cy)# grace-peri od 5
ragnarok(password-pol i cy)#

```

6. **grace-logins**—in conjunction with **grace-period**, limits user access after password or SSH public key expiration

grace-logins applies to both passwords and SSH public keys used to identify an SSH user client.

Allowable values are integers within the range 1 through 10, with a default value of 3 logins.

After password or SSH public key expiration, users are granted some number of logins (specified by this property) for some number of days (specified by the **grace-period** property). Once the number of logins has been exceeded, or once the grace time period has elapsed, the user is forced to change his or her password, or to obtain a new SSH public key.

Retain the default value, or specify a new number of allowed logins after password or SSH public key expiration.

```
ragnarok(password-pol i cy)# grace-l ogi ns 1
ragnarok(password-pol i cy)#

```

7. **password-history-count**—specifies the number of previously used passwords retained in encrypted format in the password history

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.

Passwords contained with the password history cannot be re-used.

Retain the default value, or specify a new number of retained passwords.

```
ragnarok(password-pol i cy)# password-hi story-count 5
ragnarok(password-pol i cy)#

```

8. **password-change-interval**—specifies a minimum password or SSH public key lifetime

password-change-interval applies to both passwords and SSH public keys used to identify an SSH user client.

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

Specifies the minimum time that must elapse between password or SSH public key changes.

Retain the default value, or specify a new minimum password/SSH public key lifetime in hours.

For example,

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

```

A sample password policy configuration appears below:

```
ragnarok(password-policy)# expiry-interval 60
ragnarok(password-policy)# expiry-notification-period 10
ragnarok(password-policy)# grace-period 5
ragnarok(password-policy)# grace-logs 1
ragnarok(password-policy)# password-history-count 5
ragnarok(password-policy)# done
ragnarok(password-policy)# exit
ragnarok(security)#

```

Defines a password policy that requires password or SSH public key change within 60 days, and begins notification of password or public key expiration 10 days prior to the actual event. Users with expired passwords or public keys are granted a single login over a five day period. Users must change the password or obtain a new public key at the expiration of the grace period, or at a second login (whichever comes first). The policy forbids re-use of the 5 most recently expired passwords, and (by default) sets a minimum period of 24-hours between password or public key changes.

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.

Figure 8: 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.

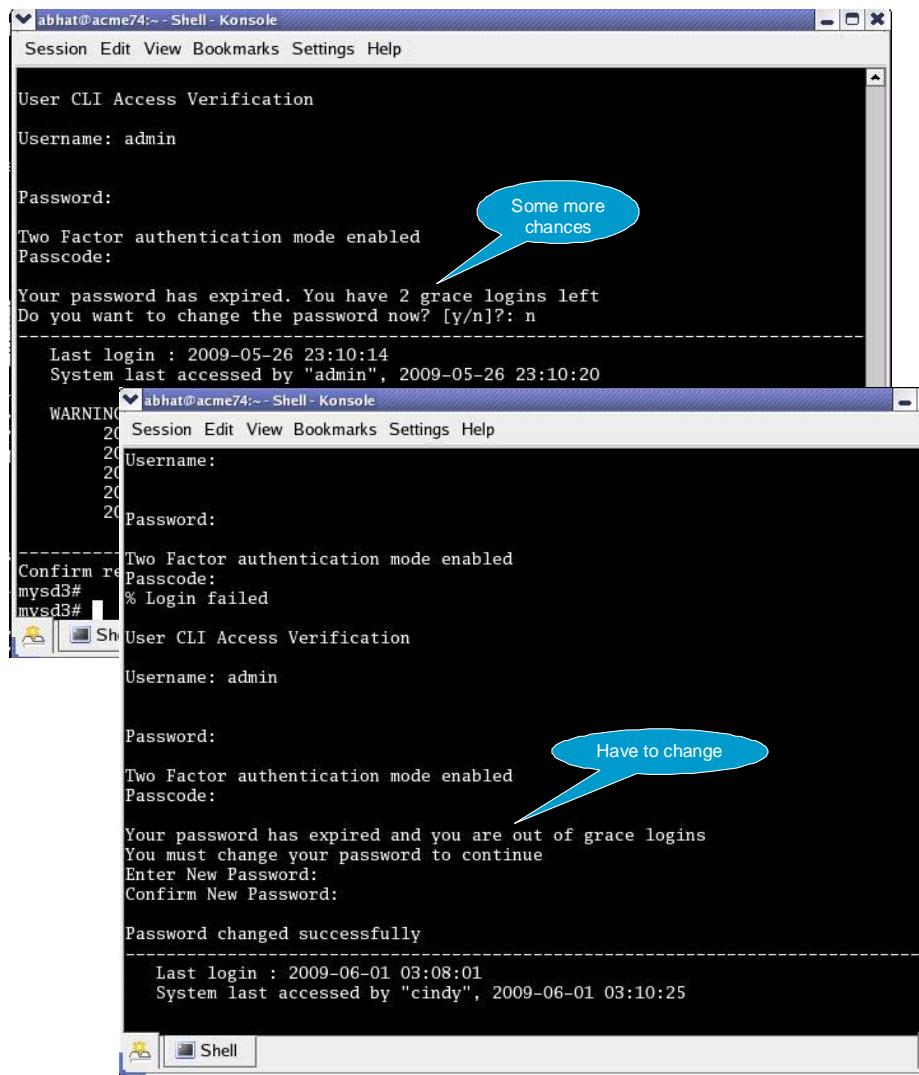


Figure 9: Grace Login Reminder/Forced Password Change

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.

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

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

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

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

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

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.

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

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

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

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

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

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.

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

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

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

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

```
ragnarok# 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.

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

Changing the admin Passcode

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

1. Enter secret enable passcode at the prompt.

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

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

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

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

```
ragnarok# 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.

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

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 Acme Packet Net-Net 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:

```
ragnarok# configure terminal > security > admin-security >
ssh-config
ragnarok(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.

```
ragnarok(ssh-confi g)# rekey-i nterval 20
ragnarok(ssh-confi g)
```

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.

```
ragnarok(ssh-confi g)# rekey-packet-count 24
ragnarok(ssh-confi g)
```

A sample SSH configuration appears below:

```
ragnarok(ssh-confi g)# rekey-i nterval 20
ragnarok(ssh-confi g)# done
ragnarok(ssh-confi g)# exi t
ragnarok(admi n-securi ty)#

```

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

Managing SSH Keys

The following procedures tell you how to import, generate, and view 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 to 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.

```
ragnarok# ssh-pub-key import known-host fedal1ah
```

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.

```
ragnarok# ssh-pub-key import known-host fedal1ah
```

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"  
AAAAB3NzaC1yc2EAAAABIwAAAQEA70Bf08jJe7MSMgerjDTgZpbPbIrx4n17L0JgPC7cI L  
cDGEtKSiVt5MjcsAv3v6AEN2pYZi0xd2Zzi0mp0019kkJ56s/ljGstEzqXMKHKUr9mBV  
qvqlEOTqb0wEi5sz2AP31GUjQTCKZRF1X0Qx8A44vHZCum93/jfNRsnWQ1mhHmaZMmT2LS  
h0r4J/NIp+vpsvpdrolV6Ftz5eiVfgocxrDrjNcVtsAMyLBpDdL6e9XebQzGSS92TPuKP/  
yqzLJ2G5NVFhxdw5i+FvdHz1vBdvB505y20Pj/i z1u3TA/307tyntB0b7beDylrg64Azc8  
G7E3AGiH49LnBt1Qf/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  
ragnarok# 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'.  
ragnarok# activate-config  
Activate-Config received, processing.  
waiting for request to finish  
SD is not QOS-capable  
Request to 'ACTIVATE-CONFIG' has finished,  
Activate Complete  
ragnarok#
```

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 Acme Packet Net-Net 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

```
ragnarok# 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.

