Oracle® Communications Diameter Signaling Router

DSR Security Guide

Release 9.0.0.0.0

F79491-02

February 2024

Oracle Communications Diameter Signaling Router DSR Security Guide, Release 9.0.0.0.0

F79491-02

Contents

1 Introduction

2 Overview

3 Implement Oracle Communications Diameter Signaling Router Security

My Oracle Support

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

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

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

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

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

Emergency Response

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

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

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

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

Locate Product Documentation on the Oracle Help Center Site

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

1. Access the Oracle Help Center site at http://docs.oracle.com.
2. Click `Industries`.
3. Under the Oracle Communications subheading, click the `Oracle Communications documentation` link.
   The Communications Documentation page appears. Most products covered by these documentation sets display under the headings `Network Session Delivery and Control Infrastructure` or `Platforms`.
4. Click on your Product and then the release Number.
   A list of the entire documentation set for the selected product and release appears.

To download a file to your location, right-click the `PDF` link, select `Save target as` (or similar command based on your browser), and save to a local folder.

Acronyms

The following table provides information about the acronyms and the terminology used in the document.

Table Acronyms

Acronym	Description
ACL	Access Control List
CLI	Command Line Interface
CSR	Customer Service Request
DSR	Diameter Signaling Router
ESP	Encapsulating Security Payload
GUI	Graphical User Interface
HA	High Availability
HIDS	Host Intrusion Detection System
IKE	Internet Key Exchange
IPsec	Internet Protocol security
IV	Initialization Vector
KPI	Key Performance Indicator
LDAP	Lightweight Directory Access Protocol
MMI	Machine to Machine Interface
MP	Message Processor
NOAMP	Network Operation, Administration, Maintenance, and Provisioning
OAM	Operation, Administrations, and Maintenance
OCH	Oracle Communications Help Center
OS	Operating System
REST	Representational State Transfer. A type of Northbound provisioning interface.
SFTP	Secure File Transfer Protocol
SOAM	System Operation, Administration, and Maintenance
SOAP	Simple Object Access Protocol
SNMP	Simple Network Management Protocol
SSO	Single Sign On
TLS	Transport Layer Security

What’s New in This Guide

This section lists the documentation updates for Release 9.0.0.0.0 in *Oracle Communications Diameter Signaling Router Security Guide*.

Release 9.0.0.0.0 - F79491-02, February 2024

Updated supported TLS versions to TLS 1.2 or TLS 1.3 in the Accessing Diameter Signaling Router System section.

Release 9.0.0.0.0 - F79491-01, April 2023

- Updated the procedure in the Disabling SSH Weak Key Exchange Algorithms section.
- Updated the following procedures in the Changing Internal Web Service Certificates and Key Material section:
  - *Creating and Distributing a separate Certificate and Key PEM File*
  - *Installing a separate PEM and CERT File on Each Distinct <hostname>*
- Added the Disabling SSH Weak Host Key Algorithms, MACs, and Ciphers section.
- Removed **iLO/ILOM Web GUI access** from Accessing Diameter Signaling Router System.

1 Introduction

This document provides information about configuring the Oracle Communications Diameter Signaling Router (DSR) to enhance the security posture of the system. The recommendations are optional. You can consider these options along with your organization's approved security strategies.

> **Note:**
>
> Additional configuration changes not included in this document are not recommended and can hinder the product's operation or Oracle's capability to provide appropriate support.

1.1 Audience

This guide is for administrators responsible for product and network security.

1.2 References

The following references capture the source material used to create this document. These documents are available in the Oracle Communications Diameter Signaling Router documentation set. For more information, see My Oracle Support.

- *Operation, Administration, and Maintenance (OAM) Guide*
- *Alarms, KPIs, and Measurements Reference*
- *DSR Upgrade Procedure*
- *PMAC Configuration Guide*
- *DSR VNFM Installation and User Guide*

2
Overview

This section explains the security objectives and provides a generic overview of the DSR deployment model.

DSR Security Objectives

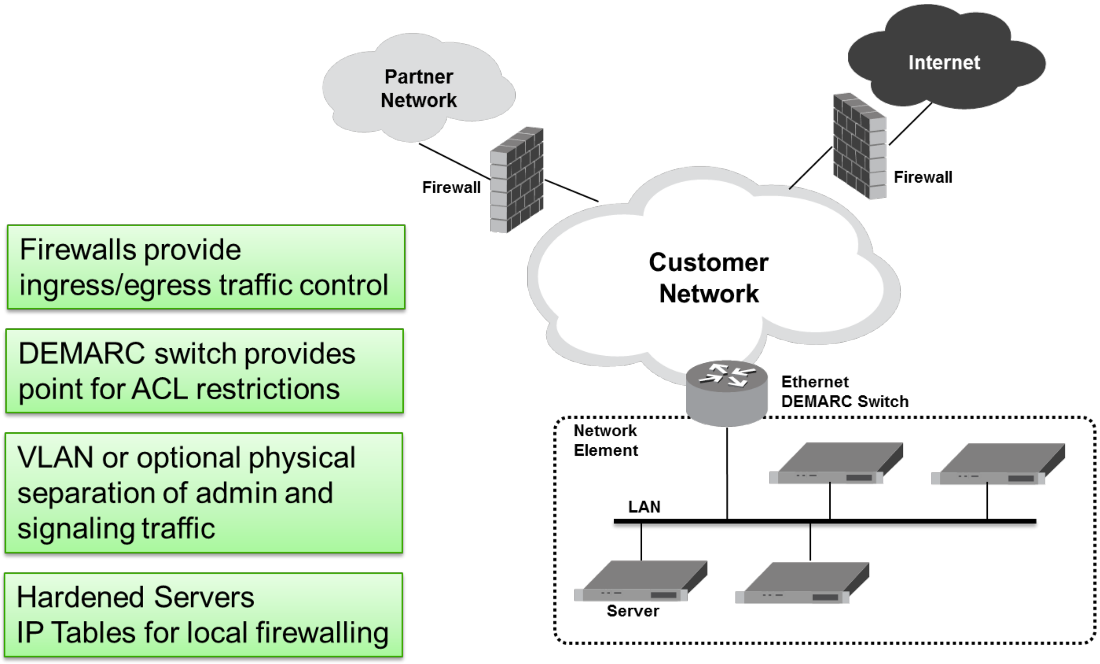
Oracle Communications Diameter Signaling Router (DSR) is developed with security in mind and delivered with a standard configuration that includes the best practices of Linux operating system security hardening. These practices include the following security objectives:

- Attack Surface Reduction
- Attack Surface Hardening
- Vulnerability Mitigation

Generic DSR Deployment Model

DSR is deployed in carrier's and service provider's core networks and provides critical signaling routing functionality for 4G, LTE, and IMS networks. The DSR solution is based on Linux servers and is highly scalable to accommodate a wide range of capacities to address networks of various sizes. The following image shows the generic deployment strategy model for DSR.

Figure 2-1 Generic DSR Deployment Strategy Model



Firewalls provide ingress/egress traffic control

DEMARC switch provides point for ACL restrictions

VLAN or optional physical separation of admin and signaling traffic

Hardened Servers
IP Tables for local firewalling

A DSR node consists of a suite of servers and related Ethernet switches that create a cluster of servers, operating as a single Network Element. It is assumed that the service provider establishes firewalls to isolate the core network elements from the internet and partner networks, as shown in Figure 2-1. In addition to these firewalls, DSR provides additional security capabilities such as:

- Access Control Lists (ACL) functionality at the demarcation switch.
- VLAN or optional physical separation of administrative and signaling traffic.
- IP Tables functionality at the servers for local firewalling.

2.1 Basic Security Considerations

The following principles are fundamental to using any application securely:

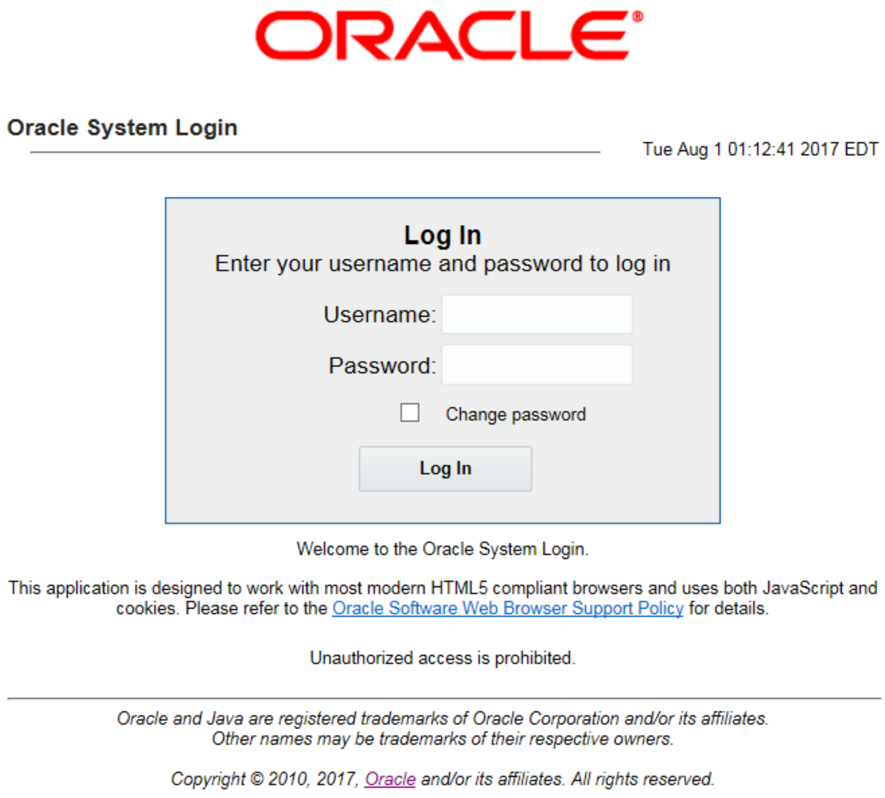
- Keep software up to date. Consider upgrading to the latest maintenance release. Consult with Oracle support team to plan for Oracle Communications Diameter Signaling Router software upgrades.
- Limit privileges. Users must be assigned to the proper user group and reviewed periodically to determine relevance to current work requirements. For more information, see the User Administration section.
- Monitor system activity. Establish who must access which system components, how often, and monitor those components. For more information, see Host Intrusion Detection System (HIDS) and Security Logs and Alarms sections.
- Configure software securely. For example, use secure protocols such as TLS and strong passwords. For more information, see the Passwords and Diameter Signaling Router OS Standard Features sections.
- Change default passwords. The initial installation of the DSR system software uses default passwords. These passwords must be changed during installation. For more information, see Changing DSR Administrative Account Passwords and Changing Internal Web Service Passwords sections.
- Obtain and install X.509 web certificates for GUI and MMI access. The DSR system ships with a self-signed certificate that must be replaced before the system is put into operation. For more information, see the Certificate Management section.
- Learn and use the Oracle Communications Diameter Signaling Router security features. For more information, see Implement Oracle Communications Diameter Signaling Router Security and Optional IPsec Configuration sections.
- Keep up to date on security information. Oracle regularly issues security alerts for important vulnerability fixes. It is advisable to install the applicable security patches as soon as possible. For more details, see Oracle Security Alerts.

2.2 Accessing Diameter Signaling Router System

There are several ways to access the Oracle Communications Diameter Signaling Router (DSR) system. They are as follows:

- **Web browser GUI** – The client access to the DSR GUI for remote administration requires a web browser that supports a TLS 1.2 or TLS 1.3 enabled session. This application is designed to work with most modern HTML5 compliant browsers and uses JavaScript and cookies. When you access the DSR system through the GUI interface, the **Log In** screen displays. Type the **Username** and **Password** credentials, and click **Log In** to access the GUI.

Figure 2-2 Diameter Signaling Router Login Page



Oracle System Login

Tue Aug 1 01:12:41 2017 EDT

Log In
Enter your username and password to log in

Username:

Password:

□ Change password

Welcome to the Oracle System Login.

This application is designed to work with most modern HTML5 compliant browsers and uses both JavaScript and cookies. Please refer to the Oracle Software Web Browser Support Policy for details.

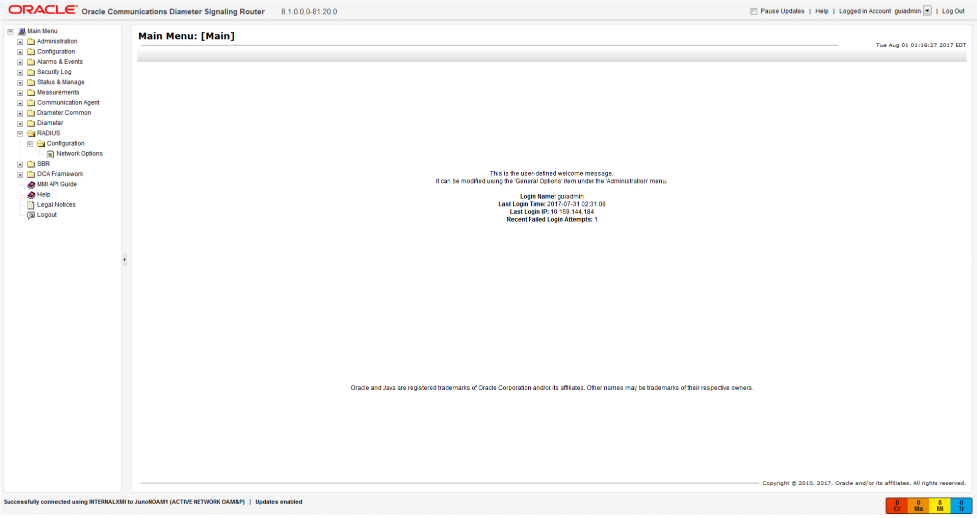
Unauthorized access is prohibited.

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

Copyright © 2010, 2017, Oracle and/or its affiliates. All rights reserved.

When successfully logged in, the Oracle Communications Diameter Signaling Router home page displays. To logout, click the upper-right corner link labelled **Logout** or select the bottom menu item.

Figure 2-3 Diameter Signaling Router Home Page



Tue Aug 01 01:16:27 2017 EDT

This is the user-defined welcome message.
It can be modified using the 'General Options' item under the 'Administration' menu.

Login Name: guiadmin
Last Login Time: 2017-07-31 02:31:08
Last Login IP: 10.159.144.184
Recent Failed Login Attempts: 1

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

Copyright © 2010, 2017, Oracle and/or its affiliates. All rights reserved.

Successfully connected using INTERNALXMI to JunoNOAM1 (ACTIVE NETWORK OAM&P) | Updates enabled

- **CLI via SSH client** – Normal login access is remote through network connections. The client access to the command-line interface (CLI) is with an SSH-capable client such as PUTTY, SecureCRT, or a similar client using the default administrative login account. SSH login is supported on the distinct management interface. To log out, enter the command, log out, and press **Enter**.
- **Local access can be supported by connecting a monitor and a keyboard**. Local access supports CLI only. When successfully logged in, a command-line prompt containing `userid@host` name followed by a `$` prompt displays. There is no requirement to add additional users, but adding users is supported.

> **Note:**
>
> Adding additional users is not supported on all hardware.

3

Implement Oracle Communications Diameter Signaling Router Security

3.1 Diameter Signaling Router Web GUI Standard Features

This section explains the security features of the Oracle Communications Diameter Signaling Router software that are available to the Administrative User through the Graphical User Interface (GUI) using a compatible web browser.

3.1.1 User Administration

A predefined user and group are included in the system for setting up the groups and users. The following are details for this predefined user:

Table 3-1 Predefined User and Group

User	Group	Description
guiadmin	admin	Full access (read or write privileges) to all functions including administration functions

The User Administration page allows the admin user to add, modify, enable, or delete user accounts. Each user that is allowed to access the DSR GUI, is assigned with a unique Username. The correct username and its password must be provided during login.

If a user has made three consecutive unsuccessful login attempts, then that user's account is disabled. The value of the number of failed login attempts, before disabling an account, can be configured through **Administrations** > **Options**. This value can be set between 0-10. The default value is 3. If the value is set to 0, then the user account is not disabled for unsuccessful login attempts.

Each user can be assigned to one or more groups. Only a user with **user admin** or **group admin** privilege can make changes to the user accounts or groups. For more details on user administration, see the *Users Administration* section in *DSR Operation, Administration, and Maintenance (OAM) Guide*.

3.1.1.1 Establishing Groups and Group Permissions

You can assign each GUI user to one or more groups. The **Groups Administration** page enables you to create, modify, and delete user groups. Also, you can assign permissions to the group. The permissions determine the functions and restrictions for the users belonging to that group. The permissions on this page are categorized as follows:

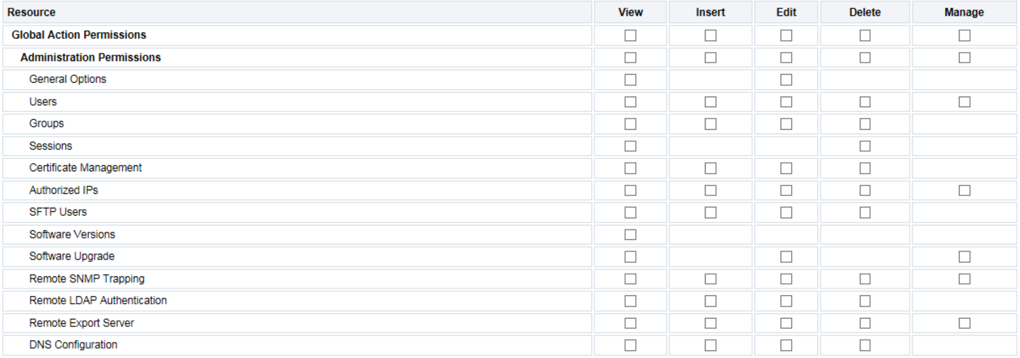
- Global Action Permissions
- Administration Permissions

- Configuration Permissions
- Alarms & Events Permissions
- Security Log Permissions
- Status & Manage Permissions
- Measurements Permissions
- Communication Agent Configuration Permissions
- Communication Agent Maintenance Permissions
- Diameter Configuration Permissions
- Diameter Maintenance Permissions
- Diameter Diagnostics Permissions
- Diameter Mediation Permissions
- Diameter Troubleshooting with IDIH Permissions
- Diameter AVP Dictionary Permissions

For more information about the available permissions for the groups, see the *Group Administration* section in *DSR Operation, Administration, and Maintenance (OAM) Guide*.

For non-admin users, a group with restricted authority is essential. To prevent non-admin users from setting up new users and groups, ensure that **Users** and **Groups** under the **Administration Permissions** section are unchecked, as shown in the following image.

Figure 3-1 Global Action and Administration Permissions



Resource	View	Insert	Edit	Delete	Manage
Global Action Permissions	□	□	□	□	□
Administration Permissions	□	□	□	□	□
General Options	□		□		
Users	□	□	□	□	□
Groups	□	□	□	□	
Sessions	□			□	
Certificate Management	□	□	□	□	
Authorized IPs	□	□	□	□	□
SFTP Users	□	□	□	□	
Software Versions	□				
Software Upgrade	□		□		□
Remote SNMP Trapping	□	□	□	□	□
Remote LDAP Authentication	□	□	□	□	
Remote Export Server	□	□	□	□	□
DNS Configuration	□	□	□	□	

3.1.1.2 Creating Users and Assigning to Groups

Before adding a user, determine which user group the user must be assigned based on the user’s operational role. The group assignment determines the functions a user can access. A user must either have user or group administrative privileges to view or make changes to the user accounts or groups. The admin user can set up or change user accounts and groups, enable or disable user accounts, set password expiration intervals, and change the user passwords.

The Insert User page displays the following elements:

- User Name
- Group
- Authentication Options
- Access Allowed
- Maximum Concurrent Logins
- Session Inactivity Limit
- Comment

For more information, see the *Administration* chapter in *DSR Operation, Administration, and Maintenance (OAM) Guide.*

The User Administration page allows the admin user to perform the following actions:

- Add a New User
- View User Account Information
- Update User Account Information
- Delete a User
- Enable or Disable a User Account
- Change a User’s Assigned Group
- Generate a User Report
- Change Password

For more information, see the *Administration* chapter in the *Operation, Administration, and Maintenance (OAM) Guide.*

3.1.2 User Authentication

In the DSR GUI, Users are authenticated using either login credentials or Single Sign-On (SSO). For more information about setting up a password, see the *Passwords* section in the *Operation, Administration, and Maintenance (OAM) Guide*.

You can configure SSO to work with or without a shared LDAP authentication server. If the SSO is configured to work with an LDAP server, then SSO will require remote (LDAP) authentication for account access on an account-by-account basis. For more information about LDAP authentication, see the *Operation, Administration, and Maintenance (OAM) Guide*.

3.1.2.1 Passwords

The Administrator can perform password configurations such as setting passwords, password history rules, and password expiration. In the DSR GUI, the password configurations can be performed from the Users Administration page. For more information, see the *Administration* chapter in the *Operation, Administration, and Maintenance (OAM) Guide*.

3.1.2.2 Changing DSR Administrative Account Passwords

The System Installation procedure creates the following default accounts:

- guiadmin – for DSR GUI

- root – for CLI
- admusr – for CLI

The installation procedure also conveys the passwords for the accounts created. As a security measure, these passwords must be changed.

To change the default password of an account created for the GUI access, see the *Administration* chapter in the *Operation, Administration, and Maintenance (OAM) Guide*.

To change the Operating System (OS) account passwords for a CLI account, see the Changing OS User Account Default Passwords section.

3.1.2.3 Password Complexity

Password complexity refers to the password selection requirements for better security. The user must ensure that the following conditions are fulfilled for a password to be valid:

- A password must contain 8 to 16 characters.
- A password must contain at least three of the four types of characters such as numeric, lower case letters, upper case letters, or special characters. For example: ! @ # $ % ^ & * ? ~.
- A password must not be the same as the Username or contain the Username in any part of the password. For example, `Username=jsmith` and `password=$@jsmithJS` would be **invalid**.
- A password cannot be the inverse of the Username. For example, `Username=jsmith` and `password=$@htimsj` would be **invalid**.
- The user must not re-use the last three passwords.

For configuring the complexity of the password, set the required values in the `MaxPasswordHistory` field on the **Administration** > **General Options** screen in the user interface.

3.1.2.4 Password Expiration

Password expiration is enforced the first time a user logs in to the user interface. The admin user grants the new user with a temporary password during the initial user account setup. After logging in for the first time using the temporary password, the user interface forces the user to change the password. The user is re-directed to a password changing page that requires the user to enter the old password and then enter a new password twice.

The admin user can also configure the password expiration parameters on a system-wide basis. By default, password expiration occurs after 90 days. For more information about how to configure password expiration, see the *Configuring the Expiration of Password* section in the *Operation, Administration, and Maintenance (OAM) Guide*.

3.1.2.5 Restricting Concurrent Logins

The Insert User page has a `Maximum Concurrent Logins` field. The value in this field indicates the maximum number of concurrent logins a user can perform for each server. This feature cannot be enabled for users belonging to the Admin group. The value of this field can be set from 0 to 50.

The User Administration page has a `Concurrent Logins Allowed` field. The value in this field is the concurrent number of logins allowed.

> **Note:**
>
> For restricting the number of concurrent login instances for OS users, contact My Oracle Support.

3.1.2.6 External Authentication

Users can be authenticated remotely where an external LDAP server is used to perform the authentication.

3.1.2.7 LDAP Authentication for Users

You can configure, update, or delete LDAP authentication servers under the Remote Servers option. If multiple LDAP servers are configured, then the first available server in the list is used to perform the authentication. The secondary server is used only if the first server is unavailable for authentication.

