Oracle® Communications Application Session Controller

Security Guide

Release 3.8.0

July 2019

ORACLE®

Contents

Figures:

Chapter 1: OCASC Security Overview

This chapter describes basic security considerations and provides an overview of OCASC security.

Basic Security Considerations

The following principles are fundamental to using any application securely:

- **Keep software up to date.** This includes the latest product release and any patches that apply to it.

- **Limit privileges as much as possible.** Users should be given only the access necessary to perform their work. User privileges should be reviewed periodically to determine relevance to current work requirements.

- **Monitor system activity.** Establish who should access which system components, and how often, and monitor those components.

- **Install software securely.** For example, use firewalls, secure protocols using TLS (SSL), and secure passwords. See “Performing a Secure OCASC Installation” for more information.

- **Learn about and use the OCASC security features.** See “Implementing OCASC Security” for more information.

- **Use secure development practices.** For example, take advantage of existing database security functionality instead of creating your own application security. See “Security Considerations for Developers” for more information.

- **Keep up to date on security information.** Oracle regularly issues security-related patch updates and security alerts. You must install all security patches as soon as possible. See the “Critical Patch Updates and Security Alerts” Web site:

  http://www.oracle.com/technetwork/topics/security/alerts-086861.html

Overview of OCASC Security

Oracle Communications Application Session Controller is an integrated communications controller and web application server that enables service providers, enterprises, and contact centres to orchestrate and control real-time voice and video communications using web services.

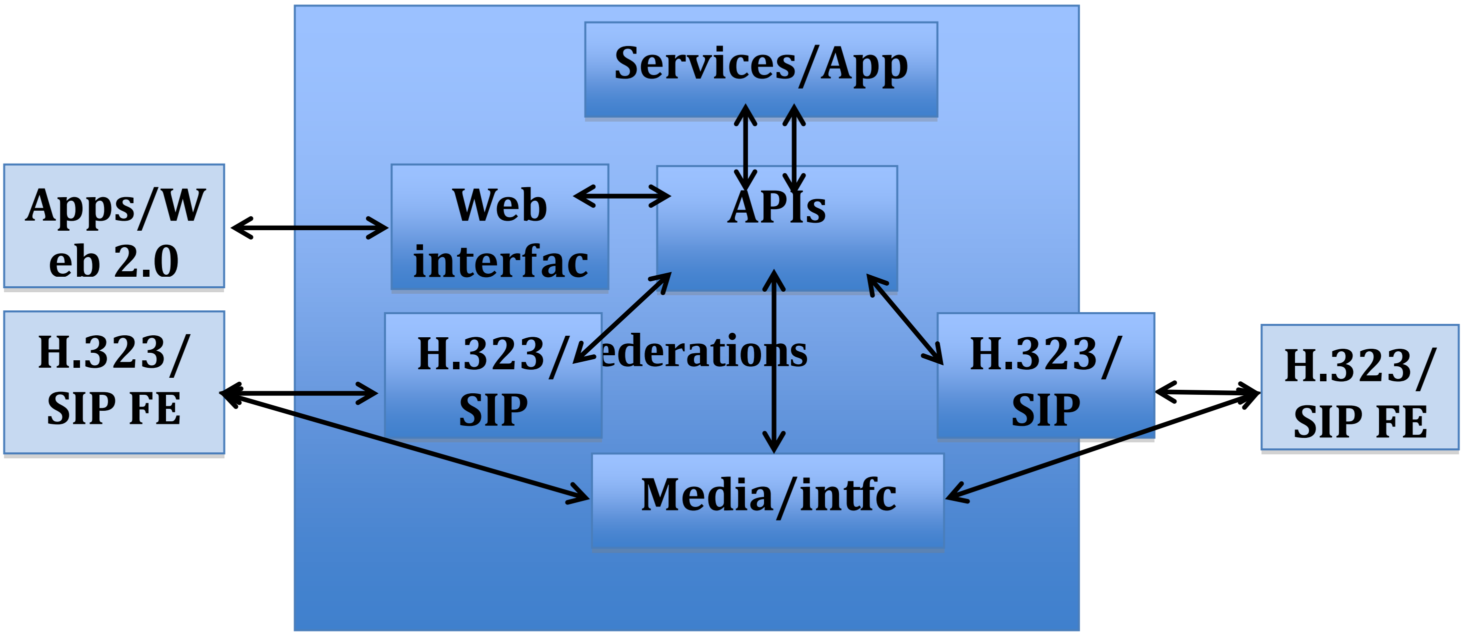


Figure 1: Functional Elements Logical Flow

As a standards-based solution, Oracle Communications Application Session Controller supports the most widely adopted IP communications protocols (SIP and H.323) and web services APIs (REST and SOAP) for ultimate choice and flexibility. Developers can leverage existing web development frameworks and toolsets to communications-enable business processes or web pages. The corresponding functional elements logical flow shows in figure 1.

OCASC is developed with security in mind and is delivered with a standard configuration that includes Oracle Linux operating system security hardening best practices. These practices include the following security objectives:

- Hardened Linux OS
- Hardened Unbreakable Enterprise Kernel (UEK)
- Vulnerability Mitigation

Access the Oracle Communications Application Session Controller System

Enabling Management Access

When you create one or more administrative users, the OS-E prompts for a username and password when anyone attempts to log in. Administrative users have read/write management access to the OS-E configuration file. Editing and saving the configuration file updates the OS-E configuration file named cxc.cfg. If desired, administrators can commit the configuration changes to the running OS-E configuration.

CLI Session

The following CLI session creates a user and password (with permissions) for management access across the entire OS-E system.

```
NNOS-E>config access
config access>config users
Creating 'users'
config users>config user "jane doe"
Creating 'user "jane doe"'
config user "jane doe">set password abcXYZ
 confirm: ****************
config user "jane doe">set permissions access permissions grant
Creating 'access\permissions grant'
config user "jane doe">return
config users>return
config access>config permissions grant
Creating 'permissions grant'
config permissions grant>set cms enabled-web-only
config permissions grant>set cli normal
config permissions grant>set call-logs enabled
config permissions grant>set actions enabled
config permissions grant>set status enabled
config permissions grant>set user-portal enabled
config permissions grant>set web-services enabled
```

If you are using the CMS to configure administrative users and permissions, use the CMS Access tab.

Configuring Management Options

There are four ways a user can access the Oracle Communications Application Session Controller system. The following image illustrates a sample network showing the supported management options.