```
ragnarok# 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"  
AAAAB3NzaC1yc2EAAABIwAAAI EAxcYTV595VqdHy12P+ml ZBI pe0Zx9sX/mSAFi hDJYdL  
qJl Wdi ZuSmny8HZI xTl C6na62i D25ml EdyLhl Y0uKnkYBCU7UsLwmx4dLDyHTbrQHz3b1q  
3Tb8auz97/J1p4pw39PT42CoR0DzPBrXJV+0gl NE/83C1yOSSJ8Bj C9LEwE=  
---- 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  
ragnarok# 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'.  
ragnarok# activate-config  
Activate-Config received, processing.  
waiting for request to finish  
SD is not QOS-capable  
Request to 'ACTIVATE-CONFIG' has finished,  
Activate Complete  
ragnarok#
```

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.

```
ragnarok# configure terminal  
ragnarok(config)# security  
ragnarok(security)# public-key  
ragnarok(public-key)#
```

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

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

creates a public key record named *tashtego*.

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

```
ragnarok(publ i c-key)# done
publ i c-key
  name          tashtego
  type          rsa
  si ze        1024
  last-modified-by    admn@consol e
  last-modified-date  2009-03-06 11:18:00
```

```
ragnarok(publ i c-key)#

```

4. Make a note of the **last-modified-date** time value.

5. Move back to admin mode, and save and activate the configuration.

```
ragnarok(publ i c-key)# exit
ragnarok(security)# exit
ragnarok(configure)# exit
ragnarok#
ragnarok# save-configuration
```

```
...
```

```
...
```

```
ragnarok# activate-configuration
```

```
...
```

```
...
```

```
...
```

```
ragnarok#
```

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.

```
ragnarok# ssh-pub-key generate tashtego
```

```
Please wait...
```

```
publ i c-key 'tashtego' (RFC 4716/SECSH format):
```

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

```
Comment: "1024-bit rsa"
```

```
AAAAB3NzaC1yc2EAAAABIwAAAEArZEP1/WiYsdGd/Pi8V6pnSwV4cVG4U+jV
OwiSwNJCC9Nk82/FKYIeLZevy9D3I rZ8ytvu+sCYy0fNk4nwvz20c2N+r86kD
ru88JkUqpeIJDx1AR718I cpr7ZaAx2L+e7cpyRSXCgbQR7rXu2H3bp9Jc0VhR
2fmkclmrGAIr7Gnc=
```

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

```
SSH publ i c-key pair generated successfully....
```

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

```
ragnarok#
```

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.

```
ragnarok# save-config
```

```
...
```

```
...
```

```
...
```

```
ragnarok# activate-config
```

```
...
```

```
...
```

```
...
```

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

```
ragnarok# configure terminal
```

```
ragnarok(configure)# security
```

```
ragnarok(security)# public-key
```

```
ragnarok(public-key)# sel
```

```
<name>:
```

```
1: acme01
```

```
2: acme02
```

```
3: tashtego
```

```
selection: 3
```

```
ragnarok(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

```
ragnarok(public-key)#
```

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

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 in 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.

```
ragnarok# show security ssh-pub-key brief
Logi n-name:
    acme74
fi nger-print:
    51: 2f: f1: dd: 79: 9e: 64: 85: 6f: 22: 3d: fe: 99: 1f: c8: 21
fi nger-print-raw:
    0a: ba: d8: ef: bb: b4: 41: d0: dd: 42: b0: 6f: 6b: 50: 97: 31

Logi n-name:
    fedal l ah
fi nger-print:
    c4: a0: eb: 79: 5b: 19: 01: f1: 9c: 50: b3: 6a: 6a: 7c: 63: d5
fi nger-print-raw:
    ac: 27: 58: 14: a9: 7e: 83: fd: 61: c0: 5c: c8: ef: 78: e0: 9c
ragnarok#
```

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

```
ragnarok# show security ssh-pub-key brief fedallah
  log-in-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
ragnarok#
```

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

```
ragnarok# 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:
```

```
AAAAB3NzaC1yc2EAAAABIwAAAQEA70Bf08jJe7MSMgerjDTgZpbPblrX4n17L0JgPC7cILcDGETKSiVt5MjcsAv3v6AEN2pYZi h0xd2Zzi smpoo019kkJ56s/IjGstEzqXMKHKUr9mBVqvqlE0TqbowEi5sz2AP31GUjQTCZRF1X00x8A44vHZCum93/jfNRsnWQ1mhHmaZMmT2LSh0r4J/NI p+vpsvpdrolV6Ftz5eiVfgocxrDrjNcVtsAMyLBpDdL6e9Xeb0zGSS92TPuKP/yqzLJ2G5NVFhxdw5i+FvdHz1vBdvB505y20PJ/i z1u3TA/307tyntB0b7beDylrg64Azc8G7E3AGiH49LnBt1Qf/aw==
```

```
  modulus: (256)
  ECE05FD3C8C97BB3123207AB8C34E06696CF6E5AD7E27D7B2D02603C2EDC94B70
  3184B4A4A256DE4C8DC49ABF7BFA004376A5866284EC5DD99CE2B26A68A34D7D9
  24279EACFC88C6B2D133A9730A1CA52BF66055AAFA8810E4EA6E8C048B9B33D80
  3F7D4652341308A6511755CE431F00E38BC7642BA6F77FE37CD46C9D64359A11E
  66993264F62D284EAF827F365A7EBE9B2FA5DAE8955E85B73E5E8957E0A1CC6B0
  EB8CD715B6C00CC8B0690DD2FA7BD5DE6D0CC6492F764CFB8A3FFCAACCB2761B9
  355161C5DC398BE16F747CF5BC176F079D39CB640F8FF8B3D6EDD303FDCEEEDCA
  7B4139BEDB783C88AE0EB803373C1BB137006887E3D2E706D9507FF6B
  exponent: (1)
  23
```

```
ragnarok#
```

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.

```
ragnarok# 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:
    AAAAB3NzaC1kc3MAAACBAPY8Z0HY2yFSJA6XYC9HRwNHxaehvx5w0J0rzZdzoS0X
    bETW6ToHv8D1UJ/z+zHo9Fi ko5XybZnDI aBDHtbl Q+Yp7Stxyl tHnXF1YLfKD1G4T
    6JYrdHYI 140m1eg9e4NnCRI eaqoZPF3UGfZi a6bXrGTQf3gJq2e7Yi sk/gF+1VAAA
    AFQDb8D5cvwHWTZDPfx0D2s9Rd7NBvQAAAI EAI N92+Bb7D4KLYk3I wRbXbl wXdkPg
    gA4pfdtW9vGfJ0/RHd+Nj B4eo1D+Odi x6tXwYGN7PKS5R/FXPNwxHPapcj 9uL1Jn2
    AWQ2dsknf+i /FAAvi oUPkmdMc0zuW0SOEsSnhVDTx3WdvVcGcBq9cetzrt0Kw0ocJ
    mJ80qadxTRHtUAAACBAN7CY+KKv1gHpRzFwdQm7HK9bb1LAo2KwaoXnadFgeptNBQ
    eSXG1v0+JsvphVMBJc9HSn24VYtYtsMu74qXvi Yj zi VucWKj j KEb11j uqnFOGDI B3
    VVmxHLmxnAz643WK42Z7dLM5sY29ouezv4Xz2PuMch5VGPP+CDqzCM4I oWgV

p: (128)
F63C64E1D8DB2152240E97602F47470347C5A7A1BF1E70389D2BCD9773A12397C
5B1135BA4E81EFF03D5427FCFECC7A3D162928E57C9B6670C86810C7B5B950F98
A7B4ADC7296D1E75C5D582DF283D46E13E8962B747608D783A6D5E83D7B836709
195E6AA193C5DD419F6626BA6D7AC64D07F7809AB67BB622B24FE017ED55

q: (20)
DBF03E5CBF01D64D90CF7D7D03Dacf5177B341BD

g: (128)
94DF76F816FB0F828B624DC8C116D76E5C177643E0800E297DDB56F6F19F274FD
11DDF8D8C1E1EA350FED1D8B1EAD5F060637B3CA4B947F1573CDC311CF6A9723F
6E2F5267D80590D9DB249DFFA2FC5000BE2A143E499D31CD33B96A12384B12361
543B57DD676F55C19C06AF5C7ADCEBB4E2963A8709989F34A9A7714D11ED5

pub_key: (128)
DEC263E28ABF5807A51CC5C1D426EC72BD6DBD4B028D8AC1AA179DA74581EA6D3
4141E4971B5BCEF89B2FA6154C04973D1D29F6E1562D62DB0CBBBE2A5EF8988F3
895B9C58A8E32846F5D63BAA9C5D060E50775559B11CB9B19C0CFAE3758AE3667
B74B339B18DBDA2E7B3BF85F3D8FB8C721E5518F3FE083AB308CE25A16815
```

ragnarok#

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

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

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

bye	Quit sftp.
cd path	Change remote directory to path.
lcd path	Change local directory to path.
chgrp grp path	Change group of file path to group. group must be a numeric GID.
chmod mode path	Change permissions of file path to mode.
chown own path	Change owner of file path to own. own must be a numeric UID.
dir (or ls)	List the files in the current directory
exit	Quit sftp.