The following elements are required to configure an LDAP server:

- Hostname
- Account Domain Name
- Account Domain Name Short
- Port
- Base DN
- Password
- Account Filter Format
- Account Canonical Form
- Referrals
- Bind Requires DN

For more information on how to configure the LDAP server, see the *LDAP Authentication* section in the *Operation, Administration, and Maintenance (OAM) Guide*.

3.1.2.8 SSO Authentication for Users

Single Sign-On (SSO) allows the user to log into multiple servers within an SSO zone by using a shared certificate among the subject servers within the zone. Once a user is authenticated successfully with any system in the SSO domain, the user can access other systems in the SSO zone without re-entering the authentication credentials.

When two zones in the SSO domain exchange certificates, a trusted relationship is established between the zones and all the systems grouped into the zone, expanding the authenticated login capability to servers in both zones. For more information on how to configure SSO zones, see the *Certificate Management* section in the *Operation, Administration, and Maintenance (OAM) Guide*.

3.1.2.9 Password Strengthening Procedures

This section describes various procedures to set the password strength for each and every server in the topology.

3.1.2.9.1 Setting Password Strength with Minimum Digit Characters

This section describes the procedure to set a strong password using minimum digits for each and every server in the topology.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Check out the file `system-auth` and `password-auth`:

   ```
   $ sudo rcstool co /etc/pam.d/system-auth
   $ sudo rcstool co /etc/pam.d/password-auth
   ```

3. Run the following commands:

   ```
   $ sudo sed -i --follow-symlinks "/pam_cracklib.so/ s/$/
   dcredit=-1/" /etc/pam.d/system-auth
   $ sudo sed -i --follow-symlinks "/pam_cracklib.so/ s/$/
   dcredit=-1/" /etc/pam.d/password-auth
   ```

4. Check in the file `system-auth` and `password-auth`:

   ```
   $ sudo rcstool ci /etc/pam.d/system-auth
   $ sudo rcstool ci /etc/pam.d/password-auth
   ```

3.1.2.9.2 Setting Password Strength with Minimum Uppercase Characters

This section describes the procedure to set a strong password using minimum uppercase characters for each and every server in the topology.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Check out the file `system-auth` and `password-auth`:

   ```
   $ sudo rcstool co /etc/pam.d/system-auth
   $ sudo rcstool co /etc/pam.d/password-auth
   ```

3. Run the following commands:

   ```
   $ sudo sed -i --follow-symlinks "/pam_cracklib.so/ s/$/ ucredit=-2/" /etc/
   pam.d/system-auth
   $ sudo sed -i --follow-symlinks "/pam_cracklib.so/ s/$/ ucredit=-2/" /etc/
   pam.d/password-auth
   ```

4. Check in the file `system-auth` and `password-auth`:

   ```
   $ sudo rcstool ci /etc/pam.d/system-auth
   $ sudo rcstool ci /etc/pam.d/password-auth
   ```

3.1.2.9.3 Setting Password Strength with Minimum Special Characters

This section describes the procedure to set a strong password using minimum special characters for each and every server in the topology.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Check out the file `system-auth` and `password-auth`:

   ```
   $ sudo rcstool co /etc/pam.d/system-auth
   $ sudo rcstool co /etc/pam.d/password-auth
   ```

3. Run the following commands:

   ```
   $ sudo sed -i --follow-symlinks "/pam_cracklib.so/ s/$/ ocredit=-2/" /etc/
   pam.d/system-auth
   $ sudo sed -i --follow-symlinks "/pam_cracklib.so/ s/$/ ocredit=-2/" /etc/
   pam.d/password-auth
   ```

4. Check in the file `system-auth` and `password-auth`:

   ```
   $ sudo rcstool ci /etc/pam.d/system-auth
   $ sudo rcstool ci /etc/pam.d/password-auth
   ```

3.1.2.9.4 Setting Password Strength with Minimum Lowercase Characters

This section describes the procedure to set a strong password using minimum lowercase characters for each and every server in the topology.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Check out the file system-auth and password-auth:

   ```
   $ sudo rcstool co /etc/pam.d/system-auth
   $ sudo rcstool co /etc/pam.d/password-auth
   ```

3. Run the following commands:

   ```
   $ sudo sed -i --follow-symlinks "/pam_cracklib.so/ s/$/
   lcredit=-2/" /etc/pam.d/system-auth
   $ sudo sed -i --follow-symlinks "/pam_cracklib.so/ s/$/
   lcredit=-2/" /etc/pam.d/password-auth
   ```

4. Check in the file system-auth and password-auth:

   ```
   $ sudo rcstool ci /etc/pam.d/system-auth
   $ sudo rcstool ci /etc/pam.d/password-auth
   ```

3.1.2.9.5 Setting Deny for Failed Password Attempts

This section describes the procedure to deny the user access for failed password attempts.

Run the following procedure for each and every server in the topology:

1. Log in as admusr on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Check out the file system-auth and password-auth:

   ```
   $ sudo rcstool co /etc/pam.d/system-auth
   $ sudo rcstool co /etc/pam.d/password-auth
   ```

3. Run the following commands:

   ```
   $ sudo sed -i --follow-symlinks "/
   ^auth.*sufficient.*pam_unix.so.*/i auth        required
   pam_faillock.so preauth silent deny=5 unlock_time=604800
   fail_interval=900" /etc/pam.d/system-auth

   $ sudo sed -i --follow-symlinks "/
   ^auth.*sufficient.*pam_unix.so.*/a auth        [default=die]
   pam_faillock.so authfail deny=5 unlock_time=604800
   fail_interval=900" /etc/pam.d/system-auth

   $ sudo sed -i --follow-symlinks "/^account.*required.*pam_unix.so/i
   account     required      pam_faillock.so" /etc/pam.d/system-auth

   $ sudo sed -i --follow-symlinks "/
   ^auth.*sufficient.*pam_unix.so.*/i auth        required
   pam_faillock.so preauth silent deny=5 unlock_time=604800
   fail_interval=900" /etc/pam.d/password-auth
   ```

```
$ sudo sed -i --follow-symlinks "/^auth.*sufficient.*pam_unix.so.*/a
auth        [default=die] pam_faillock.so authfail deny=5
unlock_time=604800 fail_interval=900" /etc/pam.d/password-auth

$ sudo sed -i --follow-symlinks "/^account.*required.*pam_unix.so/i
account     required      pam_faillock.so" /etc/pam.d/password-auth
```

4. Check in the file `system-auth` and `password-auth`:

   ```
   $ sudo rcstool ci /etc/pam.d/system-auth
   $ sudo rcstool ci /etc/pam.d/password-auth
   ```

3.1.2.9.6 Setting Minimum Password Length

This section describes the procedure to set the minimum length for a password.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Check out the file `system-auth` and grep for variable ‘*minlen*’ in the file using the following command:

   ```
   $ sudo rcstool co /etc/pam.d/system-auth
   $ grep minlen /etc/pam.d/system-auth
   ```

   - If a result is returned, then run the following command:

     ```
     $ sudo sed -i "/password.*requisite.*pam_cracklib.so/s/minlen[^ ]*/
     minlen=14/" /etc/pam.d/system-auth
     ```

   - If no result is returned, then run the following command:

     ```
     $ sudo sed -i "/password.*requisite.*pam_cracklib.so/s/$/
     minlen=14/" /etc/pam.d/system-auth
     ```

3. Check in the file `system-auth`:

   ```
   $ sudo rcstool ci /etc/pam.d/system-auth
   ```

3.1.2.10 Login and Welcome Banner Customization

The DSR GUI allows to enter custom messages to the Login screen and Welcome message after successful user login. The **Administration** > **Options** page allows the admin user to view a list of global options.

To enter a custom message to the Login screen, the admin user can enter the required message in the `LoginMessage` field. This enables the user to view the customized login message on the login screen.

To enter a custom message to the Welcome Banner, the admin user can enter the required message in the `WelcomeMessage` field. This enables the user to view the customized welcome message after successful login.

3.1.2.11 SSH Security Hardening Procedures

This section describes the security hardening procedures using Secure Socket Shell (SSH).

3.1.2.11.1 Setting SSH Client Alive Count

This section describes the procedure to set the count for SSH client.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the file `sshd_config` and grep for variable `ClientAliveCountMax` in the file :

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   $ sudo grep ^ClientAliveCountMax /etc/ssh/sshd_config
   ```

3. Run the following command if no result is returned after running step 2:

   ```
   $ sudo echo "ClientAliveCountMax 0" >> /etc/ssh/sshd_config
   ```

   Run the following command if some result is returned after running step 2:

   ```
   $ sudo sed -i "s/ClientAliveCountMax.*/ClientAliveCountMax
   0/g" /etc/ssh/sshd_config
   ```

4. Check in the file `sshd_config`:

   ```
   $ sudo rcstool ci /etc/ssh/sshd_config
   ```

3.1.2.11.2 Disabling SSH Access through Empty Passwords

This section describes the procedure to disable the SSH access through empty passwords.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the file `sshd_config` and grep for variable`PermitEmptyPasswords` in the file :

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   $ sudo grep PermitEmptyPasswords /etc/ssh/sshd_config
   ```

3. Run the following command if no result is returned after running step 2:

   ```
   $ sudo echo "PermitEmptyPasswords no" >> /etc/ssh/sshd_config
   ```

   Run the following command if some result is returned after running step 2:

   ```
   $ sudo sed -i '/PermitEmptyPasswords/c\PermitEmptyPasswords no' /etc/ssh/
   sshd_config
   ```

4. Check in the file `sshd_config`:

   ```
   $ sudo rcstool ci /etc/ssh/sshd_config
   ```

3.1.2.11.3 Enabling SSH Warning Banner

This section describes the procedure to enable the SSH warning banner.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the file `sshd_config` and grep for variable `Banner` in the file :

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   $ sudo grep Banner /etc/ssh/sshd_config
   ```

3. Run the following command if no result is returned after running step 2:

   ```
   $ sudo echo “Banner /etc/issue" >> /etc/ssh/sshd_config
   ```

   Run the following command if some result is returned after running step 2:

   ```
   $ sudo sed -i '/Banner/c\Banner \/etc\/issue' /etc/ssh/sshd_config
   ```

4. Check in the file `sshd_config`:

   ```
   $ sudo rcstool ci /etc/ssh/sshd_config
   ```

3.1.2.11.4 Denying SSH Environment Options

This section describes the procedure to deny SSH environment options on each and every server in the topology.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the file `sshd_config` and grep for variable `PermitUserEnvironment` in the file :

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   $ sudo grep PermitUserEnvironment /etc/ssh/sshd_config
   ```

3. Run the following command if no result is returned after running step 2:

   ```
   $ sudo echo "PermitUserEnvironment no" >> /etc/ssh/sshd_config
   ```

   Run the following command if some result is returned after running step 2:

   ```
   $ sudo sed -i '/PermitUserEnvironment/c\PermitUserEnvironment
   no' /etc/ssh/sshd_config
   ```

4. Check in the file `sshd_config`:

   ```
   $ sudo rcstool ci /etc/ssh/sshd_config
   ```

3.1.2.11.5 Generating RSA SSH Key for Admin User

This section describes the procedure to generate a passphrase protected RSA SSH key for 'admusr' User Account.

Run the following procedure on each server in the topology. The order of execution in the topology must be from level 'A' servers to level 'C' servers:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to stop the `apwSoapServer` process:

   ```
   $ sudo pm.set off apwSoapServer
   ```

3. Run the following command to go to `.ssh` directory and remove the old DSA keys if they exist:

   ```
   $ cd /home/admusr/.ssh
   $ sudo rm -rf id_dsa id_dsa.pub
   ```

4. Run the following command to generate a new RSA key:

   ```
   $ ssh-keygen -t rsa -b 4096
   ```

   Provide the desired location to save the key or it can be left blank. On leaving it blank, the default location `/home/admusr/.ssh/id_rsa` is used:

   ```
   $ Enter file in which to save the key (/home/admusr/.ssh/id_rsa):
   ```

   Enter the passphrase:

   ```
   $ Enter passphrase (empty for no passphrase):
   ```

   Confirm the passphrase again:

   ```
   $ Enter same passphrase again:
   ```

   A password protected RSA key is generated successfully.

5. Run the following command to start the `apwSoapServer` process:

   ```
   $ sudo pm.set on apwSoapServer
   ```

After 60 seconds, the server will use the generated RSA key.
After running the procedure, any key-based SSH login for the 'admusr' account prompts for a passphrase. Setting a passphrase on the key affects the execution of procedures that require SSH access using the ‘admusr’ account. The admin user is prompted to enter the passphrase for each SSH access. For more information on how to run the procedures that require SSH access, see the Changing TPD Web Service Password and Changing the Configuration Web Services Password sections.

3.1.2.11.6 Setting SSH Log Level

This section describes the procedure to set SSH log level to INFO.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the file `sshd_config`:

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   ```

3. Run the following command:

   ```
   $ sudo sed -i '/LogLevel/c\LogLevel INFO' /etc/ssh/sshd_config
   ```

4. Check in the file `sshd_config`:

   ```
   $ sudo rcstool ci /etc/ssh/sshd_config
   ```

3.1.2.11.7 Enabling SSH IgnoreRhosts

This section describes the procedure to enable SSH IgnoreRhosts.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the file `sshd_config`:

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   ```

3. Run the following command:

   ```
   $ sudo sed -i '/IgnoreRhosts/c\IgnoreRhosts yes' /etc/ssh/
   sshd_config
   ```

4. Check in the file `sshd_config`:

   ```
   $ sudo rcstool ci /etc/ssh/sshd_config
   ```

3.1.2.11.8 Disabling SSH X11 Forwarding

This section describes the procedure to disable SSH X11 Forwarding.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the file `sshd_config`:

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   ```

3. Run the following command:

   ```
   $ sudo sed -i '/X11Forwarding yes/s/^/#/g' /etc/ssh/sshd_config
   $ sudo sed -i '/X11Forwarding no/s/^#//g' /etc/ssh/sshd_config
   ```

4. Check in the file `sshd_config`:

   ```
   $ sudo rcstool ci /etc/ssh/sshd_config
   ```

3.1.2.11.9 Disabling SSH HostbasedAuthentication

This section describes the procedure to disable SSH host based authentication.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the file `sshd_config`:

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   ```

3. Run the following command:

   ```
   $ sudo sed -i '/HostbasedAuthentication no/s/^#//g' /etc/ssh/sshd_config
   ```

4. Check in the file `sshd_config`:

   ```
   $ sudo rcstool ci /etc/ssh/sshd_config
   ```

3.1.2.11.10 Setting SSH LoginGraceTime

This section describes the procedure to set the SSH Login grace time to 1 min.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the file `sshd_config`:

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   ```

3. Run the following command:

   ```
   $ sudo sed -i '/LoginGraceTime/c\LoginGraceTime 60' /etc/ssh/sshd_config
   ```

4. Check in the file `sshd_config`:

   ```
   $ sudo rcstool ci /etc/ssh/sshd_config
   ```

3.1.2.11.11 Disabling SSH Insecure Key Exchange Algorithms and Setting Up Key Length

This section describes the procedure to disable `diffie-hellman-group1-sha1` and `gss-group1-sha1` key exchange (Kex) algorithms, and to set the moduli (key length) longer than 1024 bits.

Run the following procedure for each server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check if the keys used are lesser than 1024 bits.

   ```
   $ sudo awk '$5 <= 1024' /etc/ssh/moduli
   ```

3. If no result is returned after running step 2, it means there are no keys lesser than 1024 bits used. You can skip steps 4 and 5.

   Else, check-out the file `moduli`:

   ```
   $ sudo rcstool co /etc/ssh/moduli
   ```

4. Run the following command to configure the SSH service to use `Diffie-Hellman moduli` that are greater than 1024 bits.

   ```
   $ sudo awk '$5 > 1024' /etc/ssh/moduli > tmp$$
   ```

   ```
   $ sudo mv tmp$$ /etc/ssh/moduli
   ```

5. Run the following command to check-in the file `moduli`:

   ```
   $ sudo rcstool ci /etc/ssh/moduli
   ```

6. Run the following command to verify if the `diffie-hellman-group1-sha1` key exchange algorithm is supported:

   ```
   $ sudo sshd -T | grep diffie-hellman-group1-sha1
   ```

7. If no result is returned after running step 6, it means `diffie-hellman-group1-sha1` key exchange algorithm is already disabled. You can skip steps 8 and 9.

   Else, check-out the files `sshd_config` and `ssh_config`:

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   ```

   ```
   $ sudo rcstool co /etc/ssh/ssh_config
   ```

8. Run the following commands to disable `diffie-hellman-group1-sha1` key exchange algorithm:

   ```
   $ sudo grep -iq "^[ ]*KexAlgorithms" /etc/ssh/ssh_config && sudo sed -i
   "s/^[ ]*KexAlgorithms.*/KexAlgorithms $(sudo sshd -T | grep diffie-
   hellman-group1-sha1 | awk 'tolower($1)==kexalgorithms {$1="\n"$1;}
   {print $2;}' | sed 's/diffie-hellman-group1-sha1,//g; s/,diffie-hellman-
   group1-sha1//g')/i" /etc/ssh/ssh_config || sudo sed -i "$ a
   KexAlgorithms $(sudo sshd -T | grep diffie-hellman-group1-sha1 | awk
   'tolower($1)==kexalgorithms {$1="\n"$1;} {print $2;}' | sed 's/diffie-
   hellman-group1-sha1,//g; s/,diffie-hellman-group1-sha1//g')" /etc/ssh/
   ssh_config
   ```

   ```
   $ sudo grep -iq "^[ ]*KexAlgorithms" /etc/ssh/sshd_config && sudo sed -i
   "s/^[ ]*KexAlgorithms.*/KexAlgorithms $(sudo sshd -T | grep diffie-
   hellman-group1-sha1 | awk 'tolower($1)==kexalgorithms {$1="\n"$1;}
   {print $2;}' | sed 's/diffie-hellman-group1-sha1,//g; s/,diffie-hellman-
   group1-sha1//g')/i" /etc/ssh/sshd_config || sudo sed -i "$ a
   KexAlgorithms $(sudo sshd -T | grep diffie-hellman-group1-sha1 | awk
   'tolower($1)==kexalgorithms {$1="\n"$1;} {print $2;}' | sed 's/diffie-
   hellman-group1-sha1,//g; s/,diffie-hellman-group1-sha1//g')" /etc/ssh/
   sshd_config
   ```

9. Run the following commands to check-in the files `sshd_config` and `ssh_config`:

   ```
   $ sudo rcstool ci /etc/ssh/sshd_config
   ```

   ```
   $ sudo rcstool ci /etc/ssh/ssh_config
   ```

10. Run the following command to check if `gss-group1-sha1-` key exchange algorithm is supported:

    ```
    $ sudo sshd -T | grep gss-group1-sha1-
    ```

11. If no result is returned after running step 10, it means `gss-group1-sha1-` key exchange algorithm is already disabled. You can skip steps 12 and 13.

    Else, check-out the files `sshd_config` and `ssh_config`:

    ```
    $ sudo rcstool co /etc/ssh/sshd_config
    ```

    ```
    $ sudo rcstool co /etc/ssh/ssh_config
    ```

12. Run the following commands to disable `gss-group1-sha1-` key exchange algorithm:

    ```
    $ sudo grep -iq "^[ ]*GSSAPIKexAlgorithms" /etc/ssh/ssh_config && sudo
    sed -i "s/^[ ]*GSSAPIKexAlgorithms.*/GSSAPIKexAlgorithms $(sudo sshd -T |
    grep gss-group1-sha1- | awk 'tolower($1)==gssapikexalgorithms
    {$1="\n"$1;} {print $2;}' | sed 's/gss-group1-sha1-,//g; s/,gss-group1-
    sha1-//g')/i" /etc/ssh/ssh_config || sudo sed -i "$ a
    GSSAPIKexAlgorithms $(sudo sshd -T | grep gss-group1-sha1- | awk
    ```

```
'tolower($1)==gssapikexalgorithms {$1="\n"$1;} {print $2;}' | sed
's/gss-group1-sha1-,//g; s/,gss-group1-sha1-//g')" /etc/ssh/
ssh_config
```

```
$ sudo grep -iq "^[ ]*GSSAPIKexAlgorithms" /etc/ssh/sshd_config &&
sudo sed -i "s/^[ ]*GSSAPIKexAlgorithms.*/GSSAPIKexAlgorithms $
(sudo sshd -T | grep gss-group1-sha1- | awk
'tolower($1)==gssapikexalgorithms {$1="\n"$1;} {print $2;}' | sed
's/gss-group1-sha1-,//g; s/,gss-group1-sha1-//g')/i" /etc/ssh/
sshd_config || sudo sed -i "$ a GSSAPIKexAlgorithms $(sudo sshd -T
| grep gss-group1-sha1- | awk 'tolower($1)==gssapikexalgorithms
{$1="\n"$1;} {print $2;}' | sed 's/gss-group1-sha1-,//g; s/,gss-
group1-sha1-//g')" /etc/ssh/sshd_config
```

13. Run the following commands to check-in the files `sshd_config` and `ssh_config`:

```
$ sudo rcstool ci /etc/ssh/sshd_config
```

```
$ sudo rcstool ci /etc/ssh/ssh_config
```

14. Run the following command to restart **sshd** service:

```
$ sudo service sshd restart
```

3.1.2.11.12 Disabling SSH Weak Key Exchange Algorithms

This section describes the procedure to disable SSH weak Key Exchange (Kex) algorithms. Only the strong and secure KexAlgorithms and GSSAPIKexAlgorithms are to be enabled.

Perform the following procedure on each server in the topology:

1. Log in as `admusr` on the server.

```
login: admusr
Password: <current admin user password>
```

2. To check-out the `/etc/sysconfig/sshd` file, run the following command:

```
$ sudo rcstool co /etc/sysconfig/sshd
```

3. To check if the `CRYPTO_POLICY` line in the `/etc/sysconfig/sshd` file is commented, run the following command:

```
$ sudo grep -i "^[[:space:]#]*CRYPTO_POLICY" /etc/sysconfig/sshd
```

4. If a result after running step 3 specifies that the `CRYPTO_POLICY` line in `/etc/sysconfig/sshd` file is commented, uncomment the line to opt out of the

system-wide crypto policies for OpenSSH server by running the following command:

```
$ sudo sed -i "s/^[ #]*CRYPTO_POLICY.*/CRYPTO_POLICY=/i" /etc/sysconfig/
sshd
```

> **Note:**
>
> Else, you can skip this step.

5. If the `04-configure-SSH.conf` custom configuration file, to specify the crypto policy, is not located in the `/etc/ssh/ssh_config.d/` directory, then create one configuration file to opt out of the system-wide crypto policies for OpenSSH client, by running the following commands:

   ```
   $ sudo touch /etc/ssh/ssh_config.d/04-configure-SSH.conf
   ```

   ```
   $ sudo chmod 644 /etc/ssh/ssh_config.d/04-configure-SSH.conf
   ```

   ```
   $ sudo chown root:root /etc/ssh/ssh_config.d/04-configure-SSH.conf
   ```

6. To check-out the `sshd_config` and `04-configure-SSH.conf` files, run the following commands:

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   ```

   ```
   $ sudo rcstool co /etc/ssh/ssh_config.d/04-configure-SSH.conf
   ```

7. To check if the required `KexAlgorithms` in the server side are enabled in order, run the following command:

   ```
   $ sudo sshd -T | grep -i "^kexalgorithms" | grep "[[:space:]]curve25519-
   sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp384,ecdh-sha2-
   nistp256,ecdh-sha2-nistp521,diffie-hellman-group16-sha512,diffie-hellman-
   group18-sha512,diffie-hellman-group14-sha256,diffie-hellman-group-
   exchange-sha256$"
   ```

8. If a result after running step 7 specifies that the required `KexAlgorithms` in the server side are enabled in order, you can skip this step.