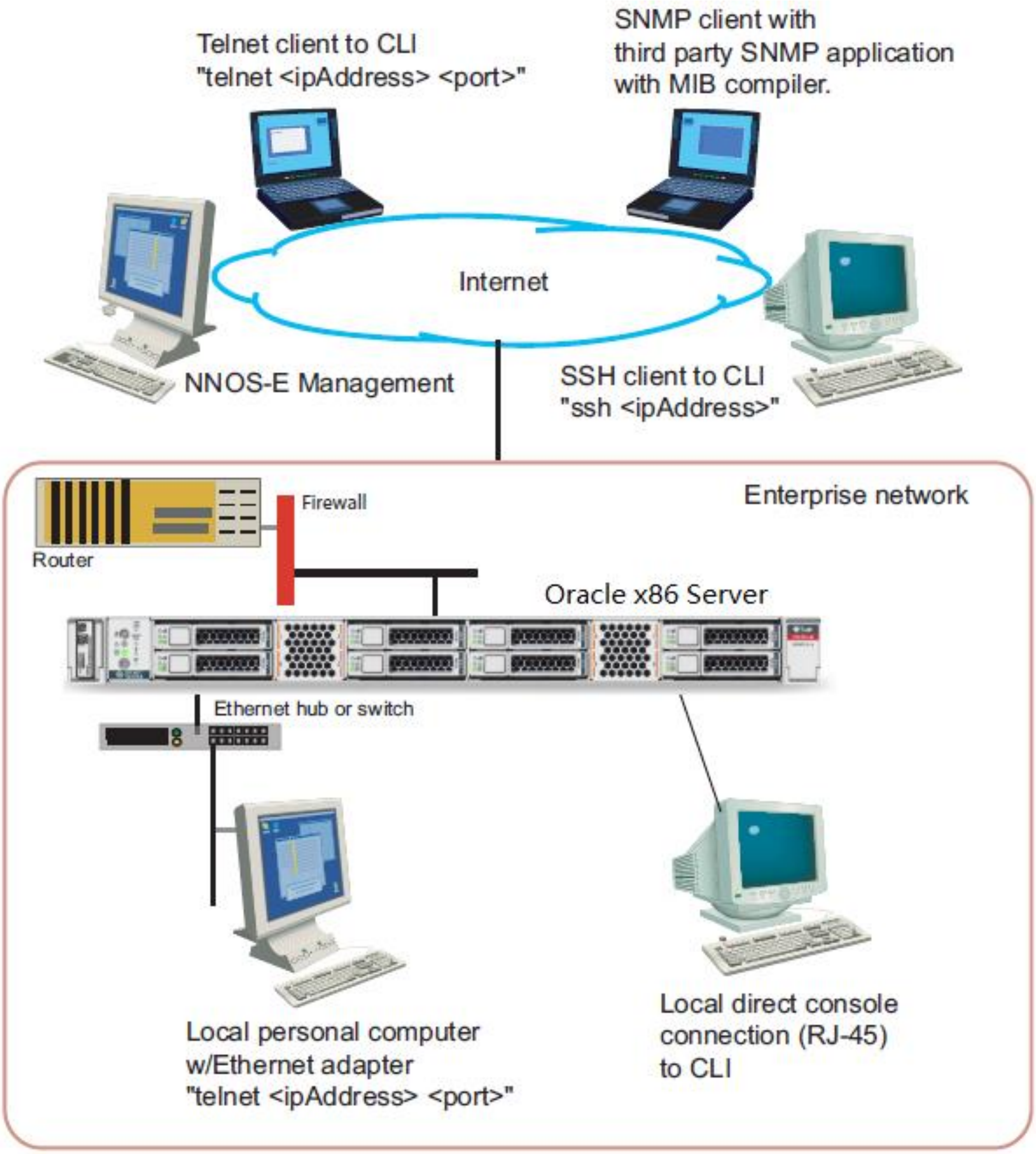


Figure 2: Oracle Communications Application Session Controller management options

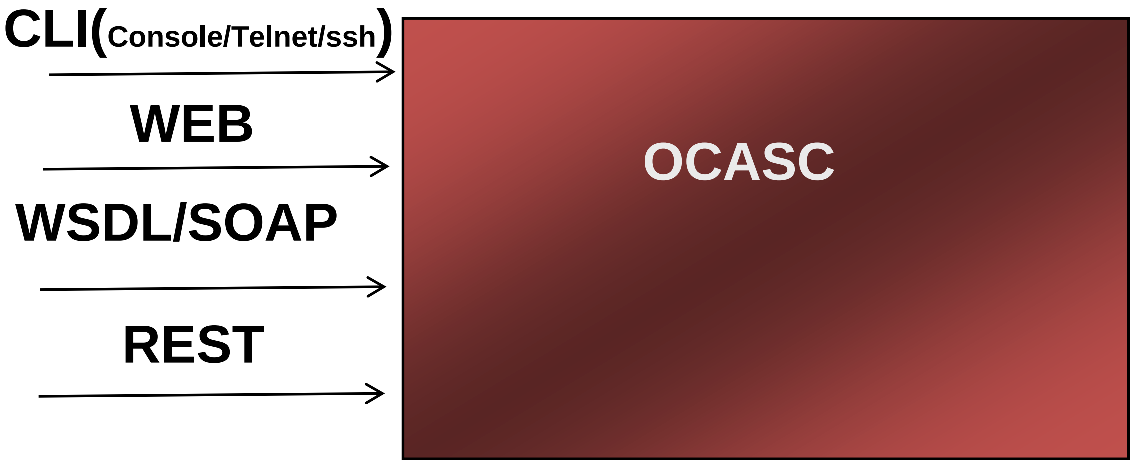


Figure 3: Ways to Access OCASC

Local Console

If you are using a directly-attached local console or terminal to configure the OS-E for the first time, use a terminal emulation program such as HyperTerminal to set the console parameters.

The following CLI session configures the console settings for communicating with the OS-E ssytem. The example session shows the console default settings.

CLI Session

```
NNOS-E >config box
config box>config console
config console>set rate 115200
config console>set data-bits 8
config console>set parity none
config console>set stop-bits 1
config console>set flow-control none
```

Telnet

Telnet is a standard TCP/IP-based terminal emulation protocol defined in RFC 854, Telnet Protocol Specification. Telnet allows a remote user to establish a terminal connection to the OS-E system over an IP network. By default, the Telnet protocol is enabled at installation time. To allow connections over Telnet, you must configure those users who are allowed access to the OS-E over Telnet.

The following CLI session configures the Telnet protocol on the local OS-E system, including the maximum number of concurrent Telnet sessions, the idle timeout period (in seconds) that ends a Telnet session due to inactivity, and the known TCP port for inbound and outbound Telnet messages.

CLI Session

```
NNOS-E>config box
config box>config interface eth0
config interface eth0>config ip local
Creating 'ip local'
config ip local>config telnet
config telnet>set admin enabled
config telnet>set max-sessions 10
config telnet>set idle-timeout 600
config telnet>set port 23
```

Secure Shell (SSH)

Secure Shell (SSH) Server Version 2 on the OS-E system provides secure client/server communications, remote logins, and file transfers using encryption and public-key authentication. To establish a secure connection and communications session, SSH uses a key pair that you generate or receive from a valid certificate authority (CA). By default, SSH is enabled at installation time.

An SSH session allows you to transfer files with Secure Shell File Transfer Protocol (SFTP), providing more secure transfers than FTP and an easy-to-use interface. SSH uses counters that record SFTP activity over the SSH connection.

When running SSH on the OS-E system, the SSH session is transparent and the CLI appears just as it would if you were connecting from a console or over Telnet. The OS-E implementation of SSH does not support all the user-configurable parameters typically supported by SSH workstations. If you try to change a parameter that the OS-E does not support, you will receive a notification that the parameter setting failed.

CLI Session

The following CLI session configures the SSH protocol on the local OS-E system, including the maximum number of concurrent SSH sessions, the idle timeout period (in seconds) that ends an SSH session due to inactivity, and the known TCP port for inbound and outbound SSH messages.

```
config box>config interface eth0
config interface eth0>config ip local
config ip 1ocal>config ssh
config ssh>set admin enabled
config ssh>set max-sessions 10
config ssh>set idle-timeout 600
config ssh>set port 22
```

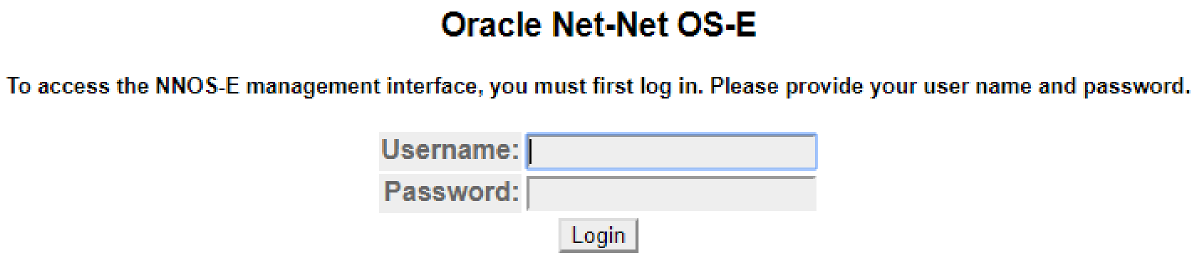
Web/HTTP

The OS-E Management System allows you to configure and manage the OS-E system remotely using your web browser.

The OS-E interface supports all management capabilities provided by the CLI. Instead of entering information on a command line, you navigate menus and supply information in menu fields.

To manage the OS-E system over the Web, enter the IP address of the management IP interface in the Internet Explorer File/Open command window and log in. For example: http://192.168.124.1/

It requires a web browser supporting a TLS1.0, TLS 1.1 or TLS 1.2 enabled session to Oracle Communications Application Session Controller. See Chapter 5. Installing Certificates and Commissioning TLS Networks in ASC3.8.0 Installation.