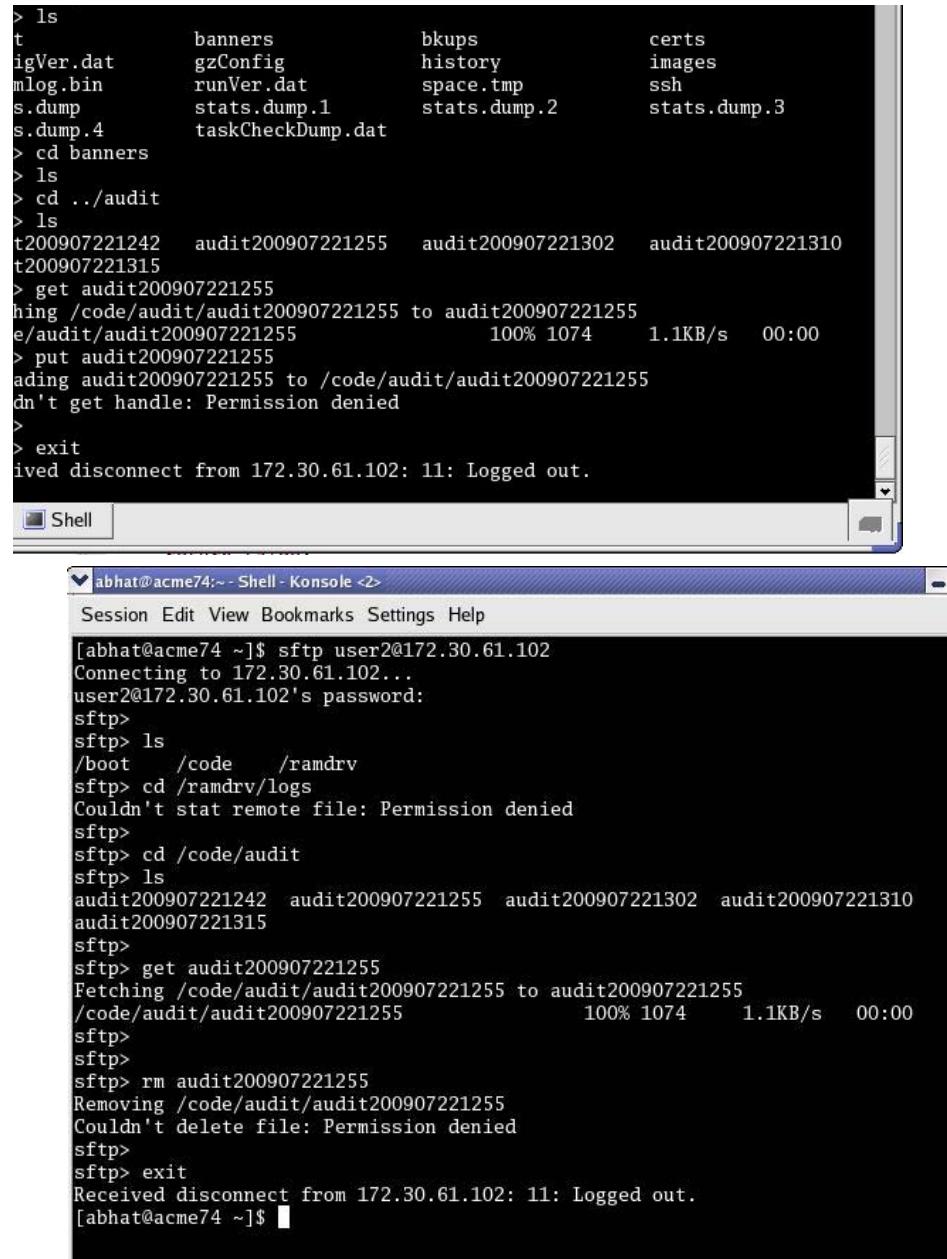
get [flags] remote-path [local-path]	Retrieve the remote-path and store it on the local machine. If the local path name is not specified, it is given the same name it has on the remote machine. If the -P flag is specified, then the file's full permission and access time are copied too.
help	Display help text.
lcd	Change the directory on the local computer
lls	See a list of the files in the current directolls [ls-options [path]] Display local directory listing of either path or current directory if path is not specified.
lmkdir path	Create local directory specified by path.
ln oldpath newpath	Create a symbolic link from oldpath to newpath.
lpwd	Print local working directory.
ls [path]	Display remote directory listing of either path or current directory if path is not specified.
lumask umask	Set local umask to umask.
mkdir path	Create remote directory specified by path.
put [flags] local-path [local-path]	Upload local-path and store it on the remote machine. If the remote path name is not specified, it is given the same name it has on the local machine. If the -P flag is specified, then the file's full permission and access time are copied too.
pwd	Display remote working directory.
quit	Quit sftp.
rename oldpath newpath	Rename remote file from oldpath to newpath.
rmdir path	Remove remote directory specified by path.
rm path	Delete remote file specified by path.
symlink oldpath newpath	Create a symbolic link from oldpath to newpath.
! command	Execute command in local shell.
!	Escape to local shell.
?	Synonym for help.

Note: Command availability is subject to Acme Packet 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.



```

> ls
t           banners      bkups      certs
igVer.dat   gzConfig    history    images
mlog.bin    runVer.dat  space.tmp  ssh
s.dump      stats.dump.1 stats.dump.2 stats.dump.3
s.dump.4    taskCheckDump.dat

> cd banners
> ls
> cd ../audit
> ls
t200907221242  audit200907221255  audit200907221302  audit200907221310
t200907221315

> get audit200907221255
hing /code/audit/audit200907221255 to audit200907221255
e/audit/audit200907221255          100% 1074    1.1KB/s  00:00
> put audit200907221255
ading audit200907221255 to /code/audit/audit200907221255
dn't get handle: Permission denied
>
> exit
ived disconnect from 172.30.61.102: 11: Logged out.

[abhat@acme74:~]# Shell
```



```

abhat@acme74:~ - Shell - Konsole <2>
Session Edit View Bookmarks Settings Help
[abhat@acme74 ~]$ sftp user2@172.30.61.102
Connecting to 172.30.61.102...
user2@172.30.61.102's password:
sftp>
sftp> ls
/boot   /code   /ramdrv
sftp> cd /ramdrv/logs
Couldn't stat remote file: Permission denied
sftp>
sftp> cd /code/audit
sftp> ls
audit200907221242  audit200907221255  audit200907221302  audit200907221310
audit200907221315
sftp>
sftp> get audit200907221255
Fetching /code/audit/audit200907221255 to audit200907221255
/code/audit/audit200907221255          100% 1074    1.1KB/s  00:00
sftp>
sftp>
sftp> rm audit200907221255
Removing /code/audit/audit200907221255
Couldn't delete file: Permission denied
sftp>
sftp> exit
Received disconnect from 172.30.61.102: 11: Logged out.
[abhat@acme74 ~]$
```

Figure 10: 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 */ramdrv*, or to delete an audit log.