   Else, enable them in order in the `sshd_config` file, by running the following command:

   ```
   $ sudo grep -iq "^[ ]*KexAlgorithms" /etc/ssh/sshd_config && sudo sed -i
   "s/^[ ]*KexAlgorithms.*/KexAlgorithms curve25519-sha256,curve25519-
   sha256@libssh.org,ecdh-sha2-nistp384,ecdh-sha2-nistp256,ecdh-sha2-
   nistp521,diffie-hellman-group16-sha512,diffie-hellman-group18-
   sha512,diffie-hellman-group14-sha256,diffie-hellman-group-exchange-
   sha256/i" /etc/ssh/sshd_config || sudo sed -i "$ a KexAlgorithms
   curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp384,ecdh-
   sha2-nistp256,ecdh-sha2-nistp521,diffie-hellman-group16-sha512,diffie-
   ```

```
hellman-group18-sha512,diffie-hellman-group14-sha256,diffie-hellman-
group-exchange-sha256" /etc/ssh/sshd_config
```

9. To check if the required `KexAlgorithms` in the client side are enabled in order, run the following command:

   ```
   $ sudo grep -i "^[[:space:]]*kexalgorithms" /etc/ssh/
   ssh_config.d/04-configure-SSH.conf | grep "[[:space:]]curve25519-
   sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp384,ecdh-sha2-
   nistp256,ecdh-sha2-nistp521,diffie-hellman-group16-sha512,diffie-
   hellman-group18-sha512,diffie-hellman-group14-sha256,diffie-hellman-
   group-exchange-sha256$"
   ```

10. If a result after running step 9 specifies that the required `KexAlgorithms` in the client side are enabled in order, you can skip this step.

    Else, enable them in order in the `04-configure-SSH.conf` file.

    If the `04-configure-SSH.conf` file is empty (zero-byte file), run the following command:

    ```
    $ sudo echo "KexAlgorithms curve25519-sha256,curve25519-
    sha256@libssh.org,ecdh-sha2-nistp384,ecdh-sha2-nistp256,ecdh-sha2-
    nistp521,diffie-hellman-group16-sha512,diffie-hellman-group18-
    sha512,diffie-hellman-group14-sha256,diffie-hellman-group-exchange-
    sha256" | sudo tee -a /etc/ssh/ssh_config.d/04-configure-SSH.conf
    > /dev/null
    ```

    If the `04-configure-SSH.conf` file is non-empty, run the following command:

    ```
    $ sudo grep -iq "^[ ]*KexAlgorithms" /etc/ssh/ssh_config.d/04-
    configure-SSH.conf && sudo sed -i "s/^[ ]*KexAlgorithms.*/
    KexAlgorithms curve25519-sha256,curve25519-sha256@libssh.org,ecdh-
    sha2-nistp384,ecdh-sha2-nistp256,ecdh-sha2-nistp521,diffie-hellman-
    group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group14-
    sha256,diffie-hellman-group-exchange-sha256/i" /etc/ssh/
    ssh_config.d/04-configure-SSH.conf || sudo sed -i "$ a
    KexAlgorithms curve25519-sha256,curve25519-sha256@libssh.org,ecdh-
    sha2-nistp384,ecdh-sha2-nistp256,ecdh-sha2-nistp521,diffie-hellman-
    group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group14-
    sha256,diffie-hellman-group-exchange-sha256" /etc/ssh/
    ssh_config.d/04-configure-SSH.conf
    ```

11. To check if the required `GSSAPIKexAlgorithms` in the server side are enabled in order, run the following command:

    ```
    $ sudo sshd -T | grep -i "^gssapikexalgorithms" | grep
    "[[:space:]]gss-group14-sha256-,gss-group16-sha512-,gss-nistp256-
    sha256-,gss-curve25519-sha256-$"
    ```

12. If a result after running step 11 specifies that the required `GSSAPIKexAlgorithms` in the server side are enabled in order, you can skip this step.

Else, enable them in order in the `sshd_config` file, by running the following command:

```
$ sudo grep -iq "^[ ]*GSSAPIKexAlgorithms" /etc/ssh/sshd_config && sudo
sed -i "s/^[ ]*GSSAPIKexAlgorithms.*/GSSAPIKexAlgorithms gss-group14-
sha256-,gss-group16-sha512-,gss-nistp256-sha256-,gss-curve25519-
sha256-/i" /etc/ssh/sshd_config || sudo sed -i "$ a GSSAPIKexAlgorithms
gss-group14-sha256-,gss-group16-sha512-,gss-nistp256-sha256-,gss-
curve25519-sha256-" /etc/ssh/sshd_config
```

13. To check if the required `GSSAPIKexAlgorithms` in the client side are enabled in order, run the following command:

```
$ sudo grep -i "^[[:space:]]*gssapikexalgorithms" /etc/ssh/
ssh_config.d/04-configure-SSH.conf | grep "[[:space:]]gss-group14-
sha256-,gss-group16-sha512-,gss-nistp256-sha256-,gss-curve25519-sha256-$"
```

14. If a result after running step 13 specifies that the required `GSSAPIKexAlgorithms` in the client side are enabled in order, you can skip this step.

    Else, enable them in order in the `04-configure-SSH.conf` file.

    If the `04-configure-SSH.conf` file is empty (zero-byte file), run the following command:

```
$ sudo echo "GSSAPIKexAlgorithms gss-group14-sha256-,gss-group16-
sha512-,gss-nistp256-sha256-,gss-curve25519-sha256-" | sudo tee -
a /etc/ssh/ssh_config.d/04-configure-SSH.conf > /dev/null
```

    If the `04-configure-SSH.conf` file is non-empty, run the following command:

```
$ sudo grep -iq "^[ ]*GSSAPIKexAlgorithms" /etc/ssh/ssh_config.d/04-
configure-SSH.conf && sudo sed -i "s/^[ ]*GSSAPIKexAlgorithms.*/
GSSAPIKexAlgorithms gss-group14-sha256-,gss-group16-sha512-,gss-nistp256-
sha256-,gss-curve25519-sha256-/i" /etc/ssh/ssh_config.d/04-configure-
SSH.conf || sudo sed -i "$ a GSSAPIKexAlgorithms gss-group14-sha256-,gss-
group16-sha512-,gss-nistp256-sha256-,gss-curve25519-sha256-" /etc/ssh/
ssh_config.d/04-configure-SSH.conf
```

15. To check-in the `/etc/sysconfig/sshd, sshd_config`, and `04-configure-SSH.conf` files, run the following commands:

```
$ sudo rcstool ci /etc/sysconfig/sshd
```

```
$ sudo rcstool ci /etc/ssh/sshd_config
```

```
$ sudo rcstool ci /etc/ssh/ssh_config.d/04-configure-SSH.conf
```

16. To restart `sshd` service, run the following command:

```
$ sudo service sshd restart
```

3.1.2.11.13 Disabling SSH Weak Host Key Algorithms, MACs, and Ciphers

This section describes the procedure to disable SSH weak Host Key Algorithms, MACs, and Ciphers. Only the strong and secure Host Key Algorithms, MACs, and Ciphers are to be enabled.
Perform the following procedure on each server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. To check-out the `/etc/sysconfig/sshd` file, run the following command:

   ```
   $ sudo rcstool co /etc/sysconfig/sshd
   ```

3. To check if the `CRYPTO_POLICY` line in the `/etc/sysconfig/sshd` file is commented, run the following command:

   ```
   $ sudo grep -i "^[[:space:]#]*CRYPTO_POLICY" /etc/sysconfig/sshd
   ```

4. If a result after running step 3 specifies that the `CRYPTO_POLICY` line in `/etc/sysconfig/sshd` file is commented, uncomment the line to opt out of the system-wide crypto policies for OpenSSH server by running the following command:

   ```
   $ sudo sed -i "s/^[ #]*CRYPTO_POLICY.*/CRYPTO_POLICY=/i" /etc/
   sysconfig/sshd
   ```

   > **Note:**
   >
   > Else, you can skip this step.

5. If the `04-configure-SSH.conf` custom configuration file, to specify the crypto policy, is not located in the `/etc/ssh/ssh_config.d/` directory, then create one configuration file to opt out of the system-wide crypto policies for OpenSSH client, by running the following commands:

   ```
   $ sudo touch /etc/ssh/ssh_config.d/04-configure-SSH.conf
   ```

   ```
   $ sudo chmod 644 /etc/ssh/ssh_config.d/04-configure-SSH.conf
   ```

   ```
   $ sudo chown root:root /etc/ssh/ssh_config.d/04-configure-SSH.conf
   ```

6. To check-out the `sshd_config` and `04-configure-SSH.conf` files, run the following commands:

   ```
   $ sudo rcstool co /etc/ssh/sshd_config
   ```

   ```
   $ sudo rcstool co /etc/ssh/ssh_config.d/04-configure-SSH.conf
   ```

7. To check if the required `HostKeyAlgorithms` in the server side are enabled in order, run the following command:

   ```
   $ sudo sshd -T | grep -i "^hostkeyalgorithms" | grep "[[:space:]]ssh-
   ed25519,ssh-ed25519-cert-v01@openssh.com,ecdsa-sha2-nistp384,ecdsa-sha2-
   nistp384-cert-v01@openssh.com,rsa-sha2-512,rsa-sha2-512-cert-
   v01@openssh.com,rsa-sha2-256,rsa-sha2-256-cert-v01@openssh.com,ecdsa-sha2-
   nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-
   nistp521,ecdsa-sha2-nistp521-cert-v01@openssh.com$"
   ```

8. If a result after running step 7 specifies that the required `HostKeyAlgorithms` in the server side are enabled in order, you can skip this step.

   Else, enable them in order in the `sshd_config` file, by running the following command:

   ```
   $ sudo grep -iq "^[ ]*HostKeyAlgorithms" /etc/ssh/sshd_config && sudo sed
   -i "s/^[ ]*HostKeyAlgorithms.*/HostKeyAlgorithms ssh-ed25519,ssh-ed25519-
   cert-v01@openssh.com,ecdsa-sha2-nistp384,ecdsa-sha2-nistp384-cert-
   v01@openssh.com,rsa-sha2-512,rsa-sha2-512-cert-v01@openssh.com,rsa-
   sha2-256,rsa-sha2-256-cert-v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-
   nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp521,ecdsa-sha2-nistp521-
   cert-v01@openssh.com/i" /etc/ssh/sshd_config || sudo sed -i "$ a
   HostKeyAlgorithms ssh-ed25519,ssh-ed25519-cert-v01@openssh.com,ecdsa-sha2-
   nistp384,ecdsa-sha2-nistp384-cert-v01@openssh.com,rsa-sha2-512,rsa-
   sha2-512-cert-v01@openssh.com,rsa-sha2-256,rsa-sha2-256-cert-
   v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-nistp256-cert-
   v01@openssh.com,ecdsa-sha2-nistp521,ecdsa-sha2-nistp521-cert-
   v01@openssh.com" /etc/ssh/sshd_config
   ```

9. To check if the required `HostKeyAlgorithms` in the client side are enabled in order, run the following command:

   ```
   $ sudo grep -i "^[[:space:]]*hostkeyalgorithms" /etc/ssh/ssh_config.d/04-
   configure-SSH.conf | grep "[[:space:]]ssh-ed25519,ssh-ed25519-cert-
   v01@openssh.com,ecdsa-sha2-nistp384,ecdsa-sha2-nistp384-cert-
   v01@openssh.com,rsa-sha2-512,rsa-sha2-512-cert-v01@openssh.com,rsa-
   sha2-256,rsa-sha2-256-cert-v01@openssh.com,ecdsa-sha2-nistp256,ecdsa-sha2-
   nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp521,ecdsa-sha2-nistp521-
   cert-v01@openssh.com$"
   ```

10. If a result after running step 9 specifies that the required `HostKeyAlgorithms` in the client side are enabled in order, you can skip this step.

    Else, enable them in order in the `04-configure-SSH.conf` file.

If the `04-configure-SSH.conf` file is empty (zero-byte file), run the following command:

```
$ echo "HostKeyAlgorithms ssh-ed25519,ssh-ed25519-cert-
v01@openssh.com,ecdsa-sha2-nistp384,ecdsa-sha2-nistp384-cert-
v01@openssh.com,rsa-sha2-512,rsa-sha2-512-cert-v01@openssh.com,rsa-
sha2-256,rsa-sha2-256-cert-v01@openssh.com,ecdsa-sha2-
nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-
nistp521,ecdsa-sha2-nistp521-cert-v01@openssh.com" | sudo tee -
a /etc/ssh/ssh_config.d/04-configure-SSH.conf > /dev/null
```

If the `04-configure-SSH.conf` file is non-empty, run the following command:

```
$ sudo grep -iq "^[ ]*HostKeyAlgorithms" /etc/ssh/ssh_config.d/04-
configure-SSH.conf && sudo sed -i "s/^[ ]*HostKeyAlgorithms.*/
HostKeyAlgorithms ssh-ed25519,ssh-ed25519-cert-
v01@openssh.com,ecdsa-sha2-nistp384,ecdsa-sha2-nistp384-cert-
v01@openssh.com,rsa-sha2-512,rsa-sha2-512-cert-v01@openssh.com,rsa-
sha2-256,rsa-sha2-256-cert-v01@openssh.com,ecdsa-sha2-
nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-
nistp521,ecdsa-sha2-nistp521-cert-v01@openssh.com/i" /etc/ssh/
ssh_config.d/04-configure-SSH.conf || sudo sed -i "$ a
HostKeyAlgorithms ssh-ed25519,ssh-ed25519-cert-
v01@openssh.com,ecdsa-sha2-nistp384,ecdsa-sha2-nistp384-cert-
v01@openssh.com,rsa-sha2-512,rsa-sha2-512-cert-v01@openssh.com,rsa-
sha2-256,rsa-sha2-256-cert-v01@openssh.com,ecdsa-sha2-
nistp256,ecdsa-sha2-nistp256-cert-v01@openssh.com,ecdsa-sha2-
nistp521,ecdsa-sha2-nistp521-cert-v01@openssh.com" /etc/ssh/
ssh_config.d/04-configure-SSH.conf
```

11. To check if the required `MACs` in the server side are enabled in order, run the following command:

```
$ sudo sshd -T | grep -i "^macs" | grep "[[:space:]]hmac-sha2-512-
etm@openssh.com,hmac-sha2-256-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-512,hmac-sha2-256,umac-128@openssh.com$"
```

12. If a result after running step 11 specifies that the required `MACs` in the server side are enabled in order, you can skip this step.

    Else, enable them in order in the `sshd_config` file, by running the following command:

```
$ sudo grep -iq "^[ ]*MACs" /etc/ssh/sshd_config && sudo sed -i
"s/^[ ]*MACs.*/MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-
etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-
sha2-256,umac-128@openssh.com/i" /etc/ssh/sshd_config || sudo sed -
i "$ a MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-
etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-
sha2-256,umac-128@openssh.com" /etc/ssh/sshd_config
```

13. To check if the required `MACs` in the client side are enabled in order, run the following command:

    ```
    $ sudo grep -i "^[[:space:]]*macs" /etc/ssh/ssh_config.d/04-configure-
    SSH.conf | grep "[[:space:]]hmac-sha2-512-etm@openssh.com,hmac-sha2-256-
    etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-
    sha2-256,umac-128@openssh.com$"
    ```

14. If a result after running step 13 specifies that the required `MACs` in the client side are enabled in order, you can skip this step.

    Else, enable them in order in the `04-configure-SSH.conf` file.

    If the `04-configure-SSH.conf` file is empty (zero-byte file), run the following command:

    ```
    $ echo "MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-
    etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-
    sha2-256,umac-128@openssh.com" | sudo tee -a /etc/ssh/ssh_config.d/04-
    configure-SSH.conf > /dev/null
    ```

    If the `04-configure-SSH.conf` file is non-empty, run the following command:

    ```
    $ sudo grep -iq "^[ ]*MACs" /etc/ssh/ssh_config.d/04-configure-SSH.conf
    && sudo sed -i "s/^[ ]*MACs.*/MACs hmac-sha2-512-etm@openssh.com,hmac-
    sha2-256-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-
    sha2-256,umac-128@openssh.com/i" /etc/ssh/ssh_config.d/04-configure-
    SSH.conf || sudo sed -i "$ a MACs hmac-sha2-512-etm@openssh.com,hmac-
    sha2-256-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-
    sha2-256,umac-128@openssh.com" /etc/ssh/ssh_config.d/04-configure-SSH.conf
    ```

15. To check if the required `Ciphers` in the server side are enabled in order, run the following command:

    ```
    $ sudo sshd -T | grep -i "^ciphers" | grep "[[:space:]]chacha20-
    poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-
    ctr,aes192-ctr,aes128-ctr$"
    ```

16. If a result after running step 15 specifies that the required Ciphers in the server side are enabled in order, you can skip this step.

    Else, enable them in order in the `sshd_config` file, by running the following command:

    ```
    $ sudo grep -iq "^[ ]*Ciphers" /etc/ssh/sshd_config && sudo sed -i
    "s/^[ ]*Ciphers.*/Ciphers chacha20-poly1305@openssh.com,aes256-
    gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-
    ctr/i" /etc/ssh/sshd_config || sudo sed -i "$ a Ciphers chacha20-
    poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-
    ctr,aes192-ctr,aes128-ctr" /etc/ssh/sshd_config
    ```

17. To check if the required `Ciphers` in the client side are enabled in order, run the following command:

    ```
    $ sudo grep -i "^[[:space:]]*ciphers" /etc/ssh/ssh_config.d/04-
    configure-SSH.conf | grep "[[:space:]]chacha20-
    poly1305@openssh.com,aes256-gcm@openssh.com,aes128-
    gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr$"
    ```

18. If a result after running step 17 specifies that the required `Ciphers` in the client side are enabled in order, you can skip this step.

    Else, enable them in order in the `04-configure-SSH.conf` file.

    If the `04-configure-SSH.conf` file is empty (zero-byte file), run the following command:

    ```
    $ echo "Ciphers chacha20-poly1305@openssh.com,aes256-
    gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-
    ctr" | sudo tee -a /etc/ssh/ssh_config.d/04-configure-SSH.conf
    > /dev/null
    ```

    If the `04-configure-SSH.conf` file is non-empty, run the following command:

    ```
    $ sudo grep -iq "^[ ]*Ciphers" /etc/ssh/ssh_config.d/04-configure-
    SSH.conf && sudo sed -i "s/^[ ]*Ciphers.*/Ciphers chacha20-
    poly1305@openssh.com,aes256-gcm@openssh.com,aes128-
    gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr/i" /etc/ssh/
    ssh_config.d/04-configure-SSH.conf || sudo sed -i "$ a Ciphers
    chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-
    gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr" /etc/ssh/
    ssh_config.d/04-configure-SSH.conf
    ```

19. To check-in the `/etc/sysconfig/sshd, sshd_config`, and `04-configure-SSH.conf` files, run the following commands:

    ```
    $ sudo rcstool ci /etc/sysconfig/sshd
    ```

    ```
    $ sudo rcstool ci /etc/ssh/sshd_config
    ```

    ```
    $ sudo rcstool ci /etc/ssh/ssh_config.d/04-configure-SSH.conf
    ```

20. To restart `sshd` service, run the following command:

    ```
    $ sudo service sshd restart
    ```

3.1.2.12 Services Hardening Procedures

This section describes various hardening procedures for the services.

3.1.2.12.1 Uninstalling tftp-server Package

This section describes the procedure to uninstall the `tftp-server` package.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server:

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to remove the `tftp-server` package:

   ```
   $ sudo yum erase tftp-server
   ```

3.1.2.12.2 Disabling xinetd Service

This section describes the procedure to disable the xinetd service.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server:

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following commands to disable the xinetd service for all run levels and to stop the xinetd, if its currently running:

   ```
   $ sudo yum erase tftp-server
   $ sudo /sbin/service xinetd stop
   ```

   > **Note:**
   >
   > This step might fail if the xinetd service is already disabled or stopped.

3.1.2.12.3 Uninstalling xinetd Service

This section describes the procedure to uninstall xinetd service.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server:

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following commands to disable the xinetd service for all run levels and to stop the xinetd, if its currently running:

   ```
   $ sudo yum erase xinetd
   ```

3.1.2.12.4 Disabling ntpdate Service

This section describes the procedure to disable the ntpdate service.

Run the following procedure for each and every server in the topology:

1. Log in as `admusr` on the server:

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following commands to disable the ntpdate service:

   ```
   $ sudo chkconfig ntpdate off
   ```

3.2 SNMP Configuration

The DSR GUI has an interface to retrieve KPIs and alarms from a remote location using Simple Network Management Protocol (SNMP). Only the active Network OAM&P server allows SNMP administration. For more information, see the *SNMP Trapping* section in the *Operation, Administration, and Maintenance (OAM)* Guide.

The Active Network OAM&P server provides a single interface to SNMP data for the entire network, and individual servers interface directly with SNMP managers. The application sends SNMP traps to SNMP Managers that are registered to receive the traps. You can view and change the IP addresses and authorization information from the SNMP Trapping page.

You must set up at least one Manager to enable the SNMP. The system allows configuring up to five different Managers to receive SNMP traps and send requests. These could be either a valid IPv4 address or a valid hostname known to the system. The hostname must be unique and is case-insensitive. You can enter up to 20 characters into the string. The valid characters are alphanumeric and the minus sign. The hostname must start and end with an alphanumeric.

The **Enabled Versions** field on this page lets you pick the version of SNMP. The traps can be enabled or disabled collectively or independently from individual servers by checking the traps enabled checkbox on this page.

The SNMP Trapping page provides the following functionalities:

- Add an SNMP manager
- View SNMP settings
- Update SNMP settings
- Delete the SNMP manager

For more information on how to perform these actions, see the *Operation, Administration, and Maintenance (OAM)* Guide.

3.2.1 Enabling SNMP Versions

The Enabled Versions field in the SNMP Trapping page lets the user to enable the required SNMP version as follows:

- SNMPv2c: Allows SNMP service only to managers with SNMPv2c authentication.
- SNMPv3: Allows SNMP service only to managers with SNMPv3 authentication.
- SNMPv2c and SNMPv3: Allows SNMP service to managers with SNMPv2c or SNMPv3 authentication. This is the default option.

> **Note:**
>
> The recommended option is SNMPv3 for secure operation.

3.2.2 Configuring Community Names

When the `SNMPv2c` is enabled in the **Enabled Versions** field, you must configure the `SNMPV2c Community Name` since it is a required field. The maximum length of the Community Name (String) is 31 characters. It is recommended to use unique, hard to guess Community Name values and avoid using well-known Community Names such as “public” and “private.”

3.3 SNMPv3 on PMAC

This section provides an overview of procedures and sub-procedures required to enable overall SNMPv3 protocol support on the PMAC system. It also provides an overview of the procedure to configure SNMP Version 3 security model and trap servers.

3.3.1 Enabling SNMPv3 Support on PMAC

There are multiple procedures and sub-procedures required to enable overall SNMPv3 protocol support on the PMAC system as follows:

- Updating the SNMP service on existing remote servers on the PMAC control network.
- Updating the SNMP service on the PMAC server service to support SNMPv3.
- Updating the PMAC messaging system to support SNMPv3.
- Updating the SNMPv3 Security settings.

For more information about performing the above steps, see *PMAC Configuration Guide*.

3.3.2 Configuring SNMPv3 Security Model and Trap Servers

The SNMPv3 security model supports only HP 6125G/XLG and Cisco 4948E/E-F switches. For more information about configuring the SNMPv3 security model and trap servers, see *Procedure 18* and *Procedure 19* in the *PMAC Configuration Guide*.