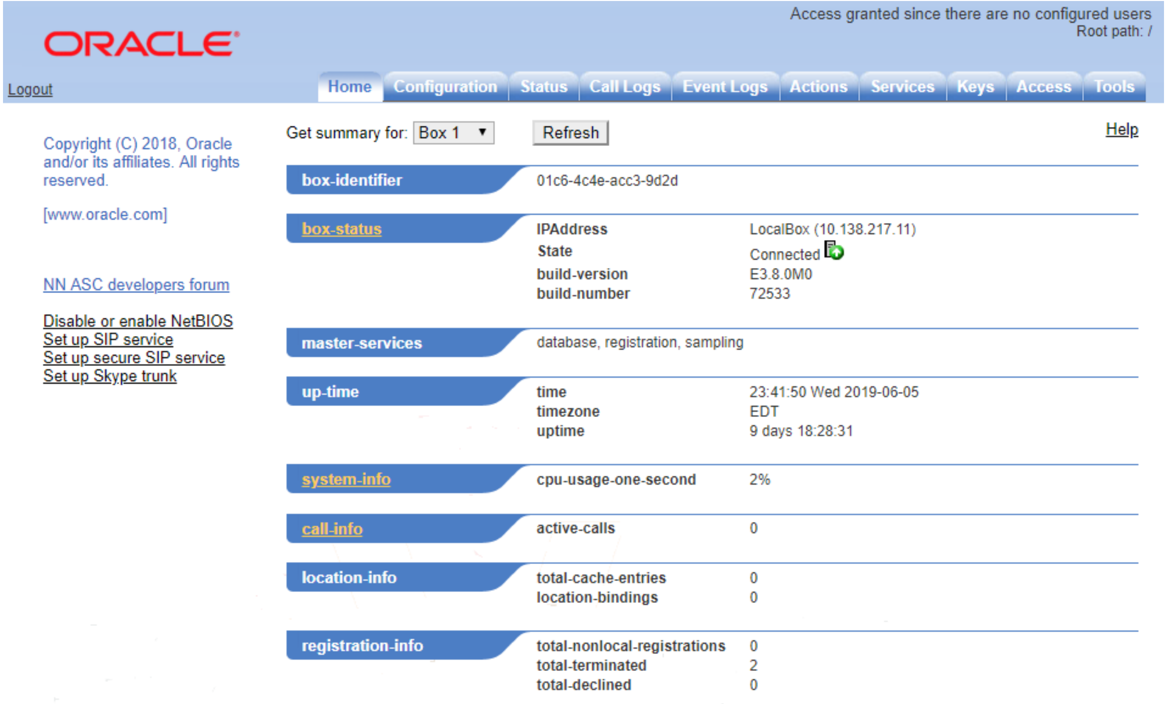
Oracle Net-Net OS-E

To access the NNOS-E management interface, you must first log in. Please provide your user name and password.

Username:

Password:

Figure 4: Oracle Communications Application Session Controller Login Page



Access granted since there are no configured users
Root path: /

Copyright (C) 2018, Oracle and/or its affiliates. All rights reserved.

[www.oracle.com]

NN ASC developers forum

Disable or enable NetBIOS
Set up SIP service
Set up secure SIP service
Set up Skype trunk

Get summary for: Box 1

box-identifier	01c6-4c4e-acc3-9d2d	
box-status	IPAddress State build-version build-number	LocalBox (10.138.217.11) Connected E3.8.0M0 72533
master-services	database, registration, sampling	
up-time	time timezone uptime	23:41:50 Wed 2019-06-05 EDT 9 days 18:28:31
system-info	cpu-usage-one-second	2%
call-info	active-calls	0
location-info	total-cache-entries location-bindings	0 0
registration-info	total-nonlocal-registrations total-terminated total-declined	0 2 0

Figure 5: Oracle Communications Application Session Controller Home Page

CLI Session

The following CLI session enables Web access to the local OS-E and specifies the TCP port over which HTTPS traffic is sent and received on the IP interface.

```
config box>config interface eth0
config interface eth0>config ip local
config ip local>config web
config web>set admin enabled
config web>set protocol https 443
```

SNMP

The Simple Network Management Protocol (SNMP) allows you to communicate with the SNMP agent on the OS-E system from a remote management station. SNMP allows you to retrieve information about managed objects on the platform as well as initiate actions using the standard and enterprise Management Information Base (MIB) files that Acme Packet makes available with the product software.

The OS-E supports the SNMP versions SNMP v1 and SNMP v2c.

CLI Session

The following CLI session enables SNMP access to the local OS-E system, specifies the TCP port over which SNMP traffic is sent and received on the management interface, sets the SNMP community string, the SNMP version, and the target system IP address to which SNMP trap messages are forwarded.

```
config box>config interface eth0
config interface eth0>config ip local
config ip local>config snmp
config snmp>set admin enabled
config snmp>set port 161
config snmp>set version 2c
config snmp>set community private
config snmp>set trap-target 192.168.124.10
```

HTTP\SOAP\WSDL Interface

The OS-E software includes a software development kit (SDK) to provide Web Services Description Language (WSDL) accessibility to the OS-E.

WSDL is an XML-based language for describing Web services, and how to access them, in a platform-independent manner. Simple Object Access Protocol (SOAP) is a communication protocol for communication between applications, based on XML.

A WSDL document is a set of definitions that describe how to access a web service and what operations it will perform. The OS-E uses it in combination with SOAP and an XML Schema to allow a client program connecting to a web service to determine available server functions. The actions and data types required are embedded in the WSDL file, which then may be enclosed in a SOAP envelope. The SOAP protocol supports the exchange of XML-based messages, with the OS-E using HTTPS.

The OS-E can perform two roles in the WSDL exchange:

- As a web service server, where an external client can make web service requests on the OS-E system.
- As a web service client, where the OS-E can make web service "call outs" to get location and policy information from an external service endpoint.

The WSDL document (and its imported schema files, such as cxc.xsd) define every possible request and response provided for the service, including error responses. Depending on how you choose to integrate with the OS-E system, you can use the OS-E SDK (using Java) or you can simply take the WSDL document and generate tools in your desired language. Because web services are language independent, you can use virtually any modern language to generate the requests and the WSDL document defines what those requests need to look like for the receiving component.

REST

REST is an API style supported by the ASC for web service which implements a URI using HTTP and a collection of resources with three defined aspects:

- The base URI for the web service.
- The format of the data returned by the REST URL. This is usually either XML or JavaScript Object Notation (JSON).
- A set of ASC web service operations.

There are two action and status report request formats available when using RESTful web service, flat and hierarchical. When possible, Oracle recommends using the hierarchical format, which is a simpler way to encode REST requests.

The default format returned by the REST URL is XML. However, you can request to receive the output in the JavaScript Notation (JSON) format instead by specifying this in the URI.

To specify the format in which you want the responses to REST requests:

output=<*response_format*>

Options are:

- json: Use JSON format
- xml: (default) Use XML format

If you choose JSON, the ASC supports Javascript callbacks when using REST. To configure a callback, specify the JavaScript method name in the URI. The JavaScript method is called with the JSON output string as a parameter.

callback=<*JavaScript_method_name*>

If you choose XML, you must specify an XML output format.

_format=<*xml_format*>

Options are:

- simplified
- legacy (default)

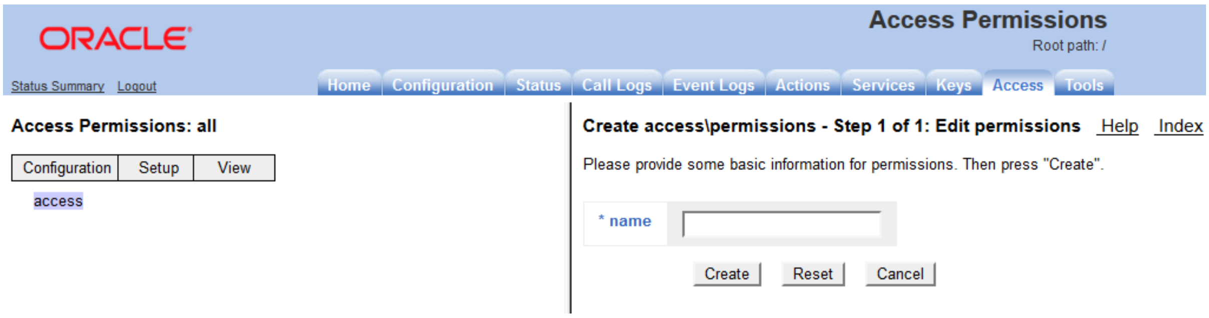
Configuring Permissions, Users, and Authorization

Configuring Permissions

Under the Access tab you can configure permissions. From this object you can enable or disable access to a variety of OS-E services. Once a permission set is created, it can be applied to configured users.