The following table summarizes SFTP authentication and authorization.

Table 4: SFTP Authentication & Authorization

<u>User Name</u>	<u>Logins into/prompt</u>	<u>Authentication</u>	<u>Authorization</u>
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

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,aciConsole-admin@console,configuration,save-config,success,CfgVersion=111,..,
activate-config	Every activate-config CLI command 2009-03-05 15:45:36,aciConsole-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	<p>a) any action that creates a configuration property b) any action that creates a file</p> <p>2009-03-05 15:45:01,acliConsole-admin@console,configuration,create, success,public-key, Element=</p> <pre><?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' /></pre>
Modify	<p>a) any action that modifies a configuration property</p> <p>2009-03-05 15:48:01,acliConsole-admin@console,configuration,modify, success,public-key, Previous=</p> <pre><?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' /></pre> <p>New=</p> <pre><?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' /></pre>

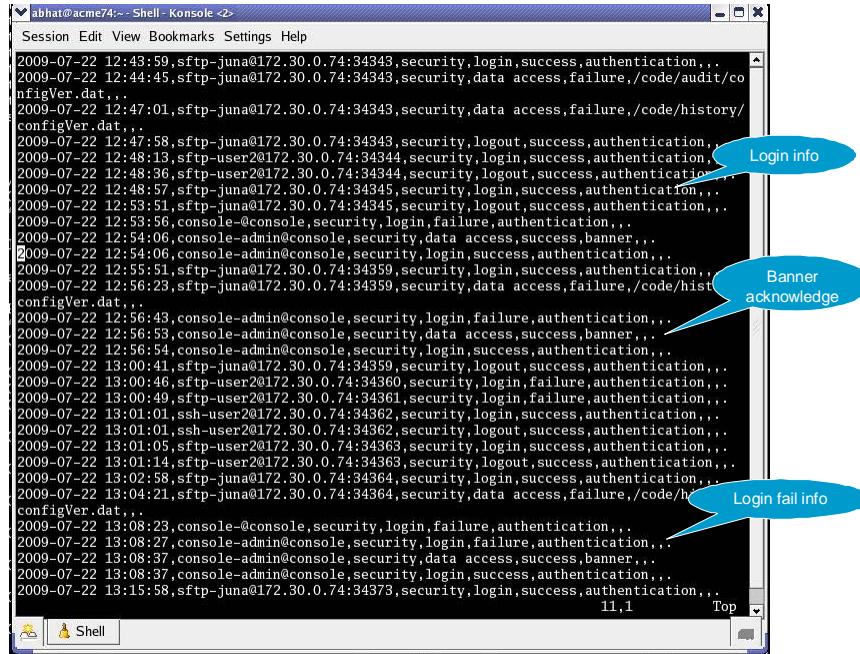
Delete	<p>a) any action that deletes a configuration property b) any action that deletes a file</p> <pre>2009-03-05 15:51:39,acliConsole-admin@console,configuration,delete, success,public-key, Element= <?xml version='1.0' standalone='yes'?> <sshPubKeyRecord name='dummy' comment="" keyType='2' encrType='2' keySize='1024' pubKey="" privKey="" fingerPrint="" fingerPrintRaw="" lastModifiedBy='acmin@console' lastModifiedDate='2009-03-05 15:51:39' </sshPubKeyRecord</pre>
--------	--

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.

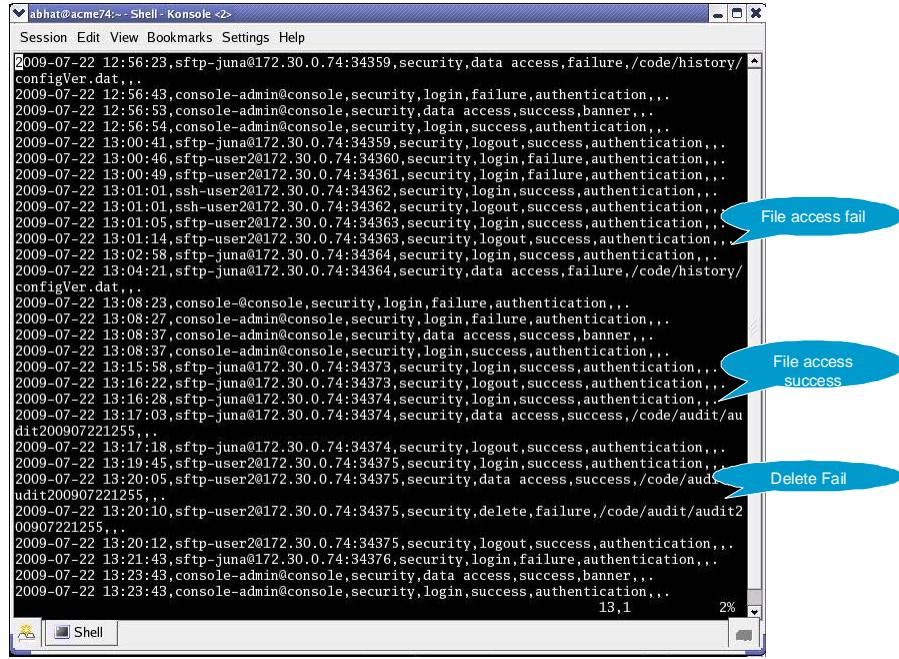
Audit Log Samples

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



```
abhat@acme74:~ - Shell - Konsole <2>
Session Edit View Bookmarks Settings Help
2009-07-22 12:43:59,ssh-juna@172.30.0.74:34343,security,login,success,authentication,..  
2009-07-22 12:44:45,ssh-juna@172.30.0.74:34343,security,data access,failure,/code/audit/co  
nfigVer.dat,..  
2009-07-22 12:47:01,ssh-juna@172.30.0.74:34343,security,data access,failure,/code/history/  
configVer.dat,..  
2009-07-22 12:48:13,ssh-juna@172.30.0.74:34344,security,login,success,authentication,..  
2009-07-22 12:48:36,ssh-juna@172.30.0.74:34344,security,logout,success,authentication,..  
2009-07-22 12:48:57,ssh-juna@172.30.0.74:34345,security,login,success,authentication,..  
2009-07-22 12:53:51,ssh-juna@172.30.0.74:34345,security,logout,success,authentication,..  
2009-07-22 12:53:56,console-@console,security,login,failure,authentication,..  
2009-07-22 12:54:06,console-admin@console,security,data access,success,banner,..  
2009-07-22 12:54:06,console-admin@console,security,login,success,authentication,..  
2009-07-22 12:55:51,ssh-juna@172.30.0.74:34359,security,login,success,authentication,..  
2009-07-22 12:56:23,ssh-juna@172.30.0.74:34359,security,data access,failure,/code/his  
tory/configVer.dat,..  
2009-07-22 12:56:43,console-admin@console,security,login,failure,authentication,..  
2009-07-22 12:56:53,console-admin@console,security,data access,success,banner,..  
2009-07-22 12:56:54,console-admin@console,security,login,success,authentication,..  
2009-07-22 13:00:41,ssh-juna@172.30.0.74:34359,security,logout,success,authentication,..  
2009-07-22 13:00:46,ssh-juna@172.30.0.74:34360,security,login,failure,authentication,..  
2009-07-22 13:00:49,ssh-juna@172.30.0.74:34361,security,login,failure,authentication,..  
2009-07-22 13:01:01,ssh-juna@172.30.0.74:34362,security,logout,success,authentication,..  
2009-07-22 13:01:01,ssh-user@172.30.0.74:34362,security,login,failure,authentication,..  
2009-07-22 13:01:05,ssh-user@172.30.0.74:34363,security,login,success,authentication,..  
2009-07-22 13:01:14,ssh-user@172.30.0.74:34363,security,logout,success,authentication,..  
2009-07-22 13:02:58,ssh-juna@172.30.0.74:34364,security,data access,failure,/code/his  
tory/configVer.dat,..  
2009-07-22 13:08:23,console-@console,security,login,failure,authentication,..  
2009-07-22 13:08:27,console-admin@console,security,login,failure,authentication,..  
2009-07-22 13:08:37,console-admin@console,security,data access,success,banner,..  
2009-07-22 13:08:37,console-admin@console,security,login,success,authentication,..  
2009-07-22 13:15:58,ssh-juna@172.30.0.74:34373,security,login,success,authentication,..
```