3.4 Authorized IPs

IP addresses that have permission to access the GUI can be added or deleted from the Authorized IPs page. If an IP address does not have permission to access the GUI and attempts to connect, a notification displays on the GUI, and access is not granted to that IP address.

Before enabling this feature, you must add the IP address of the client to the list of authorized IPs. Enabling the Authorized IPs functionality prevents unauthorized IP addresses from accessing the DSR GUI.

For more information about how to enable this feature, see the *Authorized IPs* section in the *Operation, Administration, and Maintenance (OAM) Guide*.

3.5 Certificate Management

The Certificate Management feature allows you to configure digital security certificates to secure the following:

- Diameter Signaling Router (DSR) web sessions
- user authentication through secure LDAP over TLS
- Single Sign-On (SSO) authentication across a defined zone of DSR servers.

The feature functionalities are as follows:

- supports certificates based on the hostname or fully qualified hostname.
- allows building certificate signing requests (CSRs) for signing by a known certificate authority and later import the signed certificate into the DSR.
- allows generating a Certificate Report of individual or all (wildcard) defined certificates.

For more information about the Certificate Management feature, see the *Operation, Administration, and Maintenance (OAM) Guide*.

3.5.1 Creating a New Certificate for WebLogic and Tomcat Servers

This section describes the procedures that allow you to create customized certificates and replace the default Appworks certificate provided by DSR.

3.5.1.1 Creating Keystore and Certificate Signing Request

This procedure describes the steps to create a keystore and Certificate Signing Request (CSR).

1. Log in to the application VM of IDIH using SSH.
2. Run the following command to change the user to tekelec:

   ```
   sudo su - tekelec
   ```

3. Run the following command to change the directory to the Weblogic domain (nsp):

   ```
   cd /usr/TKLC/xIH/bea/user_projects/domains/tekelec/nsp
   ```

4. Run the following commands to take a backup of the existing key and trust stores:

   ```
   cp idih.jks idih_bkp.jks
   cp idih-trust.jks idih-trust-bkp.jks
   ```

5. Run the following command to create a keystore and a private key using the genkeypair or genkey command:

   ```
   keytool -genkeypair -alias <alias_name> -keyalg RSA -keysize 1024 -dname
   "CN=<ServerName>, OU=GTI, O=<CompanyName>, L=<City>,
   ST=<State>,C=<Country> " -keypass <key_password> -keystore
   <server_keystore>.jks -storepass <store_password>
   ```

   Where,

   - `<alias_name>` indicates the alias for the keystore.
   - `<ServerName>` indicates the server name.
   - `<CompanyName>` indicates your company name.
   - `<City>` indicates your city name.
   - `<State>` indicates your state name.
   - `<Country>` indicates your country name.
   - `<key_password>` indicates the password.
   - `<server_keystore>` indicates keystore name.
   - `<store_password>` indicates the store password.

   In the above command, Common Name (CN) can be a domain name/DNS Name/ machine name or any other name. The CN must match your machine name or hostname. This allows the hostname verification to complete.

   The system generates a private and public key pair.

6. To create a Certificate Signing Request (CSR), run the following command:

   ```
   keytool -certreq -v -alias <alias_name> -file <csr-for-myserver>.pem -
   keypass <key_password> -storepass <store_password> -keystore
   <server_keystore>.jks
   ```

   Where,

   - `<alias_name>` indicates the alias that was used during the creation of keystore.
   - `<csr-for-myserver>` indicates a file name for the CSR file.
   - `<key_password>` indicates the keystore password that was provided during the keystore creation.
   - `<store_password>` indicates the store password that was provided during the keystore creation.
   - `<server_keystore>` indicates the JKS file name that was generated during the keystore creation.

   The system creates the `csr-for-myserver.pem` file. The file is sent to a Certificate Authority (CA) to create a signed public key certificate.

3.5.1.2 Importing Certificate

This procedure describes the steps to import the certificate.

1. When the CA returns the signed public key with the intermediate and root certificates, run the following command to import the intermediate and root certificates into your Keystore:

   ```
   keytool -importcert -v -noprompt -trustcacerts -alias
   <alias_for_root_certificate> -file <root_certificate_file> -
   keystore <server_keystore>.jks -storepass <store_password>
   ```

   Where,

   - - `<alias_for_root_certificate>` indicates an alias for the root certificate.
     - `root_certificate_file` indicates the file name of the root certificate issued by CA.
     - `server_keystore` indicates the JKS file name that was generated during the Keystore creation.
     - `store_password` indicates the store password that was provided during the Keystore creation.
2. Import the public certificate into the Keystore using the private key alias.
3. To obtain the certificate:
   - From the CA's website, download the root CA and intermediate CA if available.
   - Double-click the certificate file, and then go to the **Certification Path** tab.
   - The first certificate in the list is the root CA and the second one is the intermediate CA if available. If you highlight the root CA, and then click **View Certificate**, it opens the Root CA certificate. Then, you can go to the **Details** tab and click <Copy to file>. Select Base 64 as the format and save the file. Repeat the same steps to copy the intermediate CA to a file.
4. When you obtain root CA, intermediate, and certificate files, if you have an intermediate CA, edit it and copy all the content.
5. Edit the certificate file and paste the intermediate at the bottom of the server certificate. Skip this step if you do not have an intermediate CA.
6. Repeat the same step for the root CA and paste it at the end of the previously added certificate.

   The following is a sample certificate:

   ```
   -------BEGIN CERTIFICATE---------
   dfsfsdfdf
   sfsdfwehdfhdf <---------certificate
   dgdfgfgfdg
   ---------END CERTIFICATE-----------
   -------BEGIN CERTIFICATE---------
   hghjgfjgj
   sfsdfwejjhdfhdf <---------intermediate
   dgdfgiuiyuiuiyufgfdg
   ---------END CERTIFICATE-----------
   -------BEGIN CERTIFICATE---------
   dfsfsmbvmvbmdfdf
   sfsdetetrtyrfwehdfhdf <---------root CA
   ```

```
dgdfgnbnbvnvbfgfdg
--------END CERTIFICATE-----------
```

7. Run the following command to import the certificate:

   ```
   keytool -importcert -v -alias <alias_name> -file <mycert> -keystore
   <server_keystore>.jks -keypass <key_password> -storepass <store_password>
   ```

   Where,

   - `<alias_name>` indicates the alias that was used during the creation of Keystore.
   - `<mycert>` indicates the file name of the certificate issued by CA.
   - `<server_keystore>` indicates the JKS file name that was generated during the Keystore creation.
   - `<key_password>` indicates the Keystore password that was provided during the Keystore creation.
   - `<store_password>` indicates the store password that was provided during the Keystore creation.

8. Run the following command to check whether the Keystore creation is complete:

   ```
   keytool -list -v -keystore <server_keystore>.jks -storepass
   <store_password>
   ```

9. Run the following command to import the root CA of your signed certificate to the Trust KeyStore file:

   ```
   keytool -alias server_cert -import -file rootcacert.cer -keystore
   trustkeystore.jks -storepass <Password>
   ```

3.5.1.3 Configuring Keystore on WebLogic

This procedure describes the steps to configure Keystore on WebLogic.

1. Log in to the WebLogic Server Administration Console using your login credentials.
2. In the left navigation pane, click **Environment** > **Servers**.
3. In the **Customize this table** section, in the **Name** column, click **nsp(admin)**.

   nsp(admin) is the server for which the identity and trust keystores configuration is performed.
4. In the **Settings for nsp** section, click **Configuration** > **Keystores**.
5. To edit or modify the existing settings of the Keystore configuration, in the left navigation pane, click **Lock & Edit**.
6. In the **Keystores** section, edit the following fields as required:
   - **Custom Identity Keystore**: Enter the fully qualified path to the identity Keystore.
   - **Custom Identity Keystore Type**: Enter the type of Keystore.
     This attribute is a Java KeyStore (JKS). The default value is JKS.
   - **Custom Identity Keystore Passphrase**: Enter the password required for reading or writing to the Keystore, for example, weblogic1234.

- **Custom Trust Keystore**: Enter the fully qualified path to the trust Keystore.
- **Custom Trust Keystore Passphrase**: Enter the passphrase of the custom trust Keystore.
- **Confirm Custom Trust Keystore Passphrase**: Re-enter the passphrase of the custom trust Keystore.

7. Click **Save**.
8. In the **Settings for nsp** section, click **Configuration** > **SSL**.
9. In the **Identity** section, edit the following fields as required:
   - **Private Key Alias**: Enter the fully qualified path to the identity Keystore.
   - **Private Key Passphrase**: Enter the same password used for the creation of Keystore, for example, weblogic1234.
   - **Confirm Private Key Passphrase**: Re-enter the same password used in the **Private Key Passphrase** field**.**
10. Click **Save**.
11. In the left navigation pane, click **Activate Changes**.
12. Restart WebLogic by logging in to app server using admusr, and then run the following command:

    ```
    sudo service xih-apps restart
    ```

3.5.1.4 Creating Keystore in the Tomcat Server

This procedure describes the steps to create Keystore in the Tomcat server.

1. Log in to the IDIH App VM using SSH as an admusr user.
2. Run the following command to change the directory to conf folder of Tomcat:

   ```
   cd /usr/share/tomcat6/conf
   ```

3. Run the the following command to take a backup of the existing jks file:

   ```
   cp idih.jks idih-bkp.jks
   ```

4. Run the following command to copy the Keystore that was created for the WebLogic server into the Tomcat configuration folder:

   ```
   cp /usr/TKLC/xIH/bea/user_projects/domains/tekelec/nsp/<JKS file
   created for WebLogic in the previous step> .
   sudo chown tomcat:root <JKS file created for WebLogic in the
   previous step>
   ```

3.5.1.5 Modifying the Tomcat File Configuration

This procedure describes the steps to modify the Tomcat file configuration.

1. Run the following command to edit the `server.xml` file and update `keystoreFile` and `keystorePass` fields:

   ```
   sudo vim server.xml
   ```

2. Modify the following tag in the server.xml file and ensure that the `keystoreFile` field is updated with the latest jks file name and `keystorePass` with its corresponding password.

   ```
   <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"

                  maxThreads="150" scheme="https" secure="true"

                  clientAuth="false" sslProtocol="TLS"

                  keystoreFile="conf/<JKS file created for WebLogic in the
                  previous step>.jks"

                  keystorePass="<Password used during the creation of
                  keystore>" />
   ```

3. Run the following command to restart the Tomcat server:

   ```
   sudo service tomcat6 restart
   ```

3.6 SFTP Administration

Oracle Communications Diameter Signaling Router (DSR) supports SFTP sessions with external servers to transfer various files from DSR. The authentication process requires a digital certificate to authenticate the sessions. The external server drives the files transfer process.

For more information, see the *SFTP Users Administration* section in the *Operation, Administration, and Maintenance (OAM) Guide*.

4

Host Intrusion Detection System (HIDS)

This chapter describes the Host Intrusion Detection System (HIDS) security feature available to the Platform Administrator through the Linux Command Line Interface (CLI). The **platcfg** utility of the Operating System (OS) is used for configuring this feature.

4.1 Overview

The Host Intrusion Detection System (HIDS) feature monitors a server for malicious activity by periodically examining file system changes, logs, and auditing processes. The HIDS feature monitors TPD and TVOE log files and ensures that HIDS and Syscheck processes are running.

The HIDS monitoring feature monitors the following protected log files:

- All files in `/var/TKLC/log/hids`
- `/var/log/messages`
- `/var/log/secure`
- `/var/log/cron`

The log files created are as follows:

- **alarms.log** – Any HIDS functionality resulting in an alarm being raised or cleared is logged in this file, for example, file tampering alarm, Syscheck process alarm, Samhain process alarm.
- **admin.log** – The output of any HIDS command executed resulting in a success or error is logged in this file. This file also includes logs on attempts to run commands as a non HIDS administrator.
- **hids.log** – Logs any other information such as state changes and when Samhain runs but finds no file tampering errors.

No other system resources such as files, processes, actions, and so on are monitored by HIDS.

HIDS alarms are standard TPD alarms with the `alarmEventType` set to `securityServiceOrMechanismViolation`. The HIDS alarms are propagated through normal COMCOL channels, resulting in SNMP traps being sent to the client's SNMP management system if configured. The active alarms can be viewed in the platcfg GUI. You can also view the active alarms on the Diameter Signaling Router (DSR) GUI by navigating to **Alarms & Events** > **View Active**.

4.2 Checking the Host Intrusion Detection System Status

This section describes the procedure to check the status of Host Intrusion Detection System (HIDS).

Perform the following steps to determine the HIDS status:

1. Log in as `admusr` on the server.

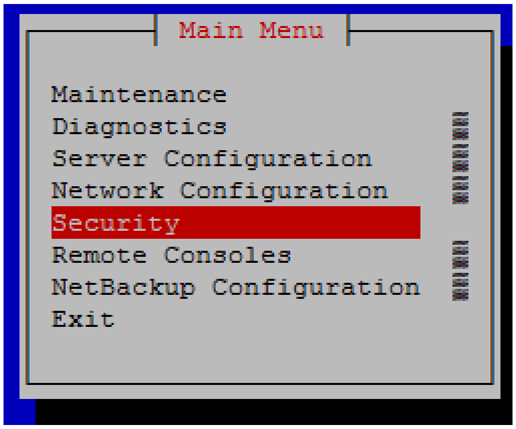
   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to open the platcfg menu:

   ```
   $ sudo su - platcfg
   ```

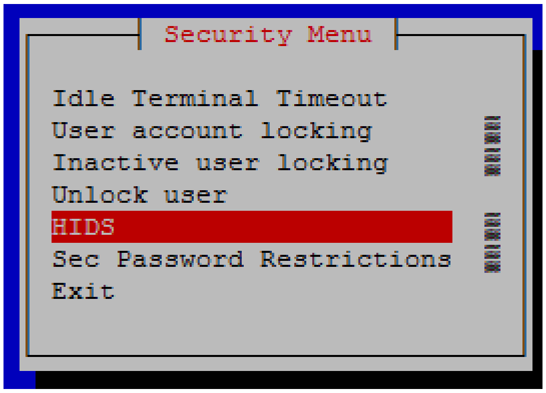
3. Select **Security** from the Main Menu and press **Enter**.

Figure 4-1 Main Menu



4. Select **HIDS** from the menu and press **Enter**.

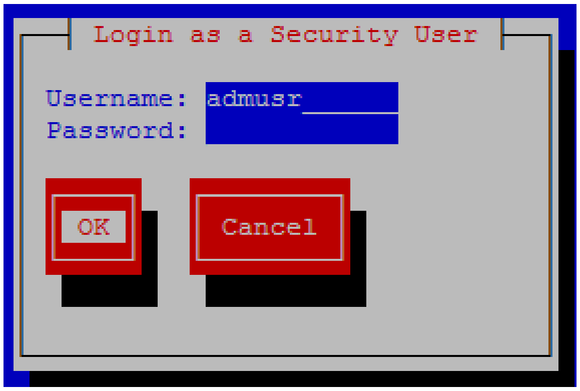
Figure 4-2 Security Menu



5. To check the HIDS status, perform the following:

   a. Type the **Username** and **Password** for a user that is part of the **secgrp** group.

Figure 4-3   Security User Login



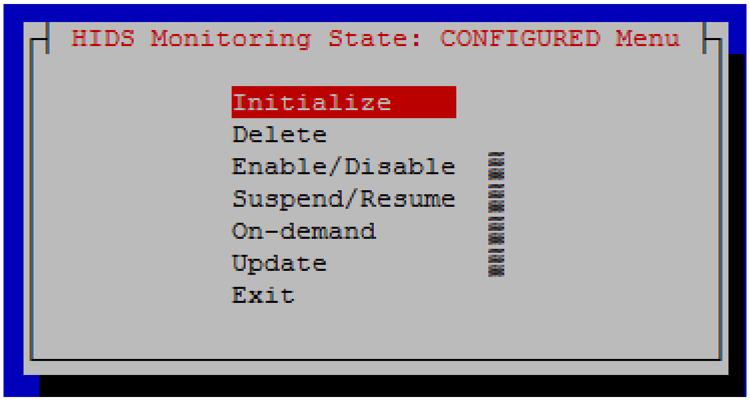
```
Username: admusr
Password:
```

> **Note:**
>
> By default, admusr is part of the secgrp group.

b. Click **OK** and press **Enter**.

   The HIDS menu displays and the HIDS Monitoring State is listed on the top of the window.

Figure 4-4   HIDS Monitoring State



6. Select **Exit** in each of the menus until a command prompt is reached.

4.3 Initializing the Host Intrusion Detection System

This section describes the procedure to initialize the Host Intrusion Detection System (HIDS).

Perform the following steps to initialize HIDS:

1. Log in as `admusr` on the server.

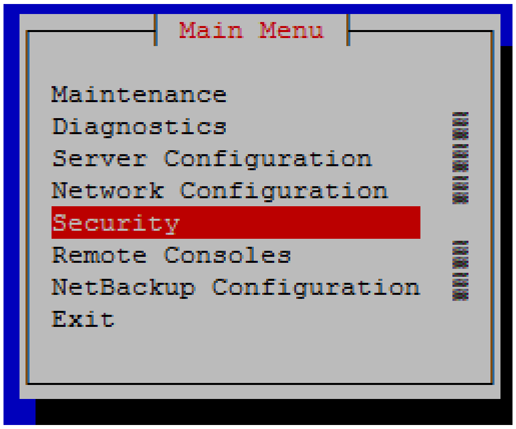
   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to open the platcfg menu:

```
$ sudo su - platcfg
```

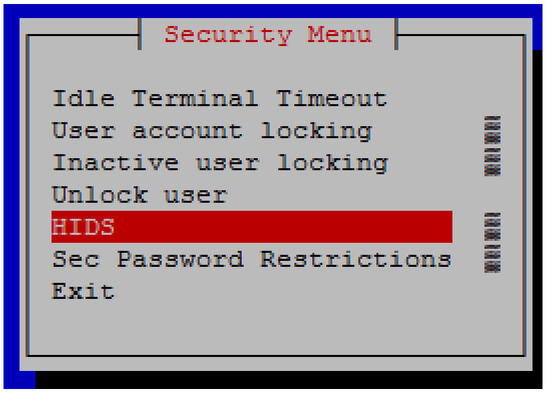
3. Select **Security** from the Main Menu and press **Enter**.

Figure 4-5 Main Menu



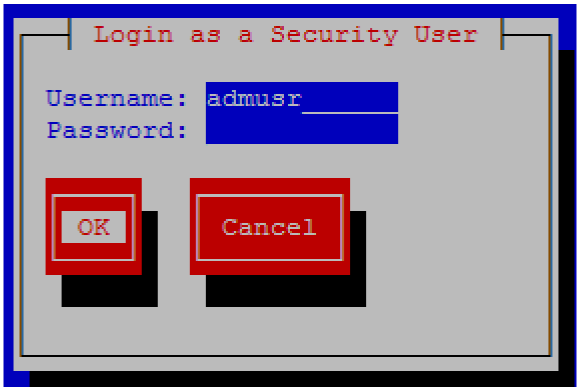
4. Select **HIDS** from the menu and press **Enter**.

Figure 4-6 Security Menu



5. To check the HIDS status, perform the following:
   a. Type the **Username** and **Password** for a user that is part of the **secgrp** group.

Figure 4-7 Security User Login



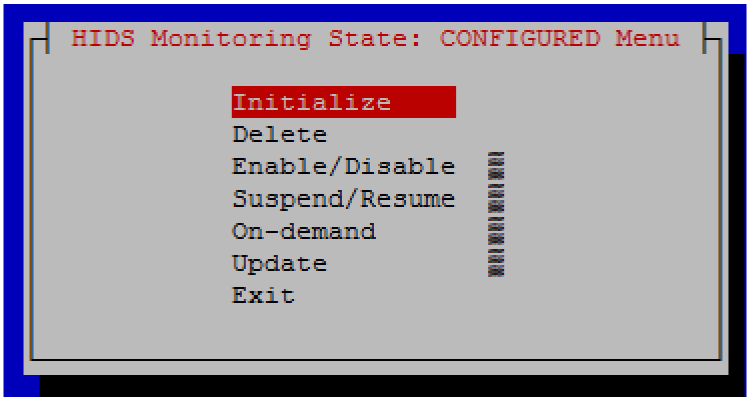
> **Note:**
>
> By default, admusr is part of the secgrp group.

b. Click **OK** and press **Enter**.

6. To initialize HIDS, perform the following:

a. Select **Initialize** and press **Enter**.

Figure 4-8 Initialize HIDS



b. Select **Yes** and press **Enter**.

c. After the HIDS baseline successfully initialized message displays, press any key to continue.

7. Select **Exit** in each of the menus until a command prompt is reached.

4.4 Enabling or Disabling Host Intrusion Detection System

This section describes the procedure to enable or disable Host Intrusion Detection System (HIDS).

The Host Intrusion Detection System (HIDS) feature must be initialized before enabling HIDS for the first time on a system.

Perform the following steps to enable or disable HIDS:

1. Log in as `admusr` on the server.

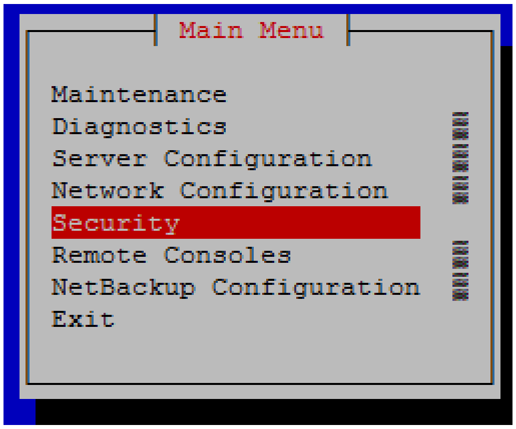
   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to open the platcfg menu:

   ```
   $ sudo su - platcfg
   ```

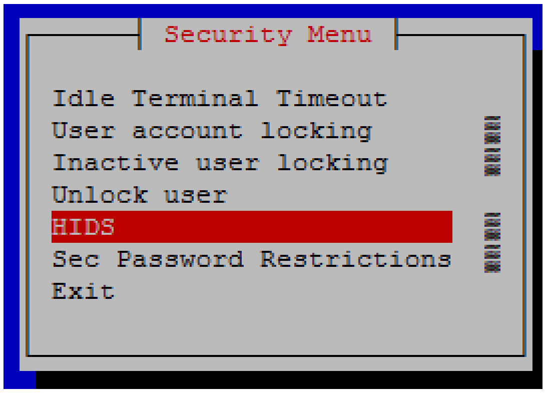
3. Select **Security** from the Main Menu and press **Enter**.

Figure 4-9   Main Menu



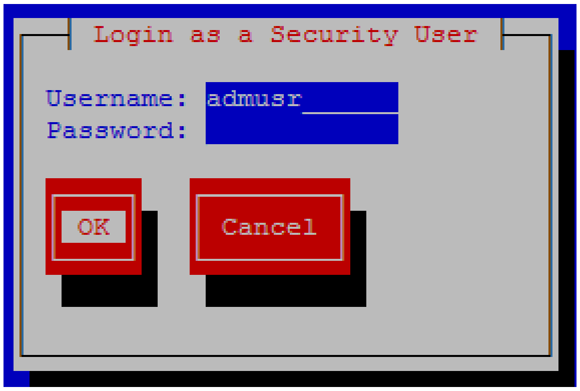
4. Select **HIDS** from the menu and press **Enter**.