To create a permission set:

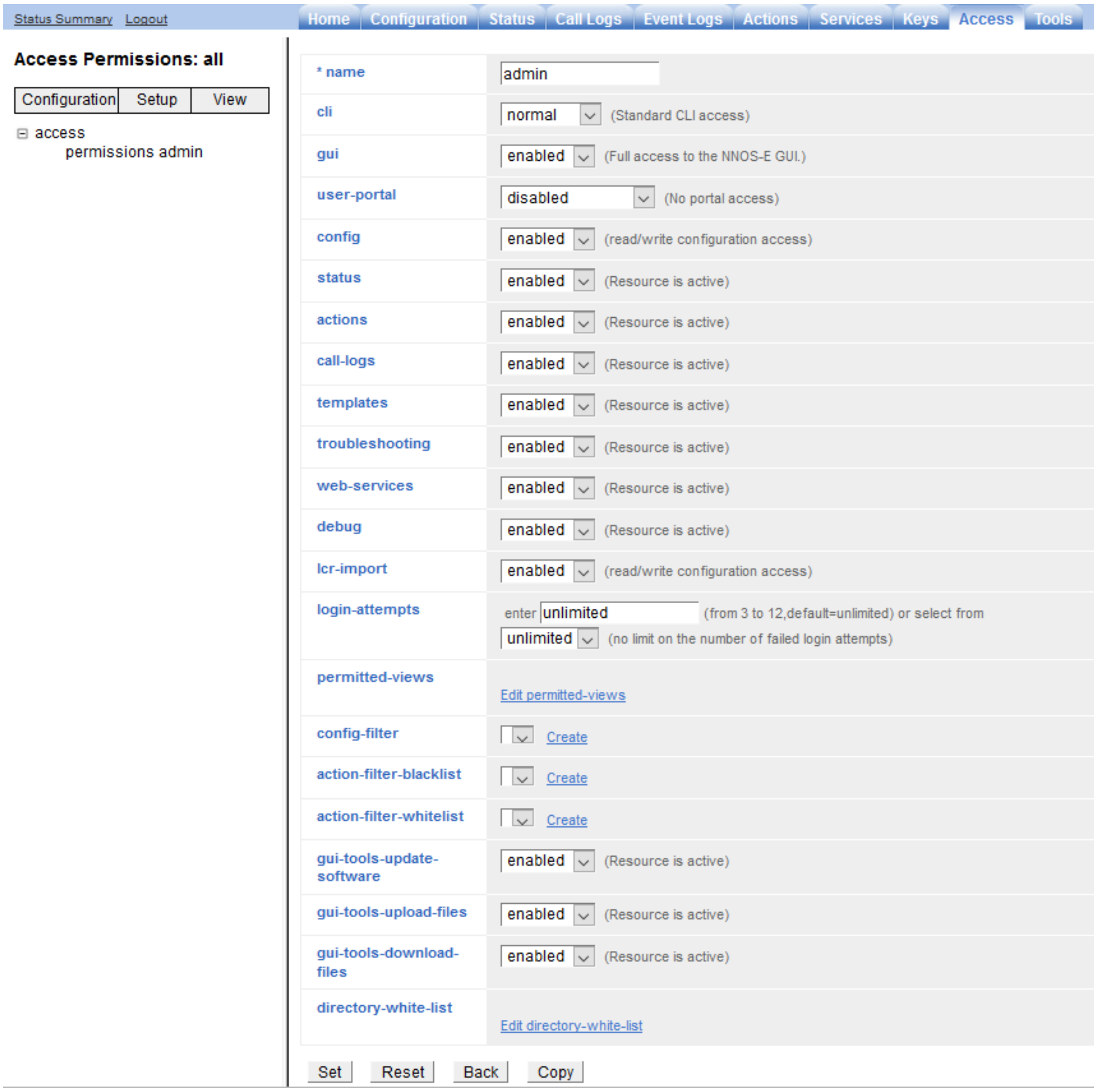
1. Select the Access tab and click Access.
2. Click Add permissions.
3. Enter the name you want to give this permission set and select Create.



Create access\permissions - Step 1 of 1: Edit permissions

Please provide some basic information for permissions. Then press "Create".

Figure 6: Create a permission set



Access Permissions: all

access
permissions admin

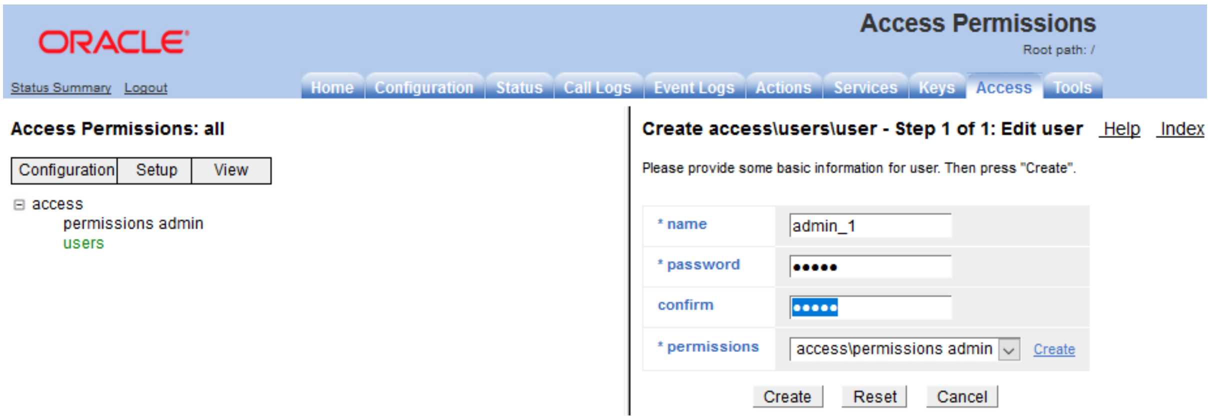
* name	admin
cli	normal (Standard CLI access)
gui	enabled (Full access to the NNOS-E GUI.)
user-portal	disabled (No portal access)
config	enabled (read/write configuration access)
status	enabled (Resource is active)
actions	enabled (Resource is active)
call-logs	enabled (Resource is active)
templates	enabled (Resource is active)
troubleshooting	enabled (Resource is active)
web-services	enabled (Resource is active)
debug	enabled (Resource is active)
lcr-import	enabled (read/write configuration access)
login-attempts	enter unlimited (from 3 to 12,default=unlimited) or select from unlimited (no limit on the number of failed login attempts)
permitted-views	Edit permitted-views
config-filter	Create
action-filter-blacklist	Create
action-filter-whitelist	Create
gui-tools-update-software	enabled (Resource is active)
gui-tools-upload-files	enabled (Resource is active)
gui-tools-download-files	enabled (Resource is active)
directory-white-list	Edit directory-white-list

Figure 7: A permission set named admin.

Configuring Users

Configure OS-E users using the Access tab’s users object.
When creating a user, you assign them a name, a password, and apply to them a configured permissions set.

To create a user:
1. Select the **Access** tab and click **users**.
2. **admin**—Set to enabled to allow configured users access to the OS-E.
3. Click **Add user**. The user object appears.



Access Permissions

Root path: /

Access Permissions: all

Create access\users\user - Step 1 of 1: Edit user

Please provide some basic information for user. Then press "Create".

* name	admin_1
* password	•••••
confirm	•••••
* permissions	access\permissions admin

Figure 8: Configure a user

4. **name**—Enter a name to give this user.
5. **password**—Enter a password for this user.
   Note: Via the **password-policy** object, you can specify password requirements for configured users.
6. **confirm**—Reenter the password.
7. **permissions**—Select a pre-configured permissions set to apply to this user from the drop-down list. If you have not configured permissions yet, click Create.
8. Click **Create**.
9. Click **Set**. Update and save the configuration.

Configuring Action and Config Filters

The OS-E supports filtering mechanisms which control which users have access to specific actions and configuration objects and properties. These filters are configured under the **access** > **permissions** object.

The three permission filters are:

- Config-filter
- Action-filter-blacklist
- Action-filter-whitelist

There are three steps necessary to assign action and configuration filters to configured users. You must create the filters, assign filters to permissions set, then assign each user a permission set.

Configuring Config-Filters

Via the **config-filter** property, you can select a config-filter containing a list of configuration objects and properties you want to restrict certain users from being able to access.

Config-filters have three permission levels.

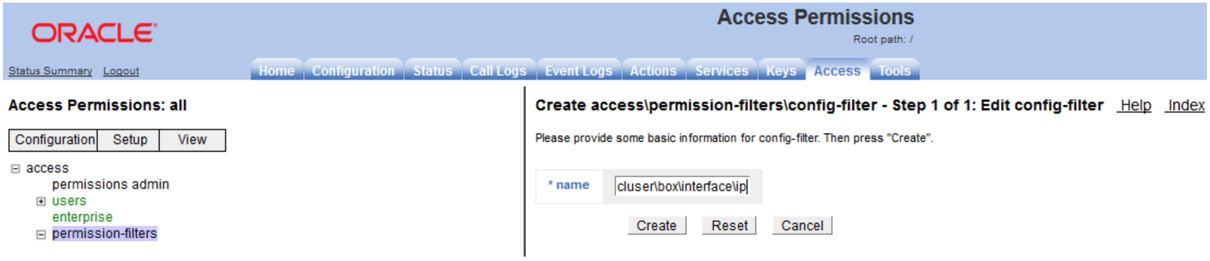
- read-write—Users can modify the configuration
- read-only—Users can view the configuration but cannot modify it
- none—Users can neither view nor modify the configuration

To configure a config-filter:

1. Select the **Access** tab and click **Access**.

2. Click **Configure** next to **permission-filters**.

3. Click **Add config-filter**.



Create access\permission-filters\config-filter - Step 1 of 1: Edit config-filter

Please provide some basic information for config-filter. Then press "Create".