Figure 11: Login Reporting

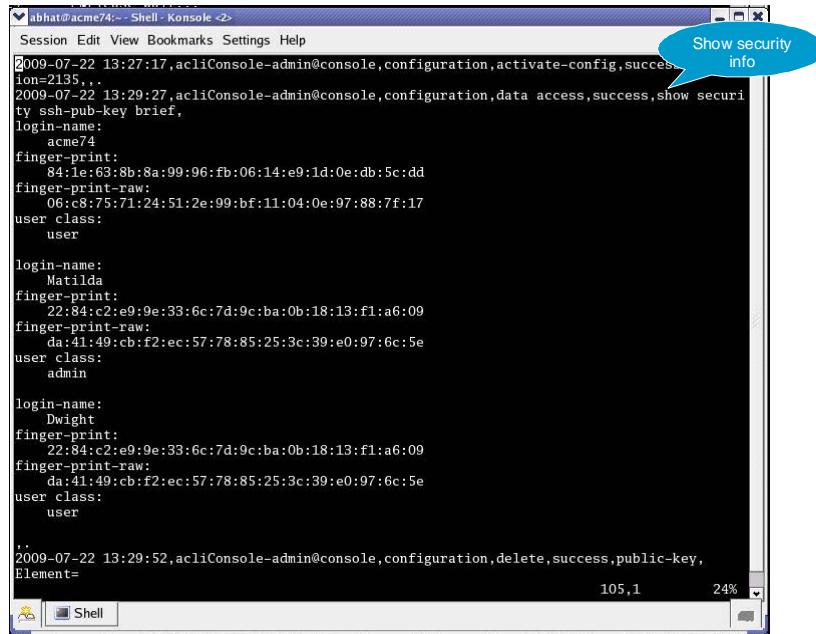


```

abhat@acme74:~/Shell-Konsole <2>
Session Edit View Bookmarks Settings Help
2009-07-22 12:56:23,sftp-juna@172.30.0.74:34359,security,data access,failure,/code/history/configVer.dat,..
2009-07-22 12:56:43,console-admin@console,security,login,failure,authentication,..
2009-07-22 12:56:53,console-admin@console,security,data access,success,banner,..
2009-07-22 12:56:54,console-admin@console,security,login,success,authentication,..
2009-07-22 13:00:41,sftp-juna@172.30.0.74:34359,security,logout,success,authentication,..
2009-07-22 13:00:46,sftp-user@172.30.0.74:34360,security,login,failure,authentication,..
2009-07-22 13:00:49,sftp-user@172.30.0.74:34361,security,login,failure,authentication,..
2009-07-22 13:01:01,ssh-user@172.30.0.74:34362,security,login,success,authentication,..
2009-07-22 13:01:01,ssh-user@172.30.0.74:34362,security,logout,success,authentication,..
2009-07-22 13:01:05,sftp-user@172.30.0.74:34363,security,login,success,authentication,..
2009-07-22 13:01:14,sftp-user@172.30.0.74:34363,security,logout,success,authentication,..
2009-07-22 13:02:58,sftp-juna@172.30.0.74:34364,security,login,success,authentication,..
2009-07-22 13:04:21,sftp-juna@172.30.0.74:34364,security,data access,failure,/code/history/configVer.dat,..
2009-07-22 13:08:23,console-admin@console,security,login,failure,authentication,..
2009-07-22 13:08:27,console-admin@console,security,login,failure,authentication,..
2009-07-22 13:08:37,console-admin@console,security,data access,success,banner,..
2009-07-22 13:08:37,console-admin@console,security,login,success,authentication,..
2009-07-22 13:15:58,sftp-juna@172.30.0.74:34373,security,login,success,authentication,..
2009-07-22 13:16:22,sftp-juna@172.30.0.74:34373,security,logout,success,authentication,..
2009-07-22 13:17:03,sftp-juna@172.30.0.74:34374,security,login,success,authentication,..
2009-07-22 13:17:18,sftp-juna@172.30.0.74:34374,security,logout,success,authentication,..
2009-07-22 13:19:45,sftp-user@172.30.0.74:34375,security,login,success,authentication,..
2009-07-22 13:20:05,sftp-user@172.30.0.74:34375,security,data access,success,/code/audit/audit00907221255,..
2009-07-22 13:20:10,sftp-user@172.30.0.74:34375,security,delete,failure,/code/audit/audit00907221255,..
2009-07-22 13:20:12,sftp-user@172.30.0.74:34375,security,logout,success,authentication,..
2009-07-22 13:21:43,sftp-juna@172.30.0.74:34376,security,login,failure,authentication,..
2009-07-22 13:23:45,console-admin@console,security,data access,success,banner,..
2009-07-22 13:23:45,console-admin@console,security,login,success,authentication,..

```

Figure 12: File Access Reporting



```

abhat@acme74:~/Shell-Konsole <2>
Session Edit View Bookmarks Settings Help
2009-07-22 13:27:17,acliConsole-admin@console,configuration,activate-config,success,Element=2135,..
2009-07-22 13:29:27,acliConsole-admin@console,configuration,data access,success,show security ssh-pub-key brief,
login-name:
    acme74
    finger-print:
        84:1e:63:8b:8a:99:96:fb:06:14:e9:1d:0e:db:5c:dd
    finger-print-raw:
        06:c8:75:71:24:51:2e:99:bf:11:04:0e:97:88:7f:17
    user class:
        user

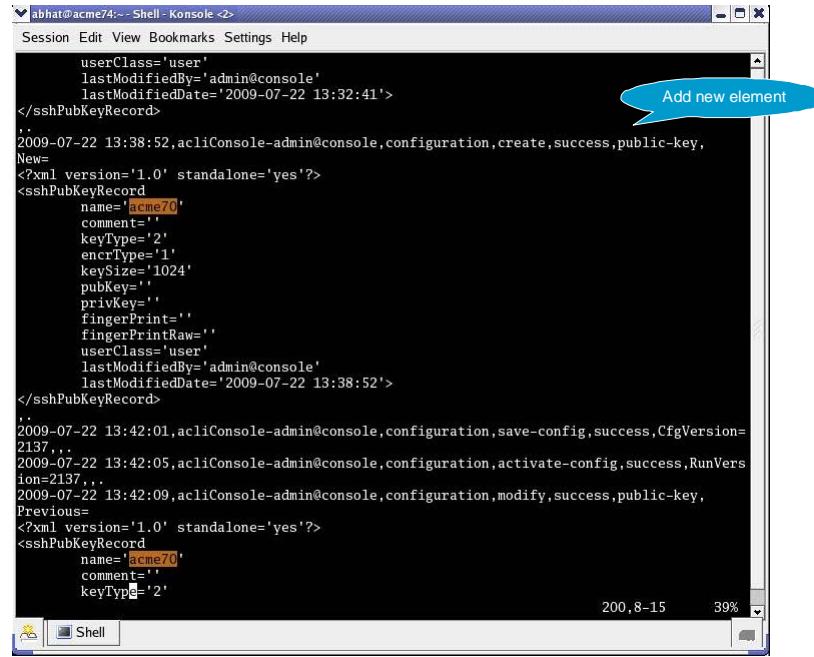
    login-name:
        Matilda
    finger-print:
        22:84:c2:e9:9e:33:6c:7d:9c:ba:0b:18:13:f1:a6:09
    finger-print-raw:
        da:41:49:cb:f2:ec:57:78:85:25:3c:39:e0:97:6c:5e
    user class:
        admin

    login-name:
        Dwight
    finger-print:
        22:84:c2:e9:9e:33:6c:7d:9c:ba:0b:18:13:f1:a6:09
    finger-print-raw:
        da:41:49:cb:f2:ec:57:78:85:25:3c:39:e0:97:6c:5e
    user class:
        user

    .
2009-07-22 13:29:52,acliConsole-admin@console,configuration,delete,success,public-key,Element=