Figure 4-10   Security Menu



5. To check the HIDS status, perform the following:
   a. Type the **Username** and **Password** for a user that is part of the **secgrp** group.

Figure 4-11 Security User Login



Login as a Security User

Username: admusr
Password:

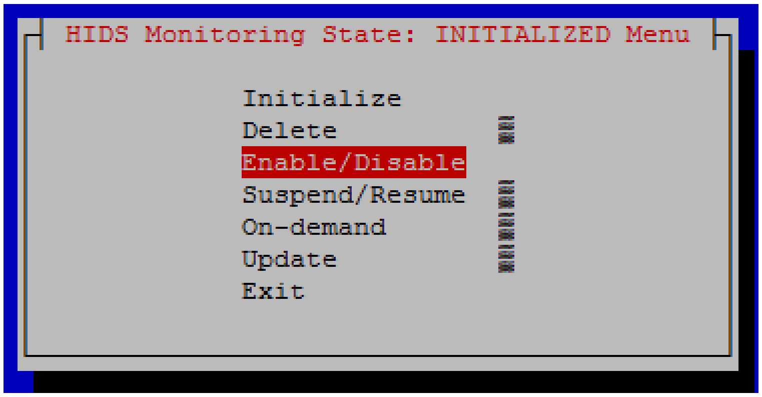
> **Note:**
>
> By default, admusr is part of the secgrp group.

b. Click **OK** and press **Enter**.

6. To enable or disable HIDS, perform the following:

   a. Select **Enable/Disable** and press **Enter**.

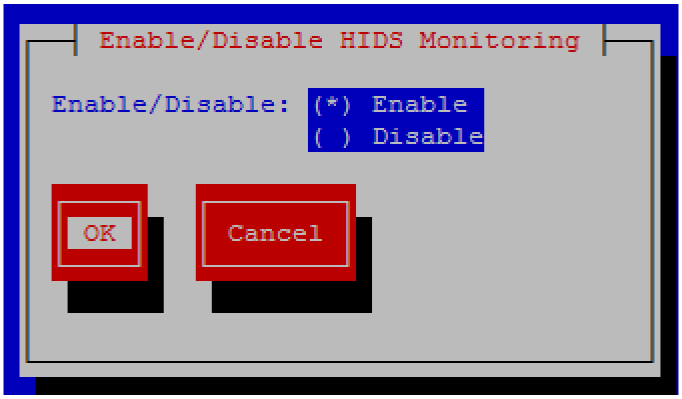
Figure 4-12 Enable or Disable Menu



HIDS Monitoring State: INITIALIZED Menu

b. Select either **Enable** or **Disable** option.

Figure 4-13 Enable or Disable HIDS



```
Enable/Disable HIDS Monitoring

Enable/Disable: (*) Enable
                ( ) Disable
```

c. Click **OK** and press **Enter**.

   A message box indicating that DB monitoring has been enabled or disabled or a failure message appears.

d. Press any key to continue.

7. Select **Exit** in each of the menus until a command prompt is reached.

4.5 Suspending or Resuming Host Intrusion Detection System

This section describes the procedure to temporarily suspend or resume Host Intrusion Detection System (HIDS) monitoring on a system that has HIDS enabled.

Perform the following steps to suspend or resume HIDS:

1. Log in as `admusr` on the server.

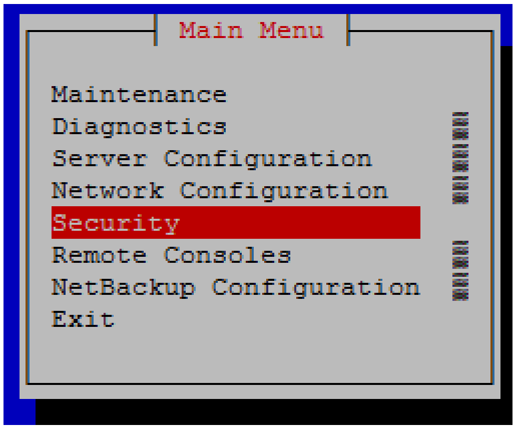
   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to open the platcfg menu:

   ```
   $ sudo su - platcfg
   ```

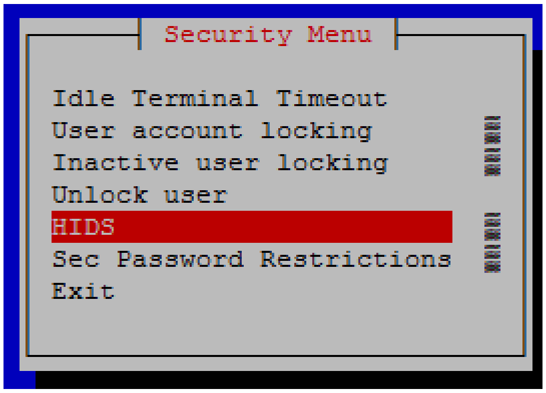
3. Select **Security** from the Main Menu and press **Enter**.

Figure 4-14 Main Menu



4. Select **HIDS** from the menu and press **Enter**.

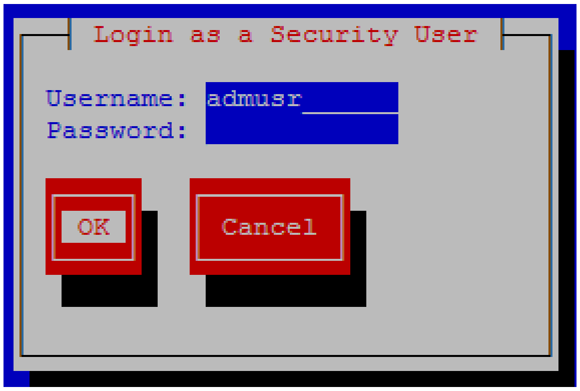
Figure 4-15 Security Menu



5. To check the HIDS status, perform the following:

   a. Type the **Username** and **Password** for a user that is part of the **secgrp** group.

Figure 4-16 Security User Login



```
Username: admusr
Password:
```

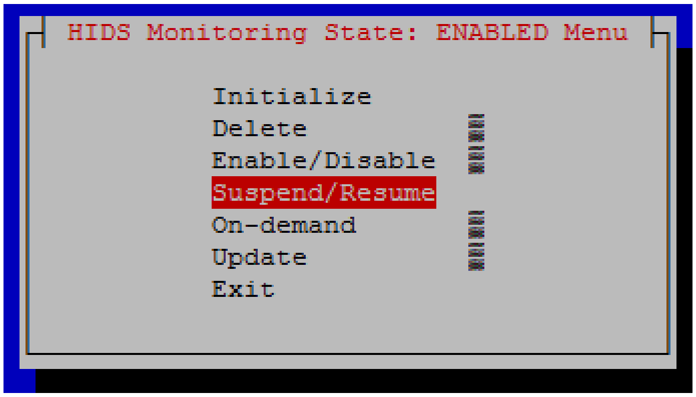
> **Note:**
>
> By default, admusr is part of the **secgrp** group.

b. Click **OK** and press **Enter**.

6. To suspend or resume HIDS, perform the following:

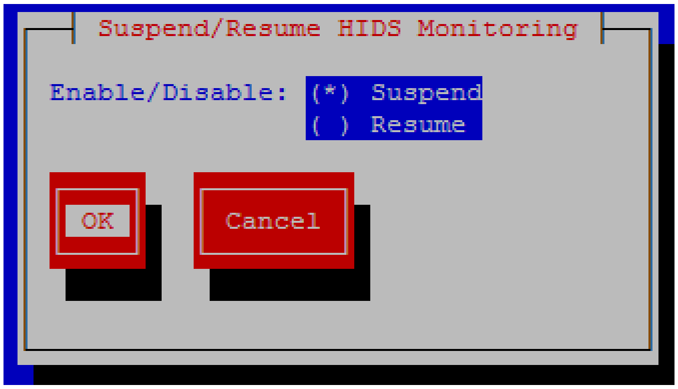
   a. Select **Supend/Resume** and press **Enter**.

Figure 4-17 Suspend or Resume in Enabled Menu



   b. Select either **Suspend** or **Resume** option.

Figure 4-18 Suspend or Resume HIDS



   c. Click **OK** and press **Enter**.

      A message box indicating that DB monitoring has been enabled or disabled or a failure message appears.

   d. Press any key to continue.

7. Select **Exit** in each of the menus until a command prompt is reached.

4.6 Running On-Demand HIDS Security Check

The HIDS tests run periodically. This section describes the procedure to force an immediate run of the HIDS tests by using the On-demand HIDS menu.

Perform the following steps to run on-demand HIDS tests:

1. Log in as `admusr` on the server.

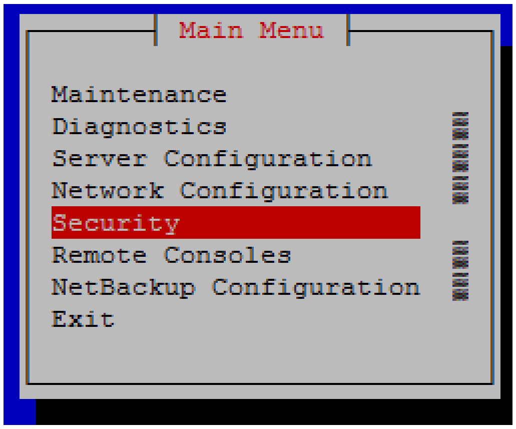
   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to open the platcfg menu:

   ```
   $ sudo su - platcfg
   ```

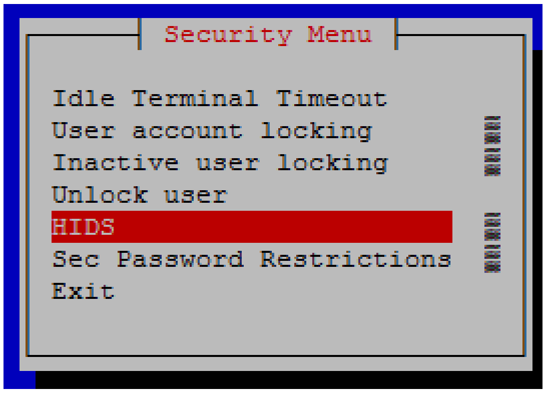
3. Select **Security** from the Main Menu and press **Enter**.

   Figure 4-19   Main Menu

4. Select **HIDS** from the menu and press **Enter**.

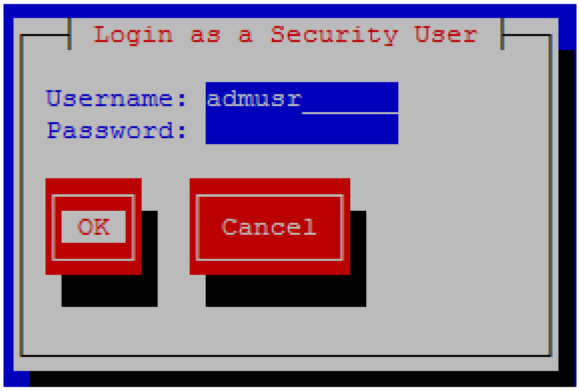
   Figure 4-20   Security Menu

5. To check the HIDS status, perform the following:

a. Type the **Username** and **Password** for a user that is part of the **secgrp** group.

Figure 4-21 Security User Login



```
Username: admusr
Password:
```

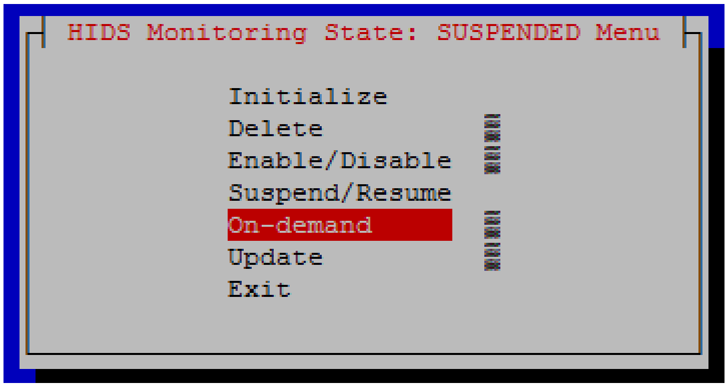
> **Note:**
>
> By default, admusr is part of the **secgrp** group.

b. Click **OK** and press **Enter**.

6. To select on-demand HIDS testing, perform the following:

    a. Select **On-demand** and press **Enter**.

Figure 4-22 On-Demand HIDS Test

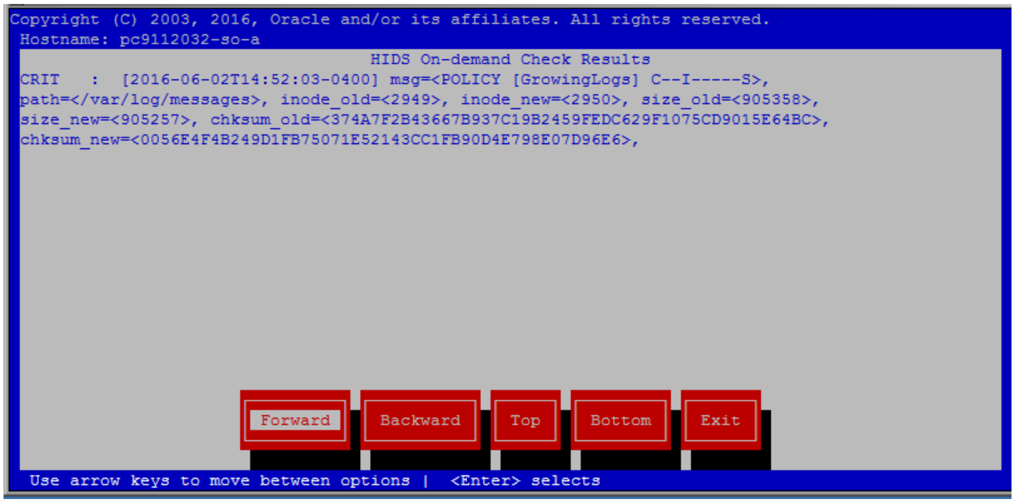


b. Click **OK** and press **Enter**.

A message box indicating the success or fail result appears.

c. Press any key to continue. If an error exists, a screen similar to the following screen displays:

Figure 4-23   HIDS On-Demand Check Results



```
Copyright (C) 2003, 2016, Oracle and/or its affiliates. All rights reserved.
 Hostname: pc9112032-so-a
                                   HIDS On-demand Check Results
CRIT   :  [2016-06-02T14:52:03-0400] msg=<POLICY [GrowingLogs] C--I-----S>,
path=</var/log/messages>, inode_old=<2949>, inode_new=<2950>, size_old=<905358>,
size_new=<905257>, chksum_old=<374A7F2B43667B937C19B2459FEDC629F1075CD9015E64BC>,
chksum_new=<0056E4F4B249D1FB75071E52143CC1FB90D4E798E07D96E6>,
```

```
  Use arrow keys to move between options |  <Enter> selects
```

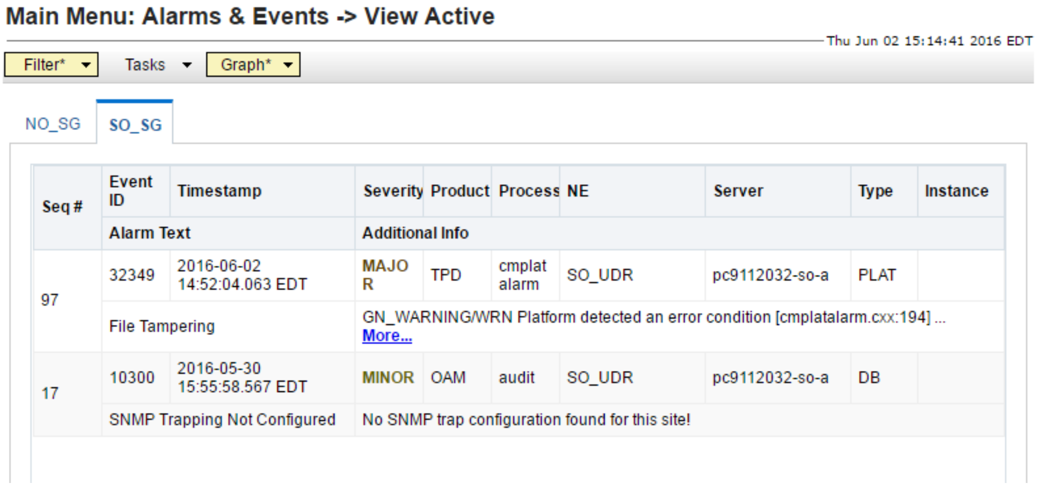
This alarm can also be seen when viewing alarms in the platcfg system. For more information, see the Host Intrusion Detection System Alarms section.

This alarm is also propagated through normal COMCOL channels ultimately resulting in the alarm being accessible on the Diameter Signaling Router (DSR) GUI by navigating to **Alarm & Events** > **View Active**, as shown in Step 8.

7. Select **Exit** in each of the menus until a command prompt is reached.
8. This is an optional step. To view HIDS error, Log into the DSR GUI and navigate to **Alarms & Events** > **View Active**.

   Examples of screens from the current error are as follows:

Figure 4-24   Alarms and Events Menu

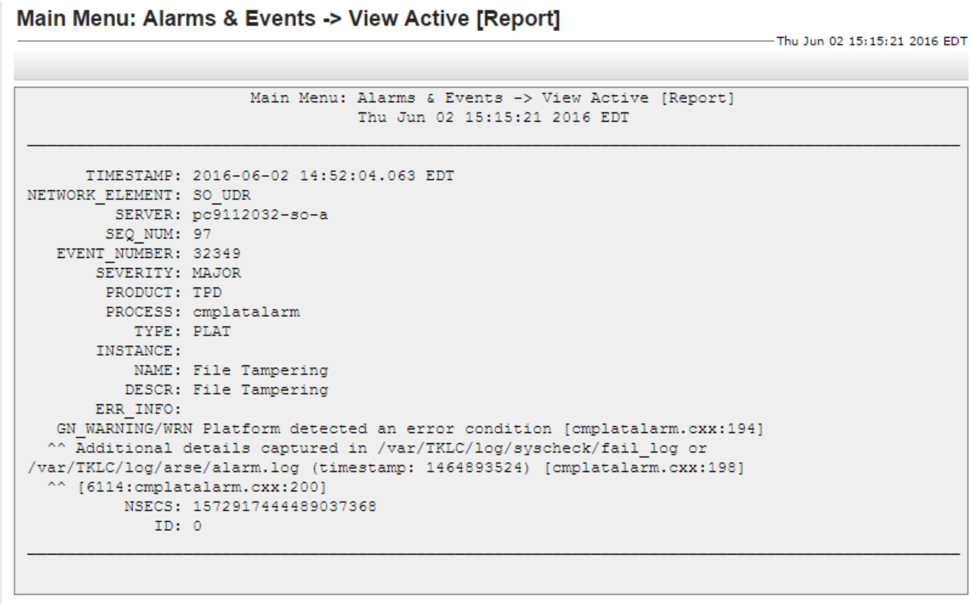


Main Menu: Alarms & Events -> View Active

Thu Jun 02 15:14:41 2016 EDT

Seq #	Event ID	Timestamp	Severity	Product	Process	NE	Server	Type	Instance
	Alarm Text		Additional Info						
97	32349	2016-06-02 14:52:04.063 EDT	MAJOR	TPD	cmplat alarm	SO_UDR	pc9112032-so-a	PLAT	
	File Tampering		GN_WARNING/WRN Platform detected an error condition [cmplatalarm.cxx:194] ... More...						
17	10300	2016-05-30 15:55:58.567 EDT	MINOR	OAM	audit	SO_UDR	pc9112032-so-a	DB	
	SNMP Trapping Not Configured		No SNMP trap configuration found for this site!						

Figure 4-25 View Report



Main Menu: Alarms & Events -> View Active [Report]

Thu Jun 02 15:15:21 2016 EDT

```
                    Main Menu: Alarms & Events -> View Active [Report]
                              Thu Jun 02 15:15:21 2016 EDT
_______________________________________________________________________________________________

      TIMESTAMP: 2016-06-02 14:52:04.063 EDT
NETWORK_ELEMENT: SO_UDR
         SERVER: pc9112032-so-a
        SEQ_NUM: 97
   EVENT_NUMBER: 32349
       SEVERITY: MAJOR
        PRODUCT: TPD
        PROCESS: cmplatalarm
           TYPE: PLAT
       INSTANCE:
           NAME: File Tampering
          DESCR: File Tampering
       ERR_INFO:
   GN_WARNING/WRN Platform detected an error condition [cmplatalarm.cxx:194]
  ^^ Additional details captured in /var/TKLC/log/syscheck/fail_log or
/var/TKLC/log/arse/alarm.log (timestamp: 1464893524) [cmplatalarm.cxx:198]
  ^^ [6114:cmplatalarm.cxx:200]
          NSECS: 1572917444489037368
             ID: 0
_______________________________________________________________________________________________
```

4.7 Updating Host Intrusion Detection System Baseline

This section describes the procedure to update the checksums on all files or specific files in the HIDS baseline, which can clear HIDS alarms associated with the updated files.

Perform the following steps to update the checksums on all files or specific files in the HIDS baseline:

1. Log in as `admusr` on the server.

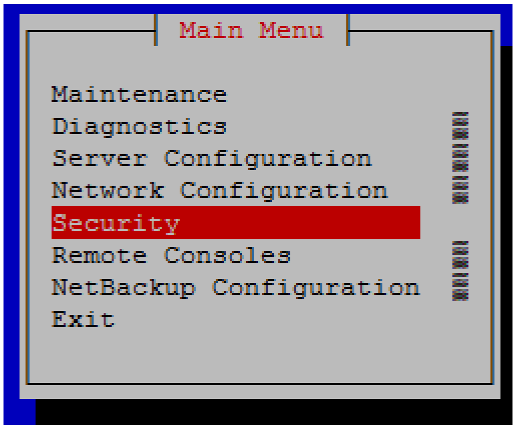
   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to open the platcfg menu:

   ```
   $ sudo su - platcfg
   ```

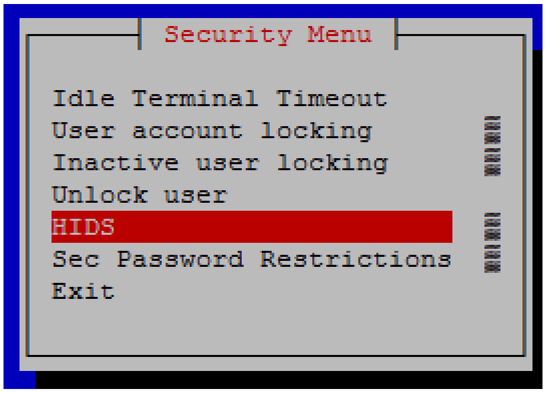
3. Select **Security** from the Main Menu and press **Enter**.

Figure 4-26   Main Menu



4. Select **HIDS** from the menu and press **Enter**.

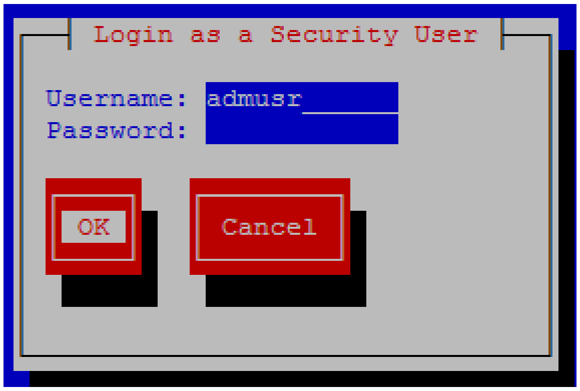
Figure 4-27   Security Menu



5. To check the HIDS status, perform the following:

   a. Type the **Username** and **Password** for a user that is part of the **secgrp** group.