* name: cluser\box\interface\ip

Figure 9: Configure a config-filter

4. name—Specify a name to give this config-filter.

5. Click **Create**. The **filter** object appears.

6. admin—Set to **enabled** to enable this config-filter.

7. Click **Add filter**.

8. filter—Specify a configuration object by entering the class, object, and property in free form, separating each with a back slash “\”.

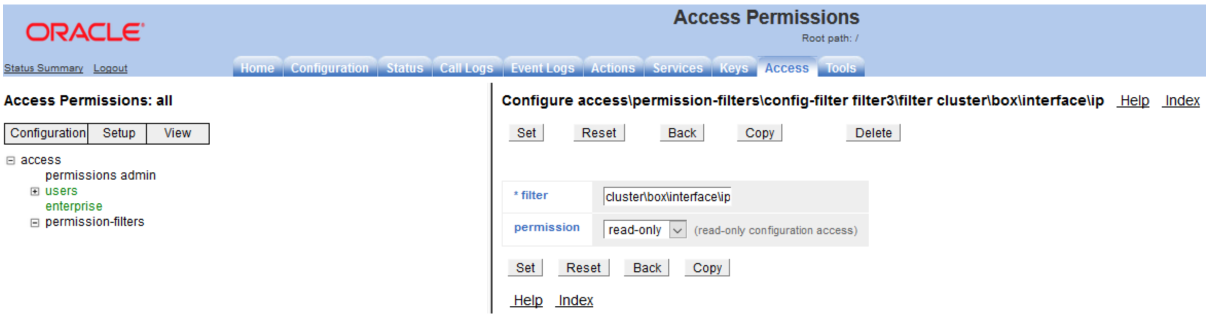
9. Click **Create**.

10. Repeat Steps 7 and 8 for as many configuration objects you want to apply to this filter.

11. Click **Set**. Update and save the configuration.

To specify a filter permission:

1. Click **Edit** next to the filter.



Configure access\permission-filters\config-filter filter3\filter cluster\box\interface\ip

* filter: cluster\box\interface\ip

permission: read-only (read-only configuration access)

Figure 10: Specify a filter permission

2. permission—Select the permission level for this filter from the drop-down list. This is set to **none** by default.

3. Repeat this for each filter.

4. Click **Set**. Update and save the configuration.

Configuring Action-Filters

Via the **action-filter-blacklist** property, you can select an action-filter containing a list of actions you want to restrict certain users from using. When a user attempts to execute a restricted action, he gets the following error message:

```
Insufficient permissions for user
```

Via the **action-filter-whitelist** property, you can select an action-filter containing a list of actions you want to allow certain users to use.

The action-filter-whitelist property supports the use of a wildcard. The wildcard is an asterisk (*) that can be located at the end of a string only. For example, to create an action-filter for all call-control actions, enter **call-control-***.

When action-filters are configured on the OS-E, the OS-E always checks the **action-filter-blacklist** settings first. If the action is found on the blacklist, the user is not allowed to use it.

If both the **action-filter-blacklist** and **action-filter-whitelist** are configured and an action does not appear on either list, the user is restricted from using the action.

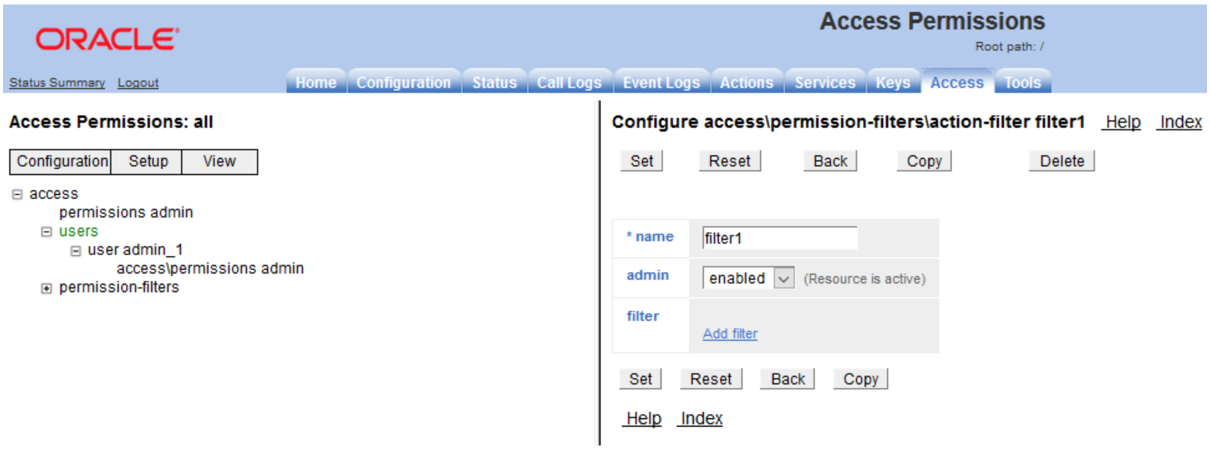
If an action is not found on the **action-filter-blacklist** and **action-filter-whitelist** is not configured, the user is allowed to use it.

> **Note:** You must enter actions into the **action-filter-blacklist** and **action-filter-whitelist** properties without any arguments. When anything more than an action name is specified, the OS-E ignores the filter.

To configure an **action-filter**:

1. Select the **Access** tab and click **Access**.
2. Click **Configure** next to **permission-filters**.
3. Click **Add action-filter**.
4. **name**—Specify a name to give this **action-filter**.
5. Click **Create**. The **filter** object appears.
6. **admin**—Set to **enabled** to enable this action-filter.
7. Click **Add filter**.
8. **filter**—Specify an action, without any arguments, to be applied to this filter.

   Note: If you enter an action with arguments, the action is ignored.
9. Repeat Steps 7 and 8 for as many actions you want to apply to this filter.
10. Click **Set**. Update and save the configuration.



Access Permissions: all

access
permissions admin
users
user admin_1
access\permissions admin
permission-filters

Configure access\permission-filters\action-filter filter1

* name: filter1
admin: enabled (Resource is active)
filter

Figure 11: Configure action-filters

Applying Filters to Permissions Sets

Once you have created config-filters and action-filters, you must apply them to a permission set.

To apply config-filters and action-filters to a permissions set:

1. Select the **Access** tab and click **Access**.

2. Click **Add permissions** to create a new permissions set or click **Edit** next to an existing permissions set.

3. config-filter—Select a config-filter from the drop-down list whose configuration objects you want to restrict users with this permissions set from using. If you have not yet created a config-filter, click Create next to this property.

4. action-filter-blacklist—Select an action-filter from the drop-down list whose actions you want to restrict users with this permissions set from using. If you have not yet created an action-filter, click Create next to this property.

5. action-filter-whitelist—Select an action-filter from the drop-down list whose actions you want to allow users with this permissions set to use. If you have not yet created an action-filter, click Create next to this property.

6. Click **Set**. Update and save the configuration.

Once you have configured config-filters and action-filters and applied them to a permissions set, you can assign the permissions set to users. For more information on applying permissions set to users, see Configuring Users.

Configuring Authorization

Once you have configured permission sets and users, you can further define user access by configuring authorization. Authorization consists of creating specific grants, or privileges.

There are three types of grants you can create:

- default-grants—Applies to all configured OS-E users
- attribute-grants—Applies to configured OS-E users based on values extracted from their attributes.
- group-grants—Applies to configured OS-E users based on group membership

The grants you can create apply to just a small segment of actions, which are divided into groups called resource-types. A resource-type is the OS-E function on which you are setting permissions.

The following table lists the resource types along with their corresponding actions.