```

Figure 13: show security Reporting

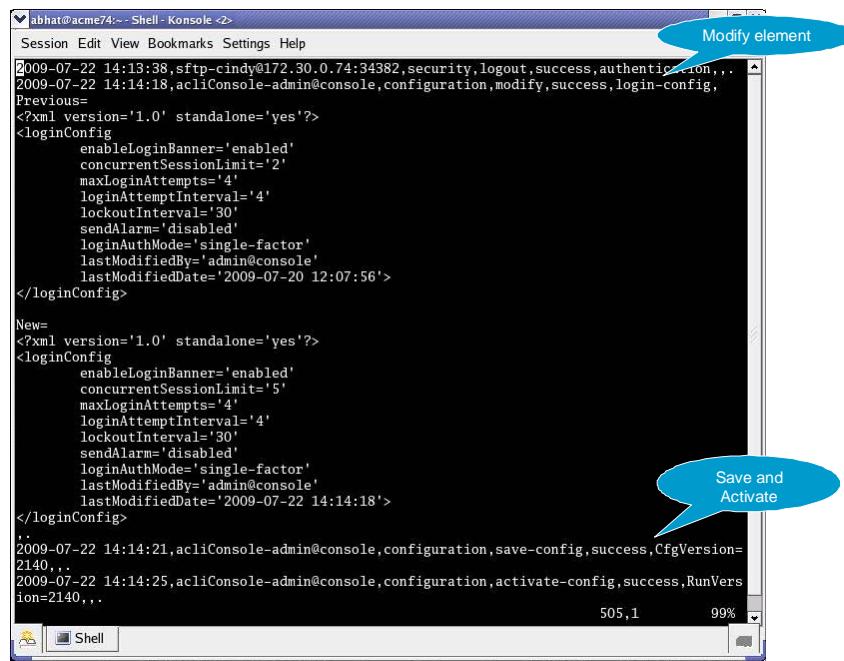


```

abhat@acme74:~ Shell - Konsole <2>
Session Edit View Bookmarks Settings Help
    userClass='user'
    lastModifiedBy='admin@console'
    lastModifiedDate='2009-07-22 13:32:41'
</sshPubKeyRecord>
...
2009-07-22 13:38:52,acliConsole-admin@console,configuration,create,success,public-key,
New=
<?xml version='1.0' standalone='yes'?>
<sshPubKeyRecord
    name='acme70'
    comment=''
    keyType='2'
    encrType='1'
    keySize='1024'
    pubKey=''
    privKey=''
    fingerPrint=''
    fingerPrintRaw=''
    userClass='user'
    lastModifiedBy='admin@console'
    lastModifiedDate='2009-07-22 13:38:52'
</sshPubKeyRecord>
...
2009-07-22 13:42:01,acliConsole-admin@console,configuration,save-config,success,CfgVersion=2137...
2009-07-22 13:42:05,acliConsole-admin@console,configuration,activate-config,success,RunVersion=2137...
2009-07-22 13:42:09,acliConsole-admin@console,configuration,modify,success,public-key,
Previous=
<?xml version='1.0' standalone='yes'?>
<sshPubKeyRecord
    name='acme70'
    comment=''
    keyType='2'
...
2009-07-15 08:15 39%

```

Figure 14: Create Element Reporting



```

abhat@acme74:~ Shell - Konsole <2>
Session Edit View Bookmarks Settings Help
2009-07-22 14:13:38,sftp-cindy@172.30.0.74:34382,security,logout,success,authentication, ...
2009-07-22 14:14:18,acliConsole-admin@console,configuration,modify,success,login-config,
Previous=
<?xml version='1.0' standalone='yes'?>
<loginConfig
    enableLoginBanner='enabled'
    concurrentSessionLimit='2'
    maxLoginAttempts='4'
    loginAttemptInterval='4'
    lockoutInterval='30'
    sendAlarm='disabled'
    loginAuthMode='single-factor'
    lastModifiedBy='admin@console'
    lastModifiedDate='2009-07-20 12:07:56'
</loginConfig>
...
New=
<?xml version='1.0' standalone='yes'?>
<loginConfig
    enableLoginBanner='enabled'
    concurrentSessionLimit='5'
    maxLoginAttempts='4'
    loginAttemptInterval='4'
    lockoutInterval='30'
    sendAlarm='disabled'
    loginAuthMode='single-factor'
    lastModifiedBy='admin@console'
    lastModifiedDate='2009-07-22 14:14:18'
</loginConfig>
...
2009-07-22 14:14:21,acliConsole-admin@console,configuration,save-config,success,CfgVersion=2140...
2009-07-22 14:14:25,acliConsole-admin@console,configuration,activate-config,success,RunVersion=2140...
505,1 99%

```

Figure 15: 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:

```
ragnarok# configure terminal > security > 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.

```
ragnarok(audit-logging)# admin-state enable  
ragnarok(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.

```
ragnarok(audit-logging)# detail-level verbose  
ragnarok(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.

```
ragnarok(audit-logging)# file-transfer-time 1  
ragnarok(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.

```
ragnarok(audi t-l oggi ng)# max-storage-space 8
ragnarok(audi t-l oggi ng)#

```
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-space** \times (**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**.

```
ragnarok(audi t-l oggi ng)# percentage-full 0
ragnarok(audi t-l oggi ng)#

```
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**.

```
ragnarok(audi t-l oggi ng)# max-file-size 0
ragnarok(audi t-l oggi ng)#

```
8. **storage-path**—specifies the directory that houses the audit log
Retain the default value (/code/audi t), or identify another local directory.

```
ragnarok(audi t-l oggi ng)# storage-path code/mgmt
ragnarok(audi t-l oggi ng)#

```

A sample audit log configuration appears below:

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

```
ragnarok(audit-logging)# push-receiver
ragnarok(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,

```
ragnarok(push-receiver)# server 192.0.2.100
ragnarok(push-receiver)#[/pre]
```
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.

```
ragnarok(push-receiver)# port 2222
ragnarok(push-receiver)#[/pre]
```
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,

```
ragnarok(push-receiver)# remote-path /home/acme/auditLogs
ragnarok(push-receiver)#[/pre]
```
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,

```
ragnarok(push-receiver)# filename-prefix avuik
ragnarok(push-receiver)#[/pre]
```
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,

```
ragnarok(push-receiver)# auth-type publickey
ragnarok(push-receiver)#[/pre]
```
7. **username**—specifies the username used to authenticate to this SFTP server
Provide the username used to authenticate/login to this server. For example,

```
ragnarok(push-receiver)# username acme1
ragnarok(push-receiver)#[/pre]
```

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,

```
ragnarok(push-receiver)# password =yetAnotherPW!
```
9. **public-key**—required when **auth-type** is **publ ickey**, 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,

```
ragnarok(push-receiver)# publ ickey certSFTP-1
```

A sample SFTP server configuration appears below:

```
ragnarok(push-receiver)# 192.0.2.100
ragnarok(push-receiver)# remote-path /home/acme
ragnarok(push-receiver)# fil ename-prefi x auvi k
ragnarok(push-receiver)# username acme
ragnarok(push-receiver)# auth-type publ ickey
ragnarok(push-receiver)# publ ickey acme01
ragnarok(push-receiver)# 192.0.2.125
ragnarok(push-receiver)# remote-path /securi ty/audi tLogs
ragnarok(push-receiver)# fil ename-prefi x auvi k
ragnarok(push-receiver)# username acme
ragnarok(push-receiver)# auth-type password
ragnarok(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 Acme Packet Net-Net 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 space 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.

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
- FTP 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.