Figure 4-28   Security User Login



```
Username: admusr
Password:
```

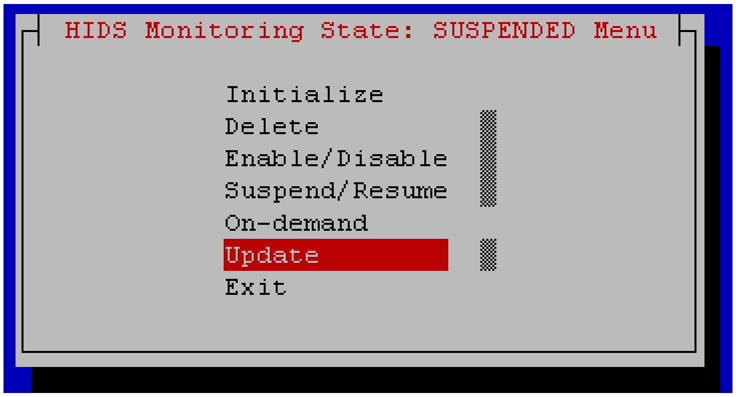
> **Note:**
>
> By default, admusr is part of the **secgrp** group.

b. Click **OK** and press **Enter**.

6. To update HIDS, perform the following:

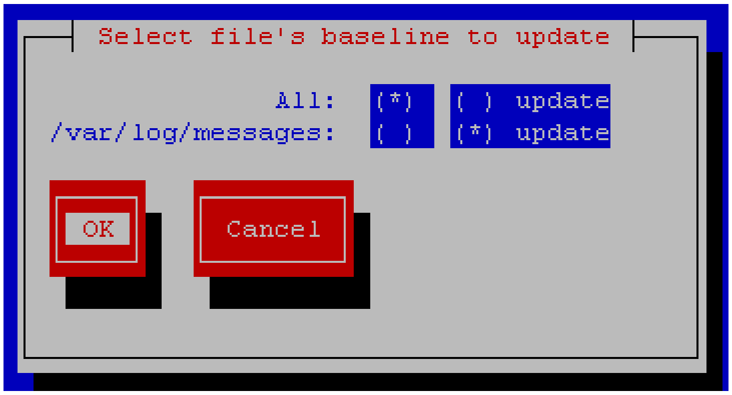
   a. Select **Update** and press **Enter**.

Figure 4-29 Update HIDS



   b. Select the file’s baseline to update.

Figure 4-30 Update File's Baseline



```
Select file's baseline to update

              All:  (*)  ( ) update
/var/log/messages:  ( )  (*) update
```

   c. Click **OK** and press **Enter**.

   A message box indicating the success or fail result appears.

   d. Press any key to continue.

7. Select **Exit** in each of the menus until a command prompt is reached.

4.8 Deleting Host Intrusion Detection System

This section describes the procedure to permanently disable HIDS or to back out from a product upgrade by using the HIDS Delete menu.

Perform the following steps to delete HIDS:

1. Log in as `admusr` on the server.

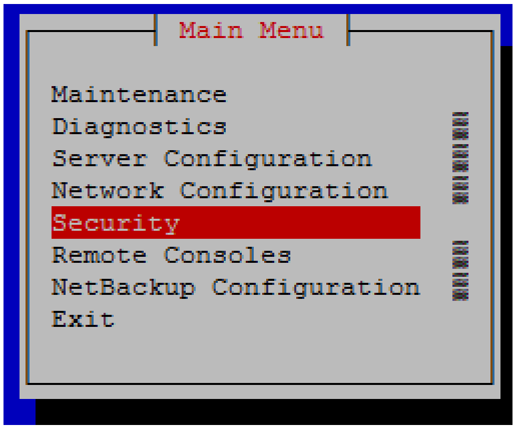
   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to open the platcfg menu:

   ```
   $ sudo su - platcfg
   ```

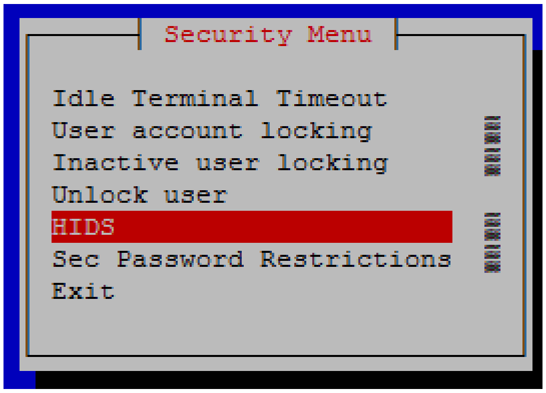
3. Select **Security** from the Main Menu and press **Enter**.

Figure 4-31 Main Menu



4. Select **HIDS** from the menu and press **Enter**.

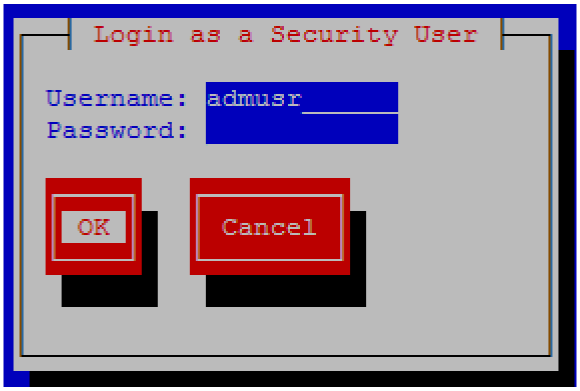
Figure 4-32 Security Menu



5. To check the HIDS status, perform the following:

a. Type the **Username** and **Password** for a user that is part of the **secgrp** group.

Figure 4-33 Security User Login



Username: admusr
Password:

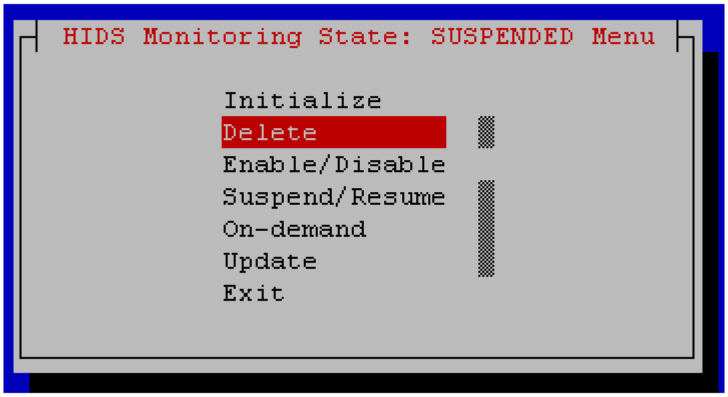
> **Note:**
>
> By default, admusr is part of the **secgrp** group.

b. Click **OK** and press **Enter**.

6. To delete HIDS, perform the following:

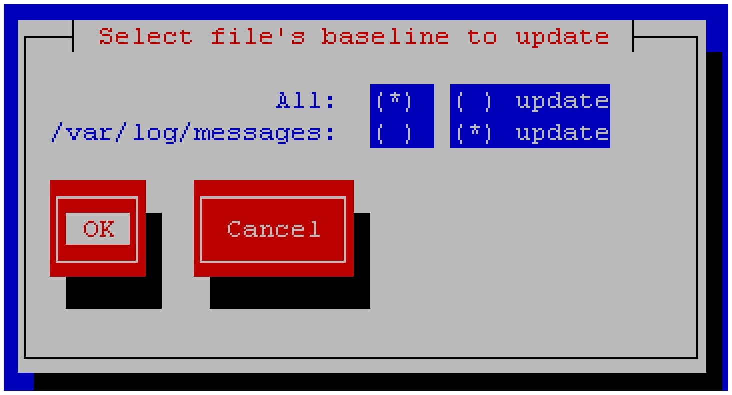
   a. Select **Delete** and press **Enter**.

Figure 4-34 Delete HIDS



   b. Select the file’s baseline to update.

Figure 4-35 Update File's Baseline



```
Select file's baseline to update

               All:  (*)  ( ) update
/var/log/messages:  ( )  (*) update
```

c. Click **OK** and press **Enter**.

   A message box indicating the success or fail result appears.

d. Press any key to continue.

7. Select **Exit** in each of the menus until a command prompt is reached.

4.9 Host Intrusion Detection System Alarms

This section describes the overview and procedure to view HIDS Alarms.

The HIDS alarms are standard TPD alarms with the `alarmEventType` set to `securityServiceOrMechanismViolation`.

The HIDS alarms are propagated through normal COMCOL channels that result in SNMP traps being sent to the client's SNMP management system if configured.

The multiple ways to view the alarms are as follows:

- You can view the current, previously cleared, and how alarms were cleared in the `/var/TKLC/logs/hids/alarms.log` file.
- You can view active alarms on the DSR GUI by navigating to **Main Menu > Alarms & Events > View Active**.
- You can view active alarms on the **platcfg** GUI, including HIDS alarms, by performing the following steps:

1. Log in as `admusr` on the server.

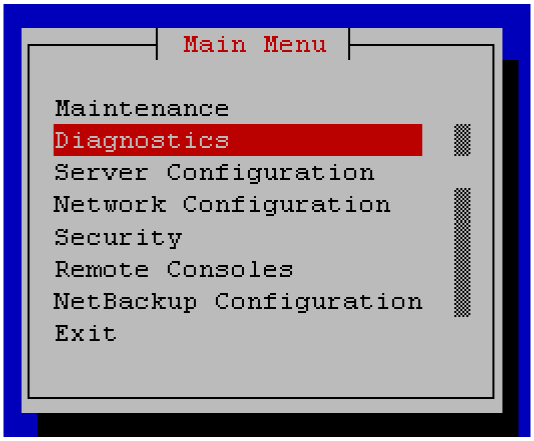
```
login: admusr
Password: <current admin user password>
```

2. Run the following command to open the platcfg menu:

```
$ sudo su - platcfg
```

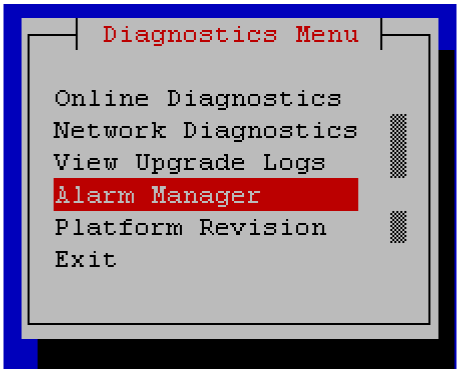
3. Select **Diagnostics** from the Main Menu and press **Enter**.

Figure 4-36 Diagnostics on Main Menu



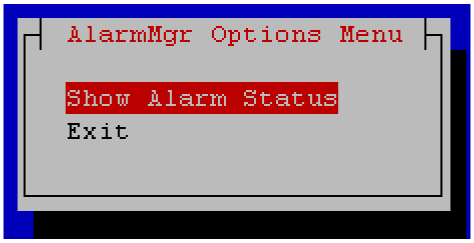
4. Select **Alarm Manager** from the menu and press **Enter**.

Figure 4-37 Alarm Manager



5. To view the alarm status, perform the following:
   a. Select **Show Alarm Status** from the menu and press **Enter**.

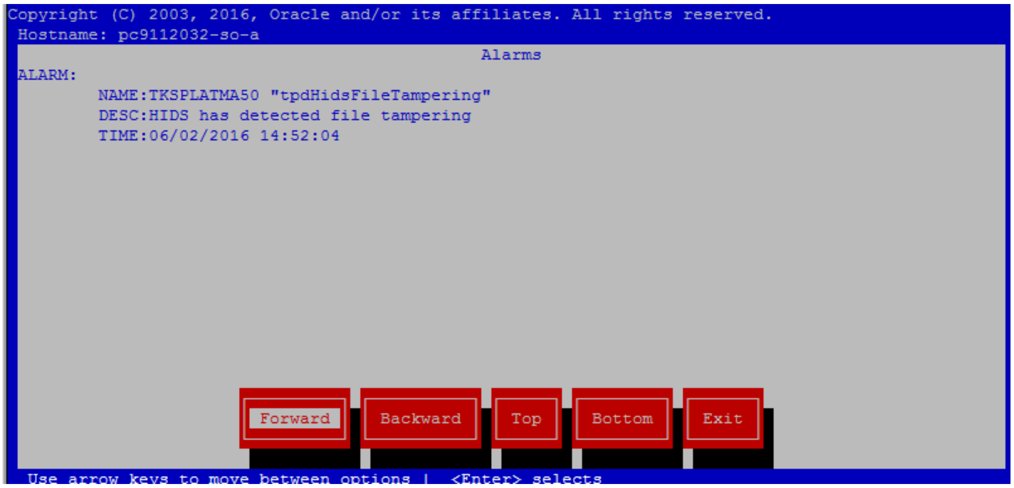
   Figure 4-38 Alarm Status

   A message box indicating the success or fail result appears.

   b. Press any key to continue. If an error exists, a screen similar to the following screen displays:

Figure 4-39   Alarm



```
Copyright (C) 2003, 2016, Oracle and/or its affiliates. All rights reserved.
 Hostname: pc9112032-so-a
                                              Alarms
 ALARM:
         NAME:TKSPLATMA50 "tpdHidsFileTampering"
         DESC:HIDS has detected file tampering
         TIME:06/02/2016 14:52:04
```

```
  Use arrow keys to move between options |  <Enter> selects
```

6. Select **Exit** in each of the menus until a command prompt is reached.

5

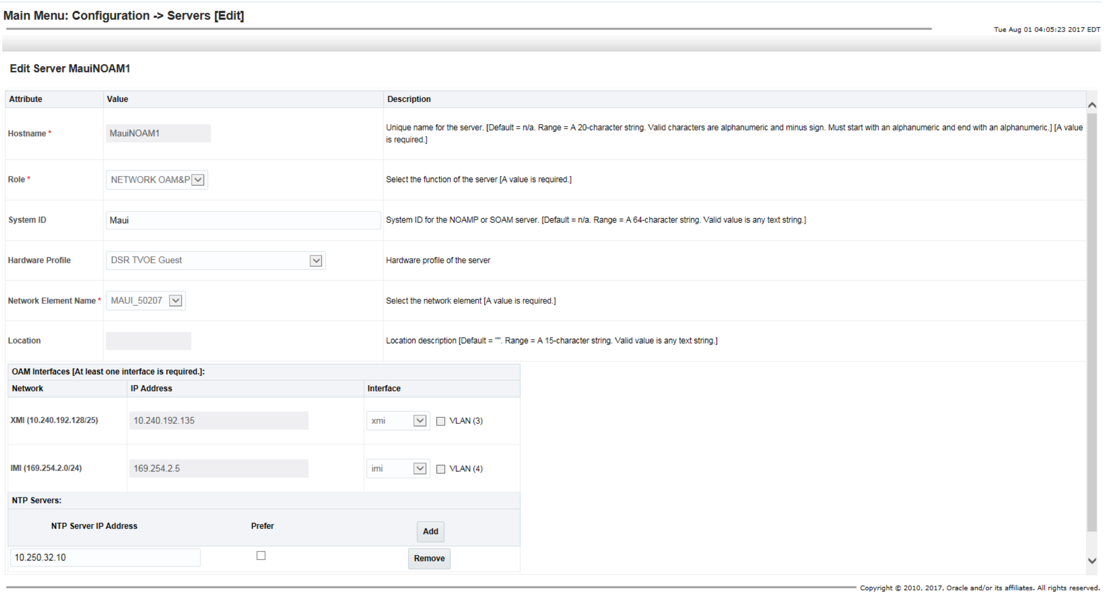
Diameter Signaling Router OS Standard Features

This chapter describes the security features of the Diameter Signaling Router (DSR) that is available to the Platform Administrator through the Linux Command Line Interface (CLI). The "platcfg" utility of the Operating System (OS) is used for configuring these features.

5.1 Configuring NTP Servers

Each server added at the NOAM server under **Administration** > **Configuration** > **Servers** has the option to specify the NTP server details. The NTP servers field is visible after selecting a network element. The following screen displays a configured server with NTP server details.

Figure 5-1 NTP Configuration GUI



Main Menu: Configuration -> Servers [Edit]

Tue Aug 01 04:05:23 2017 EDT

Edit Server MauiNOAM1

Attribute	Value	Description
Hostname *	MauiNOAM1	Unique name for the server. [Default = n/a. Range = A 20-character string. Valid characters are alphanumeric and minus sign. Must start with an alphanumeric and end with an alphanumeric.] [A value is required.]
Role *	NETWORK OAM&P	Select the function of the server [A value is required.]
System ID	Maui	System ID for the NOAMP or SOAM server. [Default = n/a. Range = A 64-character string. Valid value is any text string.]
Hardware Profile	DSR TVOE Guest	Hardware profile of the server
Network Element Name *	MAUI_50207	Select the network element [A value is required.]
Location		Location description [Default = "". Range = A 15-character string. Valid value is any text string.]

OAM Interfaces [At least one interface is required.]:

Network	IP Address	Interface
XMI (10.240.192.128/25)	10.240.192.135	xmi □ VLAN (3)
IMI (169.254.2.0/24)	169.254.2.5	imi □ VLAN (4)

NTP Servers:

NTP Server IP Address	Prefer	Add
10.250.32.10	□	Remove

For more information about how to add a server, see the *Inserting a Server* section under the *Servers* chapter in the *Operation, Administration, and Maintenance (OAM) Guide*.

5.1.1 Configuring NTP for the Host OS of the Application Guest VM

This section describes the procedure to configure NTF setting for the host Operating System hosting the application guest (for example, TVOE).

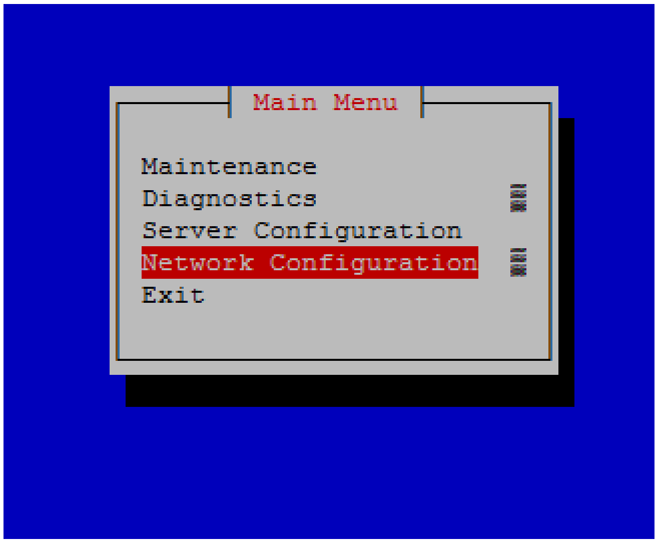
Perform the following steps:

1. Log in or switch user to `platcfg` user on the TVOE server.

   The platcfg main menu displays.

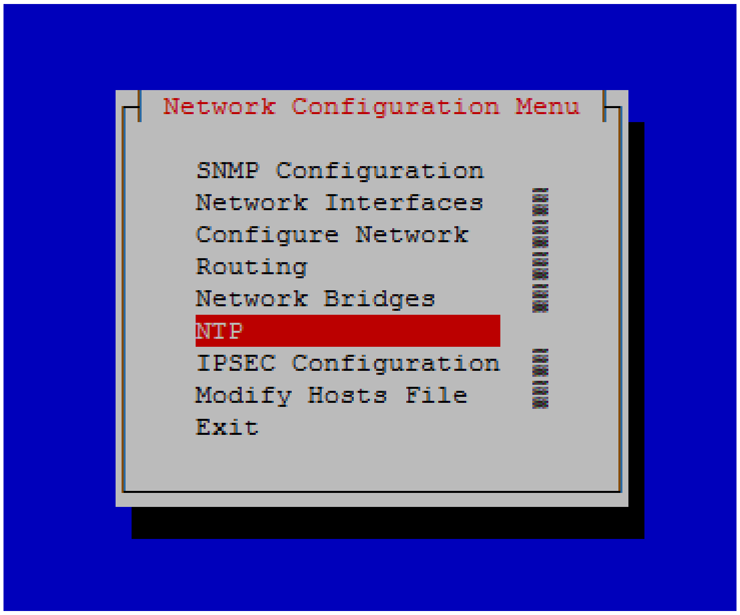
2. In the **Main Menu**, navigate to **Network Configuration**:

Figure 5-2 Main Menu



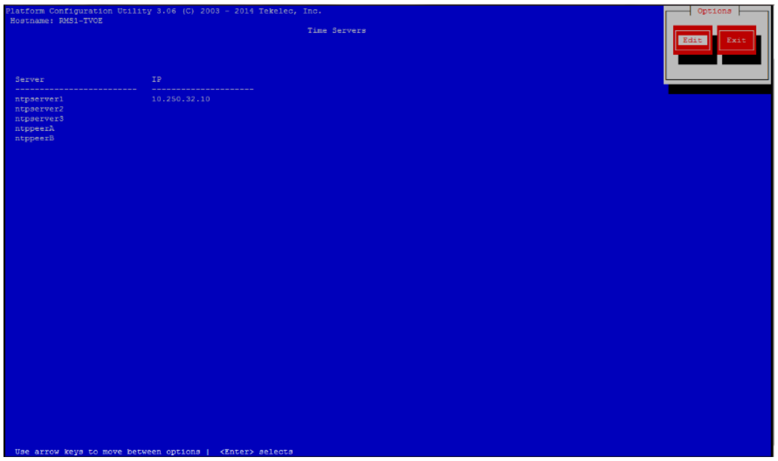
3. Select **NTP**.

Figure 5-3 Selecting NTP



4. The Time Servers screen shows the configured NTP servers and peers. Click **Edit**.

Figure 5-4   Time Servers



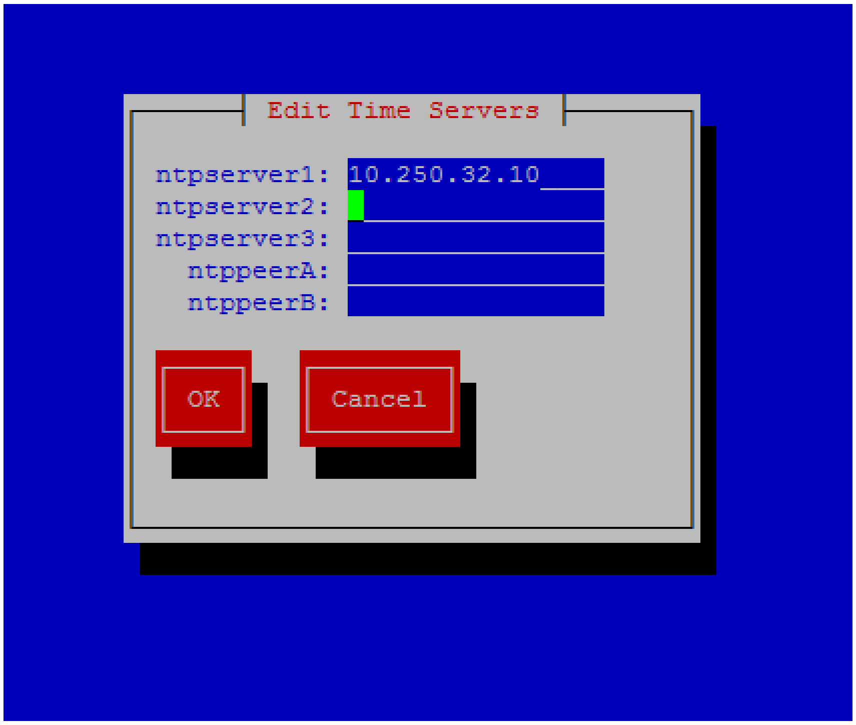
```
Platform Configuration Utility 3.06 (C) 2003 - 2014 Tekelec, Inc.
 Hostname: RMS1-TVOE
                                                              Time Servers

  Server                          IP
  -------------------------       ---------------------
  ntpserver1                      10.250.32.10
  ntpserver2
  ntpserver3
  ntppeerA
  ntppeerB

  Use arrow keys to move between options |  <Enter> selects
```

5. On the Edit Time Servers menu, enter the NTP Server information and click **OK**.

Figure 5-5   Edit Timer Servers Menu



```
Edit Time Servers

ntpserver1: 10.250.32.10
ntpserver2:
ntpserver3:
  ntppeerA:
  ntppeerB:
```

6. To exit TVOE, perform the following:
   a. Exit the platcfg menu.
   b. Ensure the time is set correctly. For more information on how to set the time on the TVOE host, see Setting the Time on the TVOE Host section.