Table 1: The resource types along with their corresponding actions

Resource-Type	Associated Actions	CRUD Privileges
Call	call-control-accept	
	call-control-annotate	
	call-control-attach	CU
	call-control-call	C
	call-control-call-to-session	CU
	call-control-connect	
	call-control-create-session	C
	call-control-destroy-session	D
	call-control-detach	D
	call-control-disconnect	D
	call-control-fork	U
	call-control-get-annotation	U
	call-control-hold	U
	call-control-info-request	U
	call-control-intercept	U
	call-control-join	U
	call-control-message-request	U
	call-control-modify	U
	call-control-mute-off	U
	call-control-mute-on	U
	call-control-notify	U
	call-control-notify-request	U
	call-control-options-request	U
	call-control-park	CU
	call-control-park-to-session	CU
	call-control-persistence	U
	call-control-record-stop	C
	call-control-redirect	U
	call-control-reject	U
	call-control-retrieve	U
	call-control-ringing	U
	call-control-send-message	U
	call-control-subscribe-request	U
	call-control-terminate	D
	call-control-transfer	U
call-recording	call-control-record-start	C
	call-control-record-stop	C
call-monitor	call-control-monitor-file	CU
	call-control-monitor-session	CU
call-media-insertion	call-control-drop-file	CU
	call-control-insert-dtmf	U
	call-control-media-pause	CU

Resource-Type	Associated Actions	CRUD Privileges
	call-control-media-resume	CU
	call-control-media-scanner-start	CU
	call-control-media-scanner-stop	CU
	call-control-media-seek	CU
	call-control-media-stop	CU
	call-control-memo-begin	CU
	call-control-memo-end	CU
	call-control-play	U
sip-request	sip-send-message	CU
	sip-send-notify	CU
	sip-send-options	CU
	sip-send-other	CU
	sip-send-subscribe	CU
	sip-send-unsubscribe	CU
registration	register	C
	unregister	D
event-channel	dynamic-event-service register	CR
	dynamic-event-service keepalive	U
	dynamic-event-service unregister	D

In cases where an action has required either <*handle*> or <*session ID*> arguments, the OS-E extracts the To and From URI identities from each call leg, matches them against the resource-identity specified in a user's privileges, and determines whether that user is authorized to perform an operation.

When configuring a grant, you must define privileges for that resource-type. Privileges specify what a user can or cannot do with that resource-type.

Privileges on the OS-E follow the standard CRUD model:

- create
- retrieve
- update
- delete

Configuring Default Grants

Configure grants under the Access tab's **authorization** object.

Default grants are one of three types of grants you can configure on the OS-E. Default grants are grants that apply to all OS-E users matching the specified resource identity.

To configure default grants:

1. Select the **Access** tab and click **authorization**.
2. Set **admin** to **enabled** to enable authorization.
3. Click **Add default-grant**. The **default-grant** object appears.
4. **name**—Enter a name to give this grant.
5. **resource-identity**—Select the type of matching to use to identify a resource-type. The following are valid values:

- equals <*value*>—The value that a user provides during an authorization request must be exactly the same as the resulting resource-identity. This is the default setting.
- matches <*expression*>—The value that a user provides during an authorization request is matched against the resource-identity using a regular expression match.

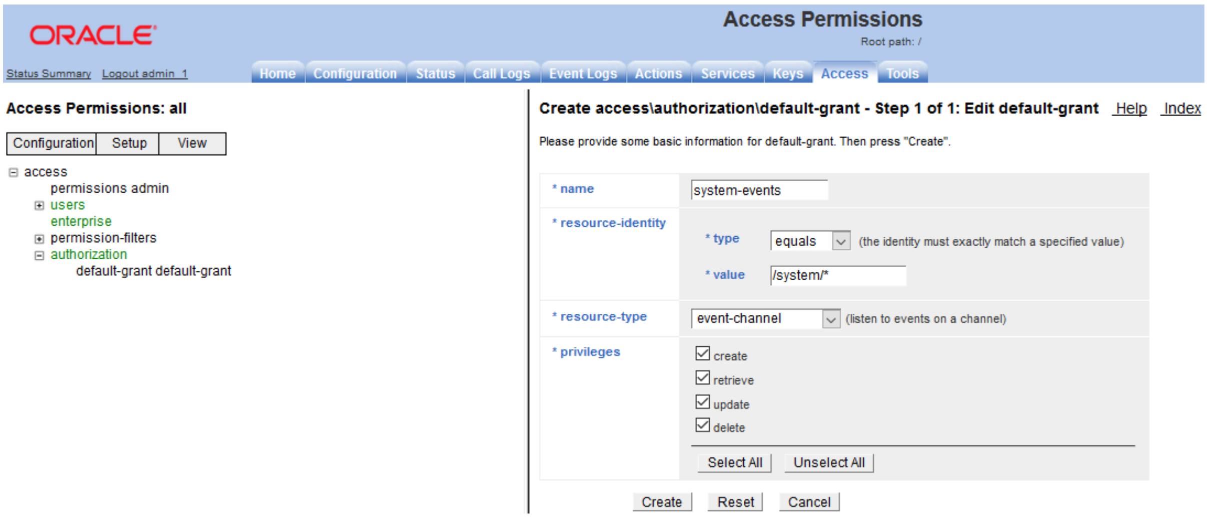
  Note: For more information on using Regular Expressions, see the Oracle Communications OS-E Objects and Properties Reference Guide.
- any—Any value a user provides during an authorization request matches.

6. resource-type—Select the resource-type for this grant from the drop-down list.

7. privileges—Check the CRUD privileges to allow for this resource-type. By default, they are all selected.

8. Click **Create**.

9. Click **Set**. Update and save the configuration.



Access Permissions

Root path: /

Access Permissions: all

⊟ access
  permissions admin
  ⊞ users
  enterprise
  ⊞ permission-filters
  ⊟ authorization
    default-grant default-grant

Create access\authorization\default-grant - Step 1 of 1: Edit default-grant

Please provide some basic information for default-grant. Then press "Create".

* name: system-events

* resource-identity
* type: equals (the identity must exactly match a specified value)
* value: /system/*

* resource-type: event-channel (listen to events on a channel)

* privileges
☑ create
☑ retrieve
☑ update
☑ delete

Figure 12: Configure default grants

Configuring Attribute Grants

Attribute grants are grants that apply to all OS-E users that have the attribute and match the specified resource-identity.

To configure attribute-grants:

1. Select the **Access** tab and click **authorization**.

2. name—Enter the name of the attribute for which you are creating this grant.

Note: The name you provide must be the name of an actual attribute used within the directory.

3. Click **Create**. The **attribute-grant** object appears.

4. Click **Add grant-pattern**.

5. name—Enter a descriptive name to give this grant.

6. pattern—Enter the regular expression pattern to use to define the attribute.

7. resource-identity—Select the type of matching to use to identify a resource-type. The following are valid values:

- equals <*value*>—The value that a user provides during an authorization request must be exactly the same as the resulting resource-identity. This is the default setting.
- matches <*expression*>—The value that a user provides during an authorization request is matched against the resource-identity using a regular expression match.

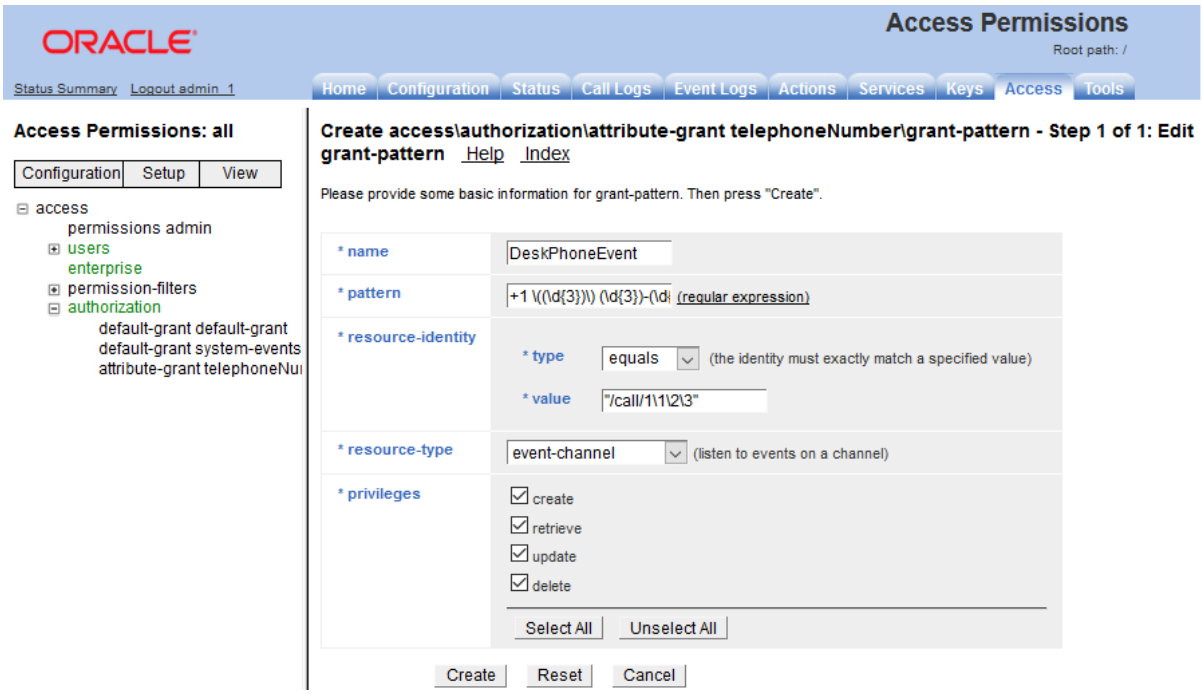
  Note: For more information on using Regular Expressions, see the Oracle Communications OS-E Objects and Properties Reference Guide.
- any—Any value a user provides during an authorization request matches.