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.

```
ragnarok(ike-confi g)# udp-port 5000  
ragnarok(ike-confi g)#

```

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.

```
ragnarok(ike-confi g)# sd-authentication-method certi fi cate  
ragnarok(ike-confi g)#

```

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.

```
ragnarok(ike-confi g)# shared-password !yetAnotherPaSSword1of87354  
ragnarok(ike-confi g)#

```

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.

```
ragnarok(ike-confi g)# certi fi cate-prof le-id val Cred-IKEv2  
ragnarok(ike-confi g)#

```

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.

```
ragnarok(ike-confi g)# dpd-time-interval 20  
ragnarok(ike-confi g)#

```

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.

```
ragnarok(i ke-confi g)# v2-ike-life-seconds 43200  
ragnarok(i ke-confi g)#

```

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.

```
ragnarok(i ke-confi g)# v2-ipsec-life-seconds 14400  
ragnarok(i ke-confi g)#

```

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.

```
ragnarok(i ke-confi g)# eap-protocol eap-radius-passthru  
ragnarok(i ke-confi g)#

```

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.

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

```
ragnarok(i ke-confi g)# eap-bypass-identity enabled  
ragnarok(i ke-confi g)#

```

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.

```
ragnarok(ike-confi g)# addr-assignment radius-only  
ragnarok(ike-confi g)#{/pre}
```

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.

```
ragnarok(ike-confi g)# overload-threshold 60  
ragnarok(ike-confi g)#{/pre}
```

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.

```
ragnarok(ike-confi g)# overload-interval 3  
ragnarok(ike-confi g)#{/pre}
```

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 Net-Net 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.

```
ragnarok(ike-confi g)# overload-action drop-new-connection  
ragnarok(ike-confi g)#{/pre}
```

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 Net-Net 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**.

```
ragnarok(ike-confi g)# overload-critical-threshold 75
ragnarok(ike-confi g)#

```

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.

```
ragnarok(ike-confi g)# overload-critical-interval 2
ragnarok(ike-confi g)#

```

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

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

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

- The active Net-Net 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 Net-Net SBC in the node.
- The standby Net-Net SBC is the backup system, which maintains a synchronous configuration with the active node. The standby Net-Net 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 *Net-Net 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.

```
ragnarok(ike-confi g)# red-port 1995
ragnarok(ike-confi g)#

```

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.

```
ragnarok(ike-confi g)# red-trans 7500
ragnarok(ike-confi g)#

```

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.

```
ragnarok(i ke-confi g)# red-sync-start-time 2500  
ragnarok(i ke-confi g)#{
```

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.

```
ragnarok(i ke-confi g)# red-sync-comp-time 750  
ragnarok(i ke-confi g)#{
```

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.

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

```
ragnarok# configure terminal  
ragnarok(configure)# security  
ragnarok(security)# ike  
ragnarok(ike)# dpd-params  
ragnarok(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.

```
ragnarok(dpd-params)# name dpdTemplate-1  
ragnarok(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.

```
ragnarok(dpd-params)# max-loop 80  
ragnarok(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.

```
ragnarok(dpd-params)# max-endpoints 20  
ragnarok(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.

```
ragnarok(dpd-params)# max-cpu-limit 50  
ragnarok(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**.

```
ragnarok(dpd-params)# load-max-loop 30  
ragnarok(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**.


```
ragnarok(dpd-params) # load-max-endpoints 3
ragnarok(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.


```
ragnarok# configure terminal
ragnarok(configure) # security
ragnarok(security) # ike
ragnarok(ike) # ike-certificate-profile
ragnarok(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.


```
ragnarok(ike-certificate-profile) # identity joho.net
ragnarok(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.


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


```
ragnarok(ike-certificate-profile) # trusted-ca-certificates
verisign3-a, verisign3-b, batimore, thawte-a
ragnarok(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).


```
ragnarok(ike-certificate-profile)# verify-depth 10
ragnarok(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 Net-Net 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 Net-Net 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 Net-Net 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 be 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 Acme Packet Net-Net 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.

```
ragnarok# configure terminal  
ragnarok(configure)# security  
ragnarok(security)# ike  
ragnarok(ike)# data-flow  
ragnarok(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.

```
ragnarok(data-flow)# name omar  
ragnarok(data-flow)#{
```

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

```
ragnarok(data-flow)# realm-id access-1  
ragnarok(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 Net-Net 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.

```
ragnarok(data-flow)# group-size 32  
ragnarok(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.

```
ragnarok(data-flow)# upstream-rate 560000000  
ragnarok(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.

```
ragnarok(data-flow)# downstream-rate 2800000000
ragnarok(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 Acme Packet Net-Net 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.

```
ragnarok# configure terminal
ragnarok(config)# security
ragnarok(security)# Ike
ragnarok(Ike)# local-address-pool
ragnarok(Iocal-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.

```
ragnarok(Iocal-address-pool)# name phelps
ragnarok(Iocal-address-pool)#

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

```
ragnarok(Iocal-address-pool)# dns-realm-id access-1
ragnarok(Iocal-address-pool)#

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

```
ragnarok(Iocal-address-pool)# data-flow dFlow-1
ragnarok(Iocal-address-pool)#

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

```
ragnarok(Iocal-address-pool)# address-range
ragnarok(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.


```
ragnarok(address-range)# network-address 192.168.0.0
ragnarok(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


```
ragnarok# configure terminal
ragnarok(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.


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

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


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

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


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

```

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

```
ragnarok(ike-interface)# ike-mode initiator  
ragnarok(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.

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

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

```
ragnarok(ike-interface)# certificate-profile-id joho.net  
ragnarok(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.

```
ragnarok(ike-interface)# dpd-params-name olivier  
ragnarok(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 99999999 (seconds) with a default of 86400 (24 hours).

```
ragnarok(ike-interface)# v2-ike-life-seconds 21600  
ragnarok(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).

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

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

```
ragnarok(ike-interface)# addr-assignment local  
ragnarok(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.

```
ragnarok# config terminal
ragnarok(config)# security
ragnarok(security)# key
ragnarok(key)# tunnel-configuration-params
ragnarok(tunnel-configuration-params)#
```

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

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

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

```
ragnarok(tunnel -ori g-params)# remote-addr 192.168.34.90  
ragnarok(tunnel -ori g-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.

```
ragnarok(tunnel -orig-p-params) # retry-limit 5  
ragnarok(tunnel -orig-p-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.

```
ragnarok(tunnel -orig-p-params) # retry-time 24  
ragnarok(tunnel -orig-p-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.

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.

```
ragnarok# config terminal
ragnarok(config)# security
ragnarok(security)# ike
ragnarok(ike)# ike-interface
ragnarok(ike-interface)# select
<address>
172.30.1.150
172.30.1.151
172.30.55.127
```

```
selection: 3
ragnarok(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'

```
ragnarok(ike-interface)# tunnel-orig-name-list "syslog FTPserver SNMP-1 SNMP-2 auditLog keyStore"
ragnarok(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 Net-Net 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 Net-Net SBC issues an **apSecurityTunnelFailureNotification** with a supported value of *initiator-timeout* assigned to the **apSecurityFailureCause** field.

After issuing the alarm the Net-Net 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:

```
ragnarok# 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:

```
ragnarok# 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.

```
ragnarok# configure terminal  
ragnarok(configure)# security  
ragnarok(security)# idle  
ragnarok(idle)# idle-sai info  
ragnarok(idle-sai info)#

```

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.

```
ragnarok(i ke-sai nfo) # name SA-1  
ragnarok(i ke-sai nfo) #
```

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.