5.2 Setting the Time on the TVOE Host

At the time of DSR installation, the date and time is set on TVOE hosts as follows:

1. Log in as **admusr**
2. Run the following commands:

   ```
   $ sudo /sbin/service ntpd stop
   $ sudo /usr/sbin/ntpdate ntpserver1
   $ sudo /sbin/service ntpd start
   ```

Result: The time is synchronized to the NTP server.

5.3 Configuring Password Settings for OS Users

This section describes the procedure to configure various password settings.

Perform the following procedure to configure various password settings for:

- Minimum password length
- Minimum time between password changes
- Maximum number of days that a password can be used
- Warning time for password expiration
- Minumum number of character differences between passwords
- Password history size (prevents reusing passwords)

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to open the platcfg menu:

   ```
   $ sudo su - platcfg
   ```

3. Select **Security** from the menu and press **Enter**.
4. Select **Sec Password Restrictions** option and press **Enter**.
5. Select **Global Password Restrictions for New Users** and press **Enter**.
6. Fill the appropriate settings:

   ```
   Minimum acceptable size for the new password: 15
   Minimum number of days allowed between password changes: 0
   Maximum number of days a password may be used: 99999
   Number of days a user is warned before password expiration: 7
   Minimum number of characters different between passwords: 0
   Minimum number of passwords between reuse: 5
   ```

7. Click **OK** and press **Enter**.
8. Select **Exit** in each of the menus until a command prompt is reached.

5.4 Configuring Passwords without Embedding Usernames

This section describes the procedure to ensure that the login name is not embedded in user passwords.

Perform the following steps to configure the password to not allow usernames to be embedded in it:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the `system-auth-ac` file:

   ```
   $ sudo rcstool co /etc/pam.d/system-auth-ac
   ```

3. Run the following command to add the reject_username setting to the `system-auth-ac` file:

   ```
   $ sudo sed -i -e '/^password.*reject_username/n' \
   -e '/^password.*pam_cracklib.so.*$/s/$/ reject_username/' \
   /etc/pam.d/system-auth-ac
   ```

4. Run the following command to check in the `system-auth-ac` file:

   ```
   $ sudo rcstool ci /etc/pam.d/system-auth-ac “reject_username”
   ```

5.5 Configuring Other Session and Account Settings for OS Users

You can configure various session and account settings for the following:

- Session inactivity
- Account locking for invalid login attempts
- Account locking for inactive accounts

5.5.1 Configuring Session Inactivity for OS Users

This section describes the procedure to configure session inactivity for OS users.

Perform the following procedure to configure session inactivity for OS users:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to open the platcfg menu:

   ```
   $ sudo su - platcfg
   ```

3. Select **Security** from the menu and press **Enter**.
4. Select **Idle Terminal Timeout** option from the security menu and enter the desired value in minutes for the **Idle Terminal Timeout** field.
5. Click **OK** and press **Enter**.
6. Select **Exit** in each of the menus until a command prompt is reached.

5.5.2 Configuring Number of Failed Login Attempts

This section describes the procedure to set the number of failed login attempts allowed before locking OS user accounts.

Perform the following procedure to configure the number of failed login attempts:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to open the platcfg menu:

   ```
   $ sudo su - platcfg
   ```

3. Select **Security** from the menu and press **Enter**.
4. Select **User Account Locking** from the menu and press **Enter**.
5. Fill out the following settings:

   ```
   Feature:  ( ) disable (*) enable
   Deny after # of attempts:  <max tries>
   Fail interval in minutes: <interval minutes>
   Unlock time in minutes: <unlock time>
   ```

6. Click **OK** and press **Enter**.
7. Select **Exit** in each of the menus until a command prompt is reached.

5.5.3 Configuring Lockout Time for Inactive Accounts

This section describes the procedure to set lockout time for inactive OS user accounts.

Perform the following procedure to lock inactive OS user accounts:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to open the platcfg menu:

   ```
   $ sudo su - platcfg
   ```

3. Select **Security** from the menu and press **Enter**.
4. Select **Inactive user locking** from the menu and press **Enter**.
5. Fill out the following settings:

   ```
   Feature:  ( ) disable (*) enable
   Deny after # of days of inactivity:  <max tries>
   ```

6. Click **OK** and press **Enter**.
7. Select **Exit** in each of the menus until a command prompt is reached.

5.6 Updating the TPD-Provd Cipher List

The procedure for this update defines the methods required to update the TPD-Provd cipher list and how to verify if the update was successful. For more detailed steps on performing these methods, refer to *Appendix P* in *PMAC Configuration Guide*.

5.7 Operational Dependencies on Platform Account Passwords

You must attempt to change passwords only on systems that are fully configured and stable. Modifying passwords during system installation is strongly discouraged. For detailed steps on performing these methods, refer to *PMAC Configuration Guide*.

5.8 Updating the SELinux Mode on the Server

By default, DSR ships with the SELinux mode as `disabled`. Run the following procedure to update the SELinux mode to `permissive`. You must run this procedure on each server in the topology.

The order of execution in the topology must be from A - level servers to C - level servers.

For A - level and B - level servers the sequence of execution must be Spare -> Stand-by -> Active.

Perform the following procedure to configure session inactivity:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the file `config` and update the SELinux state to `permissive`:

   ```
   $ sudo rcstool co /etc/selinux/config
   $ sudo sed -i 's/^SELINUX=.*$/SELINUX=permissive/g' /etc/selinux/config
   ```

3. Run the following command to check in the file `config`:

   ```
   $ sudo rcstool ci /etc/selinux/config
   ```

4. Run the following command to reboot the server:

   ```
   $ sudo init 6
   ```

6
Other Optional Configurations

The features explained in this section do not provide a GUI. The following features require the administrator to issue the Linux commands provided in the instructions.

6.1 Authentication for Single User Mode

This section describes the procedure to require authentication for single user mode.

Perform the following steps for each and every server in the topology:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the file `init` and grep for variable *PermitUserEnvironment* in the file:

   ```
   $ sudo rcstool co /etc/sysconfig/init
   $ grep ^SINGLE /etc/sysconfig/init
   ```

3. If no result is returned, run the following command:

   ```
   $ sudo echo "SINGLE=/sbin/sulogin" >> /etc/sysconfig/init
   ```

   If some result is returned after performing Step 2, then run the following command:

   ```
   $ sudo sed -i "s/SINGLE.*/SINGLE=\/sbin\/sulogin/g" /etc/sysconfig/init
   ```

4. Run the following command to check in the file `init`:

   ```
   $ sudo rcstool ci /etc/sysconfig/init
   ```

6.2 Changing OS User Account Default Passwords

This section describes the procedure to change the default passwords for all OS accounts that need to change the respective default passwords.

Perform the following steps to change the default passwords:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to change the passwords for each of the accounts being changed:

   ```
   $ sudo passwd <user account>
   Changing password for user <user account>.
   New UNIX password: <new password - will not display>
   Retype new UNIX password: <new password - will not display>
   passwd: all authentication tokens updated successfully.
   ```

3. Repeat steps 1 and 2 for all servers.

6.3 Changing Login Display Message

This section describes the procedure to change the login display message.

Perform the following steps to change the login display message:

1. Log in as `admusr` on the server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to create a backup copy of `sshd_config`:

   ```
   $ sudo cd /etc/ssh
   $ sudo cp sshd_config sshd_config.bak
   ```

3. Perform the following steps to edit the `sshd` configuration file:
   1. Edit the sshd configuration file.

      ```
      $ sudo rcstool co sshd_config
      $ sudo vi sshd_config
      ```

   2. Uncomment and edit `Banner /some/path` line to `Banner /etc/ssh/sshd-banner`.
   3. Save and exit the vi session.
4. Perform the following steps to edit the banner file.
   1. Edit the banner file.

      ```
      $ sudo vi sshd-banner
      ```

   2. Add and format the desired text.
   3. Save and exit the vi session.
5. Run the following command to restart the **sshd** service:

   ```
   $ sudo service sshd restart
   ```

6. Perform the following steps to verify the changes:

a. Test the change. Repeat steps 4 and 5 until the message is formatted correctly.

```
$ sudo ssh <current server name>
```

b. Verify message line feeds are formatted correctly.

```
$ exit
```

7. Run the following command to check the files into **rcs** to preserve changes during upgrades:

```
$ sudo rcstool init /etc/ssh/sshd-banner
$ sudo rcstool ci sshd_config
```

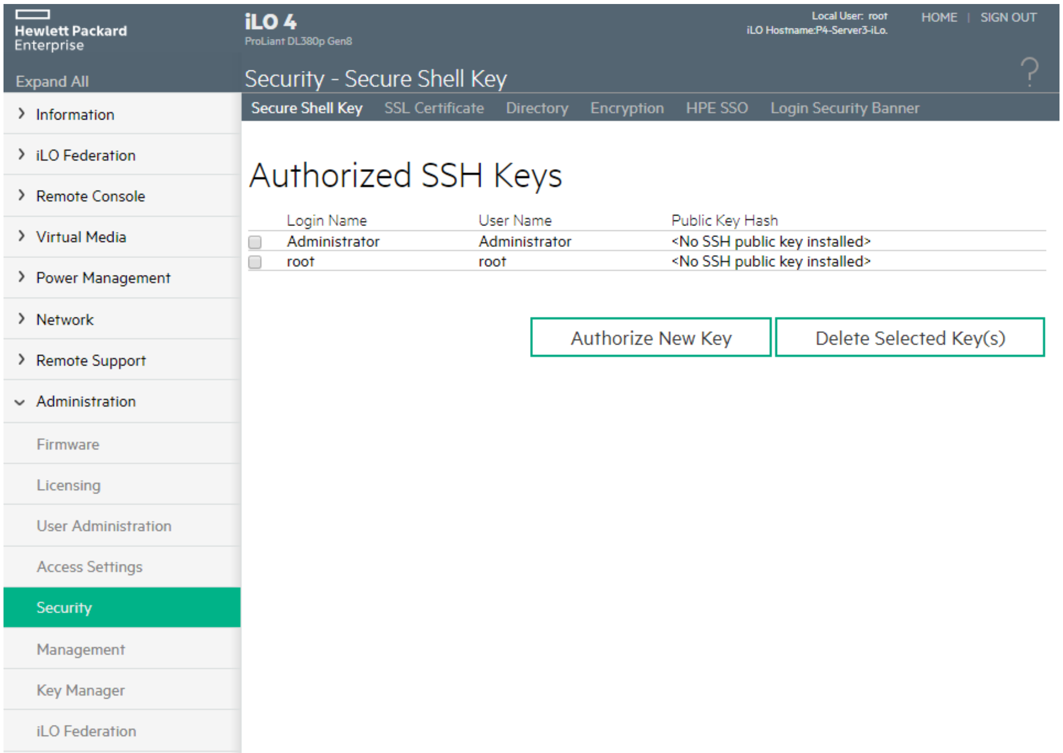
6.4 Forcing iLO to Use Strong Encryption

This section describes the procedure for an administrator to force iLO to use strong encryption.

Log in as an administrator to the iLO and perform the following steps:

1. On the **Administration** menu, click **Security**.

Figure 6-1 iLO Security Menu

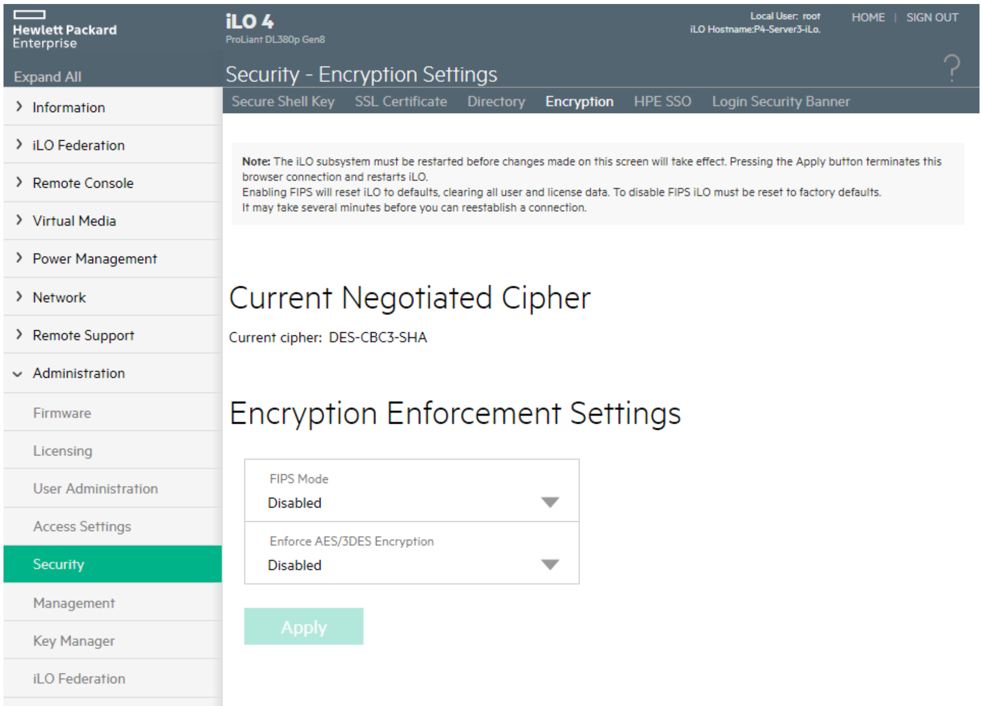


Authorized SSH Keys

	Login Name	User Name	Public Key Hash
□	Administrator	Administrator	<No SSH public key installed>
□	root	root	<No SSH public key installed>

2. Select **Encryption tab**, and under **Encryption Enforcement** Settings, set the Enforce AES/3DES Encryption to `Enabled`.

Figure 6-2 iLO Security Encryption Settings



Note: The iLO subsystem must be restarted before changes made on this screen will take effect. Pressing the Apply button terminates this browser connection and restarts iLO.
Enabling FIPS will reset iLO to defaults, clearing all user and license data. To disable FIPS iLO must be reset to factory defaults.
It may take several minutes before you can reestablish a connection.

Current Negotiated Cipher

Current cipher: DES-CBC3-SHA

Encryption Enforcement Settings

FIPS Mode: Disabled

Enforce AES/3DES Encryption: Disabled

3. Click **Apply**.
4. Logout and wait 30 seconds before logging back again.

6.5 Setting Up rsyslog for External Logging

This section describes the procedure to set up rsyslog for external logging to a central server from NOAMs and SOAMs.

Perform the following steps to set up rsyslog for external logging to a central server from NOAMs and SOAMs:

1. Log in as `admusr` on the source server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to enable remote logging:

   ```
   $ sudo syslog_config --remote=<IP of remote host to log to>
   ```

3. Repeat the steps on all necessary NOAMs and SOAMs.

> **Note:**
>
> The following restrictions exist:
>
> - Only OS level log events are forwarded, such as `/var/log/messages` and `/var/log/secure` content.
> - Application level logging is not included and should be accessed through the **Main Menu -> Administration -> Remote Servers -> Data Export** GUI screen.
> - Remote logging is over a non-secure communication channel that is not encrypted.

6.6 Adding sudo Users

This section describes how new OS users can perform priviledged operations through the configuration of the “sudo” capability.

The “sudo” configuration supports very granular authorization to an individual OS user for certain desired commands.

Perform the following procedure for the `admusr` to enter a password in order to run the commands using sudo access:

1. Log in as `admusr` on the source server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to check out the `plat.admusr.sudo` file:

   ```
   $ sudo rcstool co /usr/TKLC/plat/etc/sudoers.d/plat.admusr.sudo
   ```

3. Run the following command to suppress the NOPASSWD line:

   ```
   $ sudo sed -i '/^%admgrp ALL = NOPASSWD: ALL$/ s/^/#/' \
   /usr/TKLC/plat/etc/sudoers.d/plat.admusr.sudo
   ```

4. Run the following command to check in the `plat.admusr.sudo` file:

   ```
   $ sudo rcstool ci /usr/TKLC/plat/etc/sudoers.d/plat.admusr.sudo “require
   password”
   ```

After making this change, all activities through sudo by the `admusr` requires admusr password. Existing documentation does not and will not indicate this.

The sudo configuration file is constructed from piece parts; the syntax is complex, and editing mistakes could leave a system without the required access. For this reason, details of the configuration rules are available through *Oracle Help Center (OHC)* or by opening a ticket with My Oracle Support.

6.7 Reporting and Disabling Expired OS User Accounts

This section describes the procedure to report and disable expired user accounts.

Perform the following steps to report and disable expired user accounts:

1. Log in as `admusr` on the source server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to run the report of expired users:

   ```
   $ sudo lastlog -b <N>
   ```

   > **Note:**
   >
   > This command displays the users who have not logged in over N number of days. It also shows the users that have never logged in. To filter those users out of the display use the following command:
   >
   > ```
   > $ sudo lastlog -b <N> | grep -v Never
   > ```

3. Run the following commnd to disable the user accounts identified by the lastlog report:

   ```
   $ sudo passwd -l <user acct>
   ```

   > **Note:**
   >
   > Repeat this step for each user account you want to disable.

4. Run the following commnd to re-enable an account:

   ```
   $ sudo passwd -u <user acct>
   ```

   > **Note:**
   >
   > Repeat this step for each user account you want to re-enable.

7
Ethernet Switch Considerations

This section describes security-related configuration changes that can be made to the ethernet demarcation switches.

7.1 Configuring SNMP in Switches

This section describes the procedure to configure SNMP in all the essential switches.

It is essential to configure all the switches successfully according to the procedures described in *DSR C-Class Hardware and Software Installation Procedure 1/2 Guide* and *DSR C-Class Hardware and Software Installation Procedure 2/2 Guide*. For more information, see the References section.

- Configure Cisco 3020 switch (netConfig), and/or
- Configure HP 6120XG switch (netConfig), and/or
- Configure Cisco 4948/4948E/4948E-F (netConfig)

Perform the following steps to configure SNMP in all the switches:

1. Log into the server as root user and list all the configured switches by running the following command:

   ```
   # netConfig --repo listDevices
   ```

   Refer to application documentation to determine which switches to add/remove from the community string while making a note of the DEVICE NAME of each switch. This is used as `<switch_name>`.

2. For any given switch by `<switch_name>`, display SNMP community information by running the following command:

   ```
   # netConfig getSNMP --device=<switch_name>
   ```

3. For any given switch by `<switch_name>`, display its SNMP trap information by running the following command:

   ```
   #netConfig listSNMPNotify --device=<switch_name>
   ```

> **Note:**
>
> a. If the **Could not lock device** displays, type the following command to clear the lock to proceed:
>
> ```
> # netConfig --wipe --device=<switch_name>
> ```
>
> b. Reply **y**, if prompted.

7.2 Configuring Community Strings

This section describes the procedure to configure community strings.

Perform the following steps to configure community strings:

1. To add a community string to ANY switch by `<switch name>`, type the following command with appropriate switch name:

   ```
   #netConfig addSNMP --device=<switch name> community=<community
   string> uauth=RO
   ```

2. To delete a community string to ANY switch by `<switch name>`, type the following command with appropriate switch name:

   ```
   #netConfig deleteSNMP --device=<switch_name>
   community=<community_string>
   ```

7.3 Configuring SNMP Traps

This section describes the procedure to configure traps.

Perform the following steps to configure the traps:

1. To add a trap server, type the following command with appropriate switch name:

   ```
   #netConfig addSNMPNotify --device=<switch_name>
   host=<snmp_server_ip> version=2c auth=<community_string>
   [traplvl=not-info]
   ```

2. To delete a trap server, type the following command with appropriate switch name:

   ```
   #netConfig deleteSNMPNotify --device=<switch_name>
   host=<snmp_server_ip> version=2c auth=<community_string>
   [traplvl=not-info]
   ```

Note:

`traplvl=not-info` in the command is needed only in case of the 6120XG, 6125G, and 6125XLG switches. The switches 4948 or 3020 do not need this field in the above commands.

8 Security Logs and Alarms

The **Security Log** page in the GUI allows you to view the application's historical security logs from all the configured security logs. These logs are displayed in a scrollable, optionally filterable table. You can also export the security logs to the file management area in a `.csv` format. For more details, see the *Security Log* chapter in the *Operation, Administration, and Maintenance (OAM) Guide*.

Application Alarms and Events are unsolicited messages used in the system for trouble notification and to communicate the status of the system to the Operations Services. The application merges unsolicited alarm messages and unsolicited informational messages from all servers in a network and notifies their occurrence to you. Security alarms enable a network manager to detect security events early and take corrective action to prevent degradation in the quality of service.

Alarms provide information about the operational condition of a system for a network manager to act upon when the need arises. Alarms can have the following severity:

- Critical
- Major
- Minor
- Cleared

For more information, see the *Alarms and Events* and *Security Log* chapters in *Alarms and KPIs Reference Guide*, *Measurements Reference Guide* and *DSR Operation, Administration, and Maintenance (OAM) Guide*.

OS-level logging is captured in:

- `/var/log/messages` – general system messages
- `/var/log/secure` – security related messages
- `/var/log/httpd` (directory) – apache webserver logging

9 Optional IPsec Configuration

This section describes the security related to configuration changes required to use Internet Protocol Security (IPsec).

> **Note:**
>
> Customers are NOT required to configure IPsec.

9.1 IPsec Overview

Internet Protocol Security (IPsec) provides network layer security protocols for authentication, encryption, payload compression, and key exchange. IPsec provides Host-to-Host encrypted connections or Network-to-Network packet tunneling.

Network traffic between two end-points is encrypted and decrypted by authenticated hosts at the end-points, using a shared private key. The shared private key forms a Security Association that can be automatically changed using Security Policies based on traffic volume, expiry time, or other criteria.

IPsec works for both IPv4 and IPv6 on the Diameter interface. The provisioning interface only supports IPsec on IPv4. Oracle Communications Diameter Signaling Router supports IPsec with an SCTP/IPv6 configuration.

9.1.1 Encapsulating Security Payload

The Diameter Signaling Router IPsec uses the Encapsulating Security Payload (ESP) protocol for encryption and authentication. ESP uses encryption algorithms to encrypt either the packet payload or the entire packet, depending on the IPsec configuration, whether to use transport mode or tunnel mode. If the IPsec is in transport mode, only the packet payload is encrypted, and the IP header is not encrypted. If the IPsec is in tunnel mode, both the packet payload and the original IP header are encrypted, and then a new IP header is added.

ESP also provides authentication of the encrypted packets to prevent attacks by ensuring that the packet is from the correct source.

Many encryption algorithms use an Initialization Vector (IV). The IV encrypts to make each message unique. This makes it extremely difficult for cryptanalysis attempts to decrypt the ESP. For more details on the supported ESP encryption and authentication algorithms, see Table 9-1 table.