8. resource-type—Select the resource-type that this extracted value represents from the drop-down list.

9. privileges—Check the CRUD privileges to allow for this resource-type. By default, they are all selected.

10. Click **Create**.

11. Click **Set**. Update and save the configuration.



Create access\authorization\attribute-grant telephoneNumber\grant-pattern - Step 1 of 1: Edit grant-pattern

Please provide some basic information for grant-pattern. Then press "Create".

* name: DeskPhoneEvent

* pattern: +1 \((\d{3})\) (\d{3})-(\d (regular expression)

* resource-identity

  * type: equals (the identity must exactly match a specified value)

  * value: "/call/1\1\2\3"

* resource-type: event-channel (listen to events on a channel)

* privileges: ☑ create ☑ retrieve ☑ update ☑ delete

Figure 13: Configure attribute-grants

Configuring Group Grants

Under the **group-grant** object, you can configure default and attribute grants for specific groups. Group grants apply to users belonging to these groups and matching the resource-identity.

To add a group-grant:

1. Select the **Access** tab and click **authorization**.

2. Click **Add group-grant**.

3. name—Enter the name of the group for which you are configuring this grant.

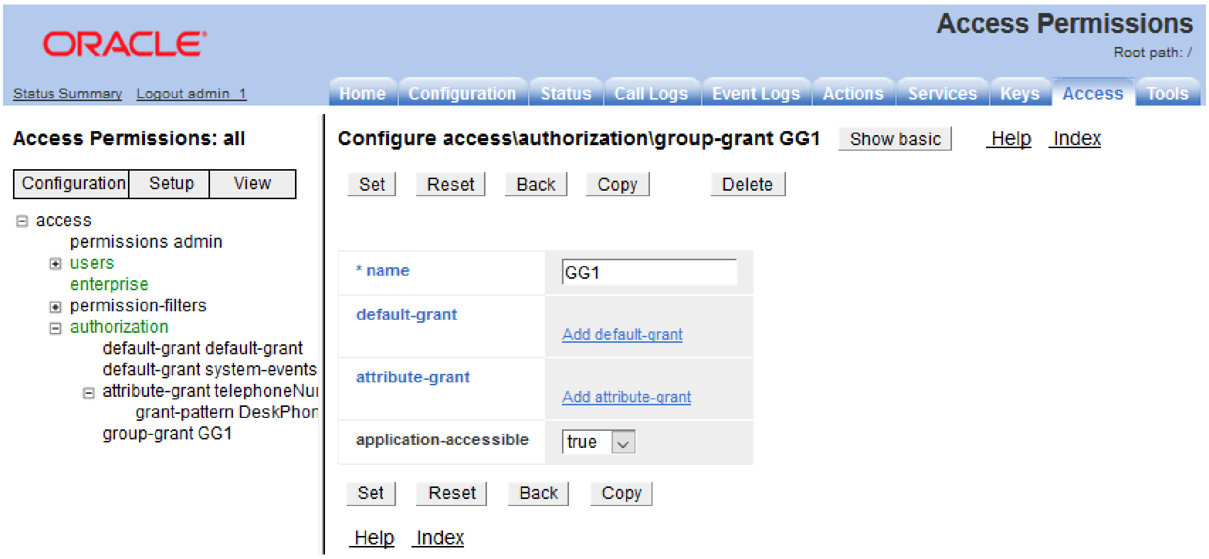
4. Click **Create**. The **group-grant** object appears.

5. Click **Add default-grant** to configure a default grant for this group or click **Add attribute-grant** to configure an attribute grant for this group.

6. Configure the default or attribute grant as described above.

   Note: For more information on configuring **default-grants** see Configuring Default Grants. For more information on configuring **attribute-grants** see Configuring Attribute grants.

7. Click **Set**. Update and save the configuration.



Configure access\authorization\group-grant GG1

* name	GG1
default-grant	Add default-grant
attribute-grant	Add attribute-grant
application-accessible	true

Figure 14: Configure group-grants

Viewing User Privilege Information

There are three show commands which allow you to view information on your grant configuration: **show authorized-user-privileges**, **show authorized-user-attributes**, and **show authorized-user-groups**.

The **show authorized-user-privileges** action displays information about users' authorization privileges from the user cache.

Note: If a user has never logged into the OS-E, their name does not appear in the cache and, therefore, is not displayed in the **show authorized-user-privileges** command output.

```
NNOS-E>show authorized-user-privileges
username resource-type privilege identity-type resource-identity
-------- ------------- --------- ------------- -----------------
admin    event-channel C+R+U+D   equals         /system/*
```

Table 2: Description of the show authorized-user-attributes action

Field	Description
username	The name of the configured OS-E user.
resource-type	The resource-type of the grant configured for this user.
privilege	The CRUD privileges of the of the resource-type configured for this user.
identity-type	The method in which the OS-E matches the users' resource-identity.
resource-identity	The value or regular expression the OS-E uses to check users' authorization privileges.

The **show authorized-user-attributes** action displays information about configured OS-E users and their attributes and values.

```
NNOS-E>show authorized-user-attributes
username attribute value
-------- --------- -----
sjones mail sjones@acmepacket.com
sjones msrtcsip-primaryuseraddress sip:sjones@acmepacket.com
sjones cn Sam Jones
sjones samaccountname sjones
sjones msrtcsip-line tel:+17815557256
sjones st MA
sjones telephonenumber +1 (781) 555-4839
```

Table 3: Description of the show authorized-user-attributes action

Field	Description
username	The configured OS-E user.
attribute	The attribute name.
value	The value of the attribute for that user.

The **show authorized-user-groups** action displays the configured users and the groups to which they belong from the user cache.

```
NNOS-E>show authorized-user-groups
username group
-------- -----
sjones eng
sjones software
sjones dev
sjones ct
sjones engineering
sjones deliveries
sjones funcspec
```

Table 4: Description of the show authorized-user-summary action

Field	Description
username	The configured OS-E user.
group	The group to which the user belongs.

The **show authorized-user-summary** action displays an abbreviated version of users' authorization privileges from the user cache.

```
NNOS-E>show authorized-user-summary
username resource-types
-------- --------------
admin event-channel
test_user event-channel
```

Table 5: Description of the show authorized-user-summary action

Field	Description
username	The name of the configured OS-E user.
resource-type	The resource-type of the grant configured for this user.

Understanding the OCASC Environment

When planning your OCASC implementation, consider the following:

- **Which resources need to be protected?**
    - You need to protect customer data, such as credit-card numbers.
    - You need to protect internal data, such as proprietary source code.
    - You need to protect system components from being disabled by external attacks or intentional system overloads.
- **Who are you protecting data from?**

  For example, you need to protect your subscribers’ data from other subscribers, but someone in your organization might need to access that data to manage it. You can analyze your workflows to determine who needs access to the data; for example, it is possible that a system administrator can manage your system components without needing to access the system data.
- **What will happen if protections on a strategic resources fail?**

  In some cases, a fault in your security scheme is nothing more than an inconvenience. In other cases, a fault might cause great damage to you or your customers. Understanding the security ramifications of each resource will help you protect it properly

Operating System Security

See the following document:

- Oracle® Linux Security Guide for Release 7

Chapter 2: Performing a Secure OCASC Installation

This chapter presents planning information for your OCASC installation.

For information about installing Oracle Communications Application Session Controller, see OCASC Installation Guide.

Installing OCASC Securely

You can perform a custom installation or a typical installation. Perform a custom installation to avoid installing options and products you do not need. If you perform a typical installation, remove or disable features that you do not need after the installation.