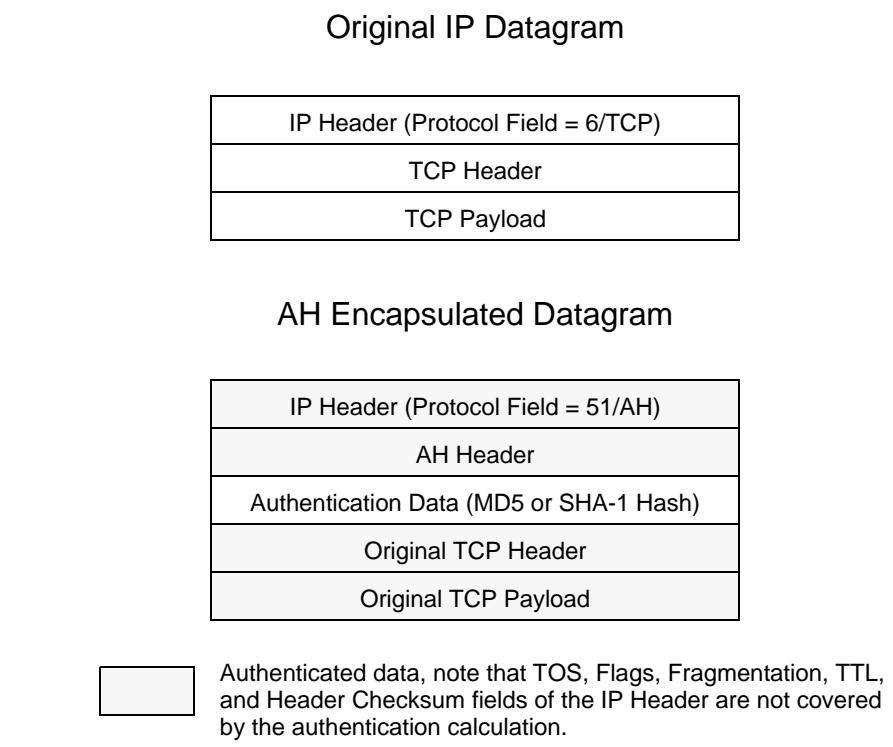
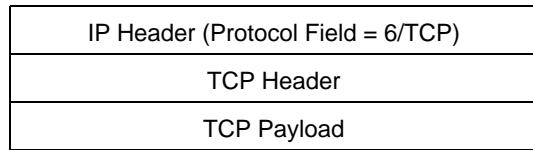
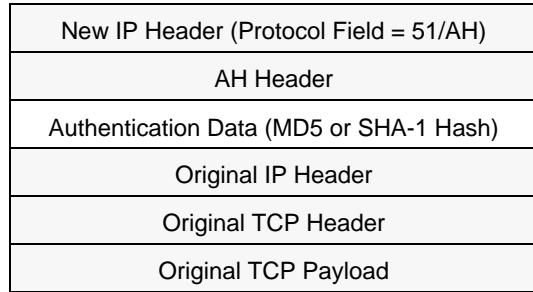


Figure 16: AH Transport Mode

Original IP Datagram



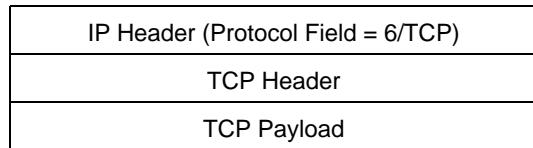
AH Encapsulated Datagram



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

Figure 17: AH Tunnel Mode

Original IP Datagram



ESP Encapsulated Datagram

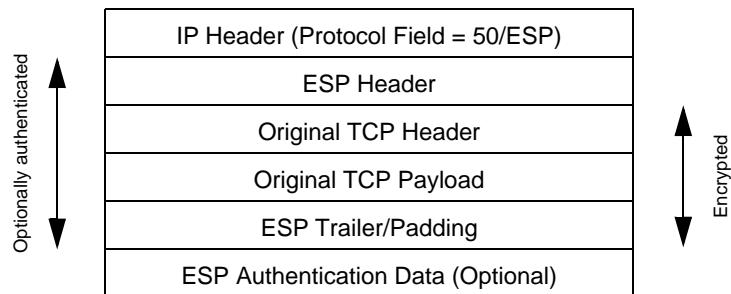
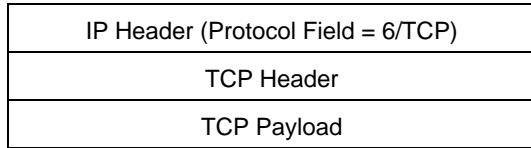


Figure 18: ESP Transport Mode

Original IP Datagram



ESP Encapsulated Datagram

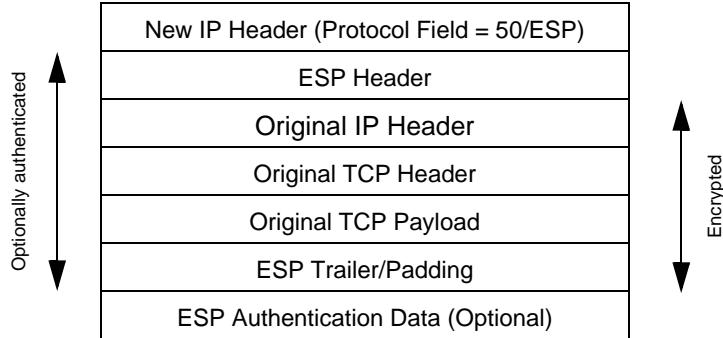


Figure 19: ESP Tunnel Mode

```
ragnarok(i ke-sai nfo)# security-protocol esp
ragnarok(i ke-sai nfo)#

```

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.

```
ragnarok(i ke-sai nfo)# auth-algo md5
ragnarok(i ke-sai nfo)#

```

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.

```
ragnarok(i ke-sai nfo)# encryption-algo aes
ragnarok(i ke-sai nfo)#

```

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.

```
ragnarok(i ke-sai nfo)# ipsec-mode tunnel  
ragnarok(i ke-sai nfo)#{/pre>
```
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*.

```
ragnarok(i ke-sai nfo)# tunnel -local-addr 192.169.204.14  
ragnarok(i ke-sai nfo)#{/pre>
```
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*.

```
ragnarok(i ke-sai nfo)# tunnel -remote-addr *  
ragnarok(i ke-sai nfo)#{/pre>
```
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.

```
ragnarok# configure terminal
ragnarok(configure)# security
ragnarok(security)# ipsec
ragnarok(ipsec)# security-policy
ragnarok(security-policy)#
```
2. Use the **ike-sainfo-name** parameter to assign an IKEv2 SA to this Security Policy.

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

```
ragnarok# show running-configuration security-policy
security-policy
    name ikepol
    network-interface wancom0
    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-sai info-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 0xFFFF
    last-modified-by admin@console
    last-modified-date 2009-11-11 19:06:32

security-policy
    name pol1
    network-interface wancom0
    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-sai
    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 0xFFFF
    last-modified-by admin@console
    last-modified-date 2009-11-11 19:07:03
```


This chapter describes implications of installing and deleting the Admin Security License on an Net-Net SBC.

Installation/Deletion Implications

The Admin Security is available for the Net-Net 3000 and 4000 SBC platform. This license enables the various security enhancements described in this guide. In the absence of an Admin Security 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.

Installation of the Admin Security license, disables access to the underlying operating system. As it is sometimes necessary to access the operating system to debug issues at customer sites, an Admin Sec-Shell license, that provides *temporary* operating system access, is available.

These two licenses relate as follows:

1. A Net-Net SBC with an Admin Security license also requires the Admin Sec-Shell license for operating system access.
2. A Net-Net SBC that has never had an Admin Security license install will have shell access enabled (as in prior software versions).
3. Removal of the Admin Security license does not re-enable operating system access (such access requires the Admin Sec-Shell license to be present). This ensures that a system cannot be compromised via the operating system by simple removing the Admin Security license.

A bit is permanently set in the NVRAM of a Net-Net SBC to denote that it currently has, or has previously had an Admin Security. This bit will be checked even if the license is removed, to determine if the Net-Net SBC should enforce the added security features.

Should the Admin Security license be removed the following restrictions are imposed:

- telnet access is not available
- FTP access is not available
- EMS (Element Management System) access is not available
- audit log deletion is not allowed
- ACP (Acme Control Protocol) is disabled
- operating system access is not allowed