9.1.2 Internet Key Exchange

Internet Key Exchange (IKE) is used to exchange secure keys to set up IPsec security associations. There are two versions of IKE: IKEv1 and IKEv2. The following are the main differences that exist between IKEv1 and IKEv2:

- IKEv1:

  - security associations are established in 8 messages.
  - does not use a Pseudo-Random function.
- IKEv2:
  - security associations are established in 4 messages.
  - uses an increased number of encryption algorithms and authentication transformations.
  - uses a Pseudo-Random function.

For more details on the encryption algorithms and authentication transformations that are supported for IKE, see Table 9-1 table.

9.2 IPsec Process

When an IPsec connection is configured, you can create Security Policies using the IPsec connection configuration files. IPsec uses Security Policies to define whether to encrypt a packet or not. The Security Policies also help determine whether an IPsec procedure is needed for a connection. The Security Policies do not change over time. After the Security Policies exist and initial network connectivity has been made, the Internet Key Exchange (IKE) process occurs.

IKE operates in two phases:

1. **Phase 1** acts as an initial handshake and creates the IKE security associations that determine how to set up an initial secure connection to begin the IPsec security association negotiation.
2. In **phase 2**, the keys are exchanged, and the IPsec security associations are created. Once the IPsec security associations exist, the IPsec connection setup process is complete. IPsec now knows how to encrypt the packets.

IPsec uses security associations to determine which type of encryption algorithm and authentication transportation should be used when creating an IPsec packet, and to apply the correct decryption algorithm when a packet is received. Since security associations change with time, a lifetime parameter is used to force the security associations to expire so that IPsec must renegotiate them.

You can set up an IPsec connection on a virtual IP that can be used for High Availability (HA). However, when a switchover occurs and the virtual IP is added to the new box, a SIGHUP is sent to the iked daemon on the newly active box so that the virtual IP is under iked management. Also, the switchover does not occur until the security associations have expired and the renegotiation can begin.

9.3 Setting Up IPsec

This section describes the procedure to set up IPsec.

Prerequisites

Before configuring IPsec, perform the following steps on the active NOAMP server:

1. Log in as root on the active NOAMP server.

2. On the active NOAMP server, run the following commands:

```
iadd -xu -fallowPgmChg -fname -fvalue LongParam \
<<'!!!'
Yes|cm.ha.enableIpsecWhack|1
!!!
```

Procedure

Adding an IPsec connection also configures it. You can edit or delete an existing IPsec connection. You can also start (enable) and stop (disable) an IPsec connection without deleting that connection completely. Perform IPsec setup on each MP that can control the connection.

> **Note:**
>
> You must not enableIPsec on a live connection. Disable the connection before enabling IPsec.

Perform the following steps to set up a new IPsec connection:

1. Open **platcfg**.
2. Add and configure an IPsec connection. For more information, see the Adding an IPsec Connection section.
3. Select an IKE version.
   a. Complete the IKE configuration for the IPsec connection.
   b. Complete the ESP configuration for the IPsec connection.
   c. Complete the IPsec connection configuration entries.
   d. Wait for the connection to be added.
4. Enable the IPsec connection. For more information, see the Enabling or Disabling Host Intrusion Detection System section.
5. Logout of **platcfg**.
6. Restart IPsec service by typing this command:

```
# service ipsec restart
```

9.4 IPsec IKE and ESP Elements

The following table describes the IPsec IKE and ESP configuration elements and provides default values where applicable:

Table 9-1 IPsec IKE and ESP Elements

Description	Valid Values	Default
Internet Key Exchange (IKE) Version	ikev1, ikev2	ikev2
IKE Configuration		

Table 9-1 (Cont.) IPsec IKE and ESP Elements

Description	Valid Values	Default
IKE Encryption	aes128_cbc, aes192_cbc, aes256_cbc, 3des_cbc, hmac_md5	aes128_cbc hmac_md5
IKE Authentication	hmac_sha1, aes_xcbc, hmac_md5	hmac_md5
Pseudo Random Function This is used for the key exchange only for ikev2.	hmac_sha1, aes_xcbc (ikev2)	
Diffie-Hellman Group The group number is used to generate the group (group - set of numbers with special algebraic properties) that is used to select keys for the Diffie-Hellman algorithm. The larger the group number, the larger the keys used in the algorithm.	2, 14 (ikev2) 2 (ikev1)	2 (IKEv1) 14 (IKEv2)
IKE SA Lifetime Lifetime of the IKE/IPsec security associations. A correct lifetime value would be <hours/mins/secs>. Example: 3 mins. **Note:** If a connection goes down, it does not re-establish until the lifetime expires. If the lifetime is set to 60 minutes and a failure causing a switchover of a VIP is required, the switchover does not occur until the 60 minutes expire. The recommendation is to set the lifetime to the lowest possible time that does not impact network connectivity, such as 3-5 minutes.	Number of time units	60
Lifetime Units	hours, mins, secs	mins
Perfect Forward Secrecy This is an algorithm used to ensure that if one of the private keys is compromised the other keys are not compromised.	yes, no	yes
ESP Configuration		
ESP Authentication Algorithm used to authenticate the encrypted ESP.	hmac_sha1, hmac_md5	hmac_sha1
Encryption Algorithm Algorithm used to encrypt the actual IPsec packets.	aes128_cbc, aes192_cbc, aes256_cbc, 3des_cbc	aes128_cbc

9.5 Adding an IPsec Connection

Perform the following steps to add an IPsec connection:

1. Log in as **admusr** on the source server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Enter the following command to open the **platcfg** menu:

   ```
   $ sudo su - platcfg
   ```

3. To update IPsec configuration:

   a. Select **Network Configuration**.
   b. Select **IPsec Configuration**.
   c. Select **IPsec Connections**.
   d. Click **Edit**.

4. To add a new connection:
   a. Select **Add Connection**.
   b. Select the Internet Key Exchange Version: either **IKEv1** or **IKEv2**.
   c. Complete the **IKE Configuration** fields for the desired connection, then click **OK**.

   For more details about the fields, see Table 9-1 table.

5. Select the desired ESP Encryption algorithm, and click **OK**.

   For more details about the fields, see Table 9-1 table.

6. Complete the **Add Connection** fields for the desired connection.
   a. Enter **Local Address**.
   b. Enter **Remote Address**.
   c. Enter **Pass Phrase**.

   > **Note:**
   >
   > Select a non-trivial passphrase.

   d. Select **Mode**.

7. Click **OK**.

   Wait for the connection to be added.

   When the connection has been successfully added, the Internet Key Exchange Version menu displays.

8. Select **Exit** in each of the menus until a command prompt is reached.

9.6 Editing an IPsec Connection

Perform the following steps to edit an IPsec connection:

1. Log in as **admusr** on the source server.

```
login: admusr
Password: <current admin user password>
```

2. Enter the following command to open the **platcfg** menu:

```
$ sudo su - platcfg
```

3. To select an IPsec connection:
   a. Select **Network Configuration**.
   b. Select **IPsec Configuration**.

   c. Select **IPsec Connections**.
   d. Click **Edit**.

4. To edit an IPsec connection:
   a. Select **Edit Connection**.
   b. Select **IPsec connection** to edit.
   c. View the IPsec connection’s current configuration.
   d. Click **Edit**.
5. To configure required IKE fields:
   a. Select either **IKEv1** or **IKEv2**.
   b. Complete the **IKE Configuration** fields if needed, then click **OK**.

      For more details about the fields, see the Table 9-1 table.
6. Select the desired **ESP Configuration** fields, and click **OK**.

   For more details about the fields, see the Table 9-1 table.
7. Complete the **Add Connection** fields for the desired connection.
   a. Enter the **Local Address**.
   b. Enter the **Remote Address**.
   c. Enter the **Pass Phrase**.
   d. Select the **Mode**.
8. To restart the connection:
   a. Click **OK**.
   b. Select **Yes** to restart the connection.

      When the connection has been successfully updated, the Internet Key Exchange Version menu displays.
9. Select **Exit** in each of the menus until a command prompt is reached.

9.7 Enabling and Disabling an IPsec Connection

Perform the following steps to enable or disable an IPsec connection:

1. Log in as **admusr** on the source server.

   ```
   login: admusr
   Password: <current admin user password>
   ```
2. Enter the following command to open the **platcfg** menu:

   ```
   $ sudo su - platcfg
   ```
3. To edit IPsec configuration:
   a. Select **Network Configuration**.
   b. Select **IPsec Configuration**.
   c. Select **IPsec Connections**.

   d. Click **Edit**.

4. To edit IPsec connection:

   a. Select **Edit Connection**.

   b. Select **IPsec connection** to edit.

   c. View the IPsec connection's current configuration.

   d. Click **Edit**.

5. To configure a IPsec connection:

   a. Select **Connection Control**.

   b. Select **IPsec connection** to enable or disable.

   c. Select **Enable** or **Disable**.

6. Click **OK** to enable or disable the selected IPsec connection.

7. Select **Exit** in each of the menus until a command prompt is reached.

9.8 Deleting an IPsec Connection

Perform the following steps to delete an IPsec connection:

1. Log in as **admusr** on the source server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Enter the following command to open the **platcfg** menu:

   ```
   $ sudo su - platcfg
   ```

3. To edit IPsec configuration:

   a. Select **Network Configuration**.

   b. Select **IPsec Configuration**.

   c. Select **IPsec Connections**.

   d. Click **Edit**.

4. To edit IPsec connection:

   a. Select **Delete Connection**.

   b. Select **IPsec connection** to delete.

   c. Click **Yes** to confirm the delete.

   Wait for the connection to be deleted.

   When the IPsec connection has been successfully deleted, the **Connection Action** menu displays.

5. Select **Exit** in each of the menus until a command prompt is reached.

10 Firewall Configuration Changes

This chapter describes the firewall configuration changes for Diameter Signaling Router (DSR).

10.1 IP tables

DSR comes with various IP tables rules preconfigured and dynamically adjusts IP table rules as new diameter peers are defined. In general, we do not recommend making any IP table rule adjustments without prior consultation with DSR product support.

10.2 TCP Wrappers

DSR does not use TCP wrappers. Customers wishing to add TCP wrapper rules (`hosts.allow / hosts.deny`) must ensure that the management and signaling traffic are not impacted. In general, we do not recommend making any TCP Wrapper rule adjustments without prior consultation with DSR product support.

11 Internal Web Services

DSR uses many internal web services in support of centralized configuration and management. These web services use the SOAP protocols and implement WS-Security profiles to authenticate internal clients. These services ship with self-signed certificates and default passwords. You must plan to update the default passwords during installation. You can also replace the self-signed certificates with certificates signed by a trusted authority. The following sections provide procedures to perform these actions.

11.1 Changing Internal Web Service Passwords

In general, after the initial configuration is complete and before deploying or turning up services, you must update the internal web service passwords.

11.2 Changing TPD Web Service Password

Perform the following procedures to change the OS-level provisioning web service password.

Updating TPD Web Service Password on Active NO

1. Log in as `admusr` on the source server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following command to reset the TPD web service password:

   ```
   $ /usr/TKLC/appworks/sbin/resetTpdPassword
   ```

3. You are prompted to provide a password:

   ```
   password: <enter the new password>
   ```

   Step result: The command copies and installs the new password to each reachable server in the topology, and flushes client password caches.

4. Run the following command to verify that the web service is still functional:

   ```
   $ AppWorks Network interfaces
   ```

   Step result: You can see a list of network interfaces reported by the Web Service backend:

   ```
   {
   “element”:[
   “eth0”,
   “eth1”
   ```

```
]
}
```

This update command synchronizes the TPD web service (`tpdprovd`) password on all reachable servers in the topology. Any servers added to the topology after running this command are automatically configured to use the new password. If any servers were not reachable when this command is run, run the command again later when those servers are reachable.

Updating TPD Web Service Password on PMAC

Some DSR deployments include a PMAC system to support installation and growth. Once you update the tpdProvd password on servers in the DSR topology, the PMAC loses the ability to inventory deployed DSR nodes. You can restore the inventory function by running the following procedure on the PMAC:

1. Update the password on a given server or group of servers (assuming all passwords are the same for the group) either using `linux passwd` command on the server(s) or by some other means.
2. From a PMAC shell, use the following command to add the password(s) to the PMAC database and update the PMAC messaging interface. This command prompts the user for the password and echo asterisks as characters are entered.

   ```
   /usr/bin/sudo /usr/TKLC/smac/bin/pmacadm addProvdCredentials --
   flushBAs=yes
   ```

   > **Note:**
   >
   > `--flushBAs` can be set to `no` if entering multiple passwords and set to `yes` on the last added password. If `--flushBAs` is not set to `yes` on the last password entry, a sentry restart must be performed on the PMAC to flush out all the Broker Agents (server interfaces) in the PMAC messaging system and must be rebuilt using the new passwords.

   a. The new password can be verified using the following command:

      ```
      /usr/bin/sudo /usr/TKLC/smac/bin/pmaccli getHostCommStr --
      ip=<ipv4 address of the server> --accessType=ro
      ```

      This should return a valid response with a password. If it fails, there may be a tpdProvd password mismatch issue between the PMAC and the server.

   b. If a password must be removed and the exact spelling of the password is known, it can be deleted from the PMAC database and messaging system using the following command:

      ```
      /usr/bin/sudo /usr/TKLC/smac/bin/pmacadm deleteProvdCredentials
      --flushBAs=yes
      ```

> **Note:**
>
> You are prompted for the password.

11.3 Changing the Configuration Web Services Password

The following procedures are used to change the configuration web services password.

Update Configuration Web Service Password on Active NO

Perform the following steps to update configuration web service password on an active NO:

1. Log in as `admusr` on the active NOA server.

   ```
   Login: admusr
   Password: <current admin user password>
   ```

2. Reset the TPD web service password by running:

   ```
   $ /usr/TKLC/appworks/sbin/resetSoapPassword
   ```

   You are not be prompted for a password. The `resetSoapPassword` command generates a large random string which is used as the new password.

   > **Note:**
   >
   > The command copies and installs the new password to each reachable server in the topology, and flushes client password caches. You might see output related to these activities.

3. Restart all the servers in the topology from active NOA GUI:
   a. Log in to the active NOA GUI.
   b. Navigate to **Main Menu**, and then **Status & Manage**, and then **Server**.
   c. Restart all the servers in the topology in below mentioned order.
      i. The non-Active OAM Servers, that is Standby or Spare NO, Standby or Spare SO, DR-NO.
      ii. All the C-level servers.
      iii. The active OAM servers, that is active NO and active SO.
4. Verify if the web service is functional:

   ```
   $ AppWorks Alarms getData
   ```

You should see a list of active alarms as reported by the Web Service backend.

```
[
    <alarm list (if any)>
]
```

This update command synchronizes the configuration web services password on all reachable servers in the topology. After running this command, any servers added to the topology is configured to use the new password. If any servers were not reachable when this command is running, then run the command again later when those servers are reachable.

Some DSR deployments include an IDIH system to support message trace and debugging. Once you update the servers in the DSR topology, IDIH loses the ability to interact with the deployed DSR nodes. You can restore the IDIH function by running this procedure on the IDIH:

1. Log in as `admusr` on the active NOA server.

   ```
   Login: admusr
   Password: <current admin user password>
   ```

2. Retrieve the current configuration web services password in plain text. This is needed below in step 4.

   ```
   $ /usr/TKLC/appworks/bin/aw.wallet credential get cmsoapa password
   ```

   The command prints the current plain text configuration web service password

   Example:

   ```
   7w57q9U0OvOtKtgtLVTMajDcXfhCj2F4nyXw45qK6EXNHA9jACyQ
   ```

3. Log in as `admusr` on the IDIH application server.

   ```
   Login: admusr
   Password: <current admin user password>
   ```

4. Change the user to tekelec by executing `sudo su - tekelec` command. Then, reset the configuration web service password by running:

   ```
   $ cd /usr/TKLC/xIH/apps/trace-refdata-adapter/
   $ ./resetSoapPassword.sh
   ```

   You are prompted to provide a password:

   ```
   password: <enter the password from step 2>
   ```

   The command stores the new SOAP password into IDIH Oracle database.

5. After executing the command in Step 4, the WebLogic application server has to be restarted on IDIH application server. Type exit to become admusr.

   ```
   sudo service xih-apps stop
   sudo service xih-apps start
   ```

   The Weblogic server may take a few minutes to resume its service after executing the command.

   > **Note:**
   >
   > - TraceRefDataAdapter(TRDA) sync must happen automatically after WebLogic server has been restarted. If TRDA sync does not happen automatically, then execute the following command to sync IDIH with DSR. As tekelec user, navigate to /usr/TKLC/xIH/apps/trace-refdata-adapter directory and execute the command ./trda-config.sh < SOAM VIP >, where <SOAM VIP> is a place-holder for SOAM VIP address.
   > - To verify TRDA sync, look into application.log in the path /var/TKLC/xIH/log/apps/weblogic/apps/application.log. Ensure that this log does not show any java exceptions.

11.4 Changing Internal Web Service Certificates and Key Material

In general, the TPD and web services are configured to work with self-signed certificates. You can replace the appworks certificates using the procedures described in this section.

Assumptions

The following procedure assumes that you have already obtained a signed certificate or key file from the customer's certificate authority and that these files are in the .pem format.

Each server in the topology needs its own certificate or key pair. The certificate must have a DN field that matches the hostname of the server. The following procedures assume the customer provides files in this naming convention:

- <hostname>_crt.pem – a PEM encoded X.509 certificate for the host <hostname>
- <hostname>_priv.pem – a PEM encoded private key for the host <hostname>

Ensure that the private key file is not protected with a passphrase.

Creating and Distributing a separate Certificate and Key PEM File

Perform the following steps to create and distribute a separate certificate and key PEM file:

1. Log in as admusr on the active NOA server.

   ```
   Login: admusr
   Password: <current admin user password>
   ```

2. Copy all of the `<hostname>_crt.pem` and `<hostname>_priv.pem` files to the home directory for `admusr` on the active NOA using a utility such as `scp` or `rsync`.
3. Run the following steps to confirm and verify the certificate or key pairs:
   a. Confirm each of the certificate or key pairs are compatible (assume `<hostname>` is **noa**):

      ```
      $ openssl rsa -noout -modulus -in noa_priv.pem | openssl md5
      (stdin)= eef417fb3f018862034ae8e9f3a0b56e


      $ openssl x509 -noout -modulus -in noa_crt.pem | openssl md5
      (stdin)= eef417fb3f018862034ae8e9f3a0b56e
      ```

   b. Verify the md5 output matches for each `<hostname>` certificate or private key pair. Additionally, the md5 should be different for different `<hostnames>`.
4. Copy the private key and certificate to the server (again, assume <hostname> is noa):

   ```
   $ scp noa_priv.pem admusr@noa:
   $ scp noa_crt.pem admusr@noa:
   ```

> **Note:**
>
> Repeat the above procedure for each `<hostname>`.

Installing a separate PEM and CERT File on Each Distinct <hostname>

After creating the separate certificate and private key PEM file for each server in the topology, you must log into each server in the topology and install the PEM file as follows:

1. Log in as `admusr` on the `<hostname>` (assume `<hostname>` is **noa**):

   ```
   $ ssh admusr@noa
   ```

2. Run the following commands to copy your new certificate/private key pair PEM file into place (assume `<hostname>` is **noa**):

   ```
   $ sudo cp noa_priv.pem /usr/TKLC/appworks/etc/ssl
   $ sudo chown awadmin:awadm /usr/TKLC/appworks/etc/ssl/noa_priv.pem
   $ sudo chmod 640 /usr/TKLC/appworks/etc/ssl/noa_priv.pem

   $ sudo cp noa_crt.pem /usr/TKLC/appworks/etc/ssl/
   $ sudo chown awadmin:awadm /usr/TKLC/appworks/etc/ssl/noa_crt.pem
   $ sudo chmod 640 /usr/TKLC/appworks/etc/ssl/noa_crt.pem
   ```

3. Run the following commands to replace the existing combined certificate or private key file with the new file:

   ```
   $ sudo mv /usr/TKLC/appworks/etc/ssl/server.crt /usr/TKLC/
   appworks/etc/ssl/old_server.crt
   ```

```
$ sudo ln -s /usr/TKLC/appworks/etc/ssl/noa_crt.pem /usr/TKLC/
appworks/etc/ssl/server.crt

$ sudo mv /usr/TKLC/appworks/etc/ssl/server.pem /usr/TKLC/
appworks/etc/ssl/old_server.pem

$ sudo ln -s /usr/TKLC/appworks/etc/ssl/noa_priv.pem /usr/TKLC/
appworks/etc/ssl/server.pem
```

4. Run the following commands to restart the configuration web services and exit:

```
$ sudo pm.kill apwSoapServer
$ sudo pm.kill cmsoapa
$ exit
```

Note:

Repeat the above procedure for each and every distinct `<hostname>`.

12
Updating the MySQL Password

Perform the following procedure to change the MySQL password. Ensure that the following commands are executed only from Active NO:

1. Log in as `admusr` on the source server.

   ```
   login: admusr
   Password: <current admin user password>
   ```

2. Run the following steps to update password for default user or root user:

   a. Run the following command to reset the MySQL **default user** password:

      ```
      $ /usr/TKLC/appworks/bin/resetMysqlPassword
      ```

   b. You are prompted to provide a password:

      ```
      Enter password: <enter the new password>
      Enter Password Again: <re-enter the new password>
      ```

   c. Run the following command to reset the MySQL **root** password:

      ```
      $ /usr/TKLC/appworks/bin/resetMysqlPassword root
      ```

   d. You are prompted to provide a password:

      ```
      Enter password: <enter the new password>
      Enter Password Again: <re-enter the new password>
      ```

Result: The command copies the new password to each reachable server in the topology, and flushes client password caches.

This update command synchronizes the MySQL password on all reachable servers in the topology. Any servers added to the topology after running the update command are automatically configured to use the new password. No server in the topology should be rebooting while the password is being changed. If any servers were not reachable when this command was executed, run the command again later when those servers are reachable.

> **Note:**
>
> You must run the `resetMysqlPassword` script only after all the servers in the topology have been upgraded to DSR 8.5 or later.

13 Appendix

This chapter describes the Appendix for DSR.

13.1 Secure Deployment Checklist

The following security checklist helps you secure Oracle Communications Diameter Signaling Router (DSR) and its components:

- Change default passwords
- Utilize LDAP for authentication purposes
- Utilize authorized IP addresses feature
- Use TLS or IPSEC
- Enforce strong password management
- Restrict admin functions to the required few administrator groups
- Configure community strings and traps explained in Other Optional Configurations chapter
- Restrict network access by enabling the DSR firewall feature
- Enforce iLO to use strong encryption
- Available Ciphers for SSH and HTTPS/SSL

The DSR system has been preconfigured to require modern strong ciphers for both SSH and TLS. The supported ciphers/MACs for SSH connections are:

```
Ciphers aes128-ctr,aes192-ctr,aes256-ctr
MACs hmac-sha2-512,hmac-sha2-256
```

This is configured in `/etc/ssh/sshd_conf`. The supported cipher set (using openssl notation) for HTTPS/TLS is:

```
ECDH+AES128:ECDH+AESGCM:ECDH+AES256:DH+AES:DH+AESGCM:DH+AES256:RSA+AES:RSA+AE
SGCM:!aNULL:!MD5:!DSS:!SSLv3:!3DES
```

For the default TLS (https) connection, this is configured in `/etc/httpd/conf.d/ssl.conf`. For certificates loaded via the GUI, this is configured in `/var/TKLK/appworks/etc/https.template`.

For detailed information on importing HTTPS or SSL Certificate into VNFM, see the *DSR VNFM Installation and User Guide*.