Configuring the External Firewalls

The **near-side-nat** object allows you to configure the OS-E system to perform Network Address Translation (NAT) on SIP traffic that traverses the enterprise firewall between the OS-E system and the Internet. By configuring the IP address of the public-facing interface on the enterprise firewall, the OS-E produces a contact header that replaces enterprise private IP addresses with the public-facing firewall address.

NAT, defined in RFC 1631, *The IP Network Address Translator*, ensures that internal private network addresses are rewritten so that they appear to come from the designated external network firewall address. The OS-E modifies outgoing packets so that the return address is a valid Internet host (the external firewall). The firewall then changes the destination address on incoming packets to the OS-E system's private address.

SIP traffic that matches the configured UPD and TCP port ranges will use NAT so that only the public-side IP address can be observed by remote SIP users across the Internet. The following image illustrates a sample network with showing the public-side IP address 204.124.1.50.

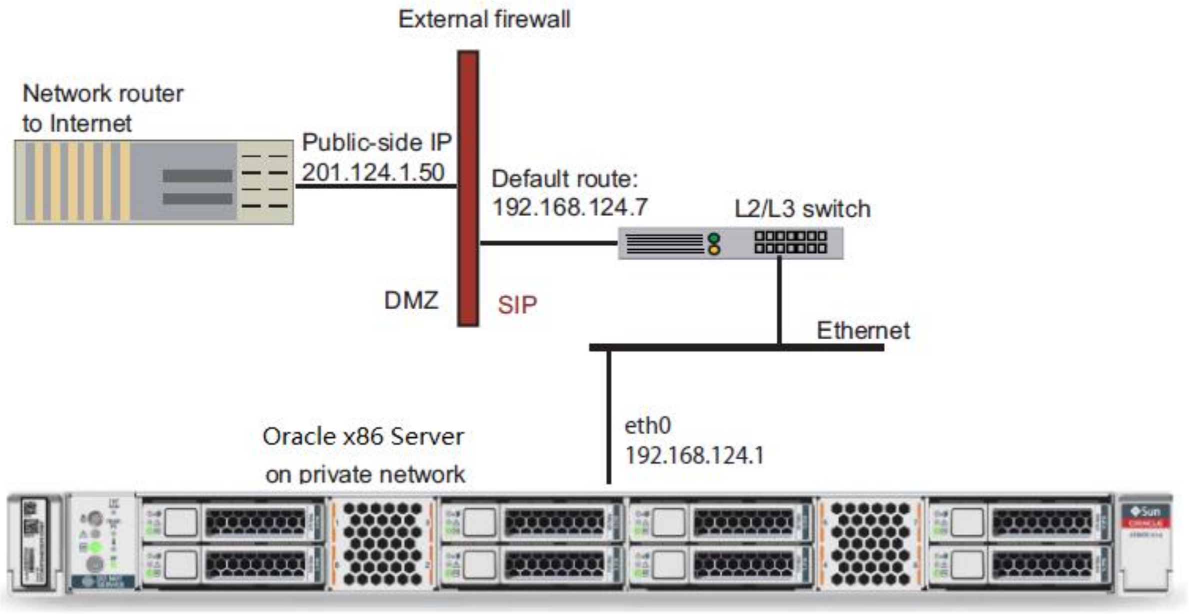

External firewall
Network router to Internet
Public-side IP 201.124.1.50
Default route: 192.168.124.7
L2/L3 switch
DMZ
SIP
Ethernet
Oracle x86 Server on private network
eth0 192.168.124.1

Figure 15: A sample network

CLI Session

The following CLI session configures the external firewall public IP address to 201.124.1.50 and sets the UDP and TCP port ranges over which to listen for SIP messages and IP address replacement.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth#> config ip boston1
config ip boston1> config near-side-nat 201.124.1.50
Creating 'near-side-nat 201.124.1.50'
config near-side-nat 201.124.1.50> set admin enabled
config near-side-nat 201.124.1.50> set public-ip 201.124.1.50
config near-side-nat 201.124.1.50> set udp-range 5060 1
config near-side-nat 201.124.1.50> set tcp-range 5060 2
```

You typically configure UDP port 5060 and TCP ports 5060 and 5061 for SIP traffic. Be certain that the port numbers you enter here are the same as those you configured in the **ip sip** object, as described in this chapter.

In addition, you may configure the NAT pool addresses within this object, typically UDP ports 20000 through 30000.

Configuring the STUN, TURN, and ICE Protocols

The OS-E, as a STUN server, uses three protocols that operate together to handle SIP signaling and media traversal across NAT routers and firewalls. These protocols are:

- STUN—Simple Traversal of User Datagram Protocol Through Network Address Translators
- TURN—Traversal Using Relay NAT
- ICE—Interactive Connectivity Establishment

The OS-E system implements draft-ietf-behave-rfc3489bis-04 (for STUN, in addition to RFC3489), and draft-ietf-behave-turn-01, both released in July 2006.

For complete information on STUN and TURN refer to:

- RFC 3489—STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)
- draft-ietf-behave-rfc3489bis-04—Simple Traversal Underneath Network Address Translators (NAT) (STUN)
- draft-ietf-behave-turn-01—Obtaining Relay Addresses from Simple Traversal of UDP Through NAT (STUN)

STUN

Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) (STUN), described in RFC 3489, enables SIP clients to discover the presence and types of NATs and firewalls that exist between them and the public Internet. A STUN server and receives and transmits UDP messages over UDP port 3478 (default). The STUN protocol helps prevent NAT-associated network application failures by transmitting exploratory STUN messages over UDP between the server and clients.

STUN identifies the public side NAT details by inspecting exploratory STUN messages that arrive at the STUN server. The STUN-enabled client sends an exploratory message to the STUN server to determine the transmit and receive UDP port to use. The STUN server examines the incoming message and informs the client which public IP address and ports were used by the NAT. These are then used in the call establishment messages sent to the SIP destination server.

For complete information on STUN, refer to *RFC 3489 -STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs).*

Traversal Using Relay NAT (TURN)

Depending on the network topology and the NAT implementation, IP addresses obtained by STUN may not be usable by all peers. A client must be able to obtain a publicly visible transport address that can receive media from any peer that can send packets to the public Internet. This is done by relaying data though a server that resides on the public Internet.

The Traversal Using Relay NAT (TURN) protocol allows a client to obtain a transport address from a relay,

1. To send traffic to the peer through that address

2. To receive all traffic sent to that address by the peer

The **relay-interface** property specifies the interface over which the SIP client receives public visibility, as well as the interface from which the OS-E system allocates TURN relay ports. This interface must have **media-ports** enabled and a port pool range defined.

Interactive Connectivity Establishment (ICE)

Both STUN and TURN work in conjunction with the Interactive Connectivity Establishment (ICE) protocol to determine what type of NAT firewalls exist between SIP clients and to determine a set of “candidate” transport addresses by which they are able to establish contact. STUN and TURN are sometimes used by ICE.

An ICE-enabled client (the initiator) uses STUN and TURN, and a locally configured policy, to determine a prioritized list of candidate addresses before sending this list to the responder client in the SDP portion of a SIP message. When the responder client receives this message, it performs a “connectivity check” on each candidate address by sending a STUN request to that address and waiting for a reply. The highest priority candidate to pass the connectivity check is then used for the actual connection.

The OS-E system does not require any ICE-specific configuration.

Sample Configuration

For STUN to operate properly, follow these rules when configuring STUN servers:

- Create STUN server instances in pairs (for compliance with the RFC 3489).
- Put each instance of the pair on a different IP address.
- Assign exactly two UDP ports to each; the port number assignments must be identical for each.

- The secondary interface of each STUN server instance must point at the IP address of the other STUN server instance.

  For example, with a STUN server configured on interface A, ports 100 and 200, configure an additional STUN server on interface B, ports 100 and 200. In the interface A configuration, set the **secondary-interface** property to B, and vice versa.

The following CLI session configures a STUN server on the OS-E system.

CLI Session

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip a
config ip a> config stun-server
config stun-server> set admin enabled
config stun-server> set stun-auth-level allow
config stun-server> set port 3478
config stun-server> set allow-turn enabled
config stun-server> set relay-interface "cluster box 1 interface eth1 ip
                     abc"
config stun-server> set secondary-interface "cluster box 1 interface eth0 ip
                    b"
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip b
config ip b> config stun-server
config stun-server> set admin enabled
config stun-server> set stun-auth-level allow
config stun-server> set port 3478
config stun-server> set allow-turn enabled
config stun-server> set relay-interface "cluster box 1 interface eth1 ip
                    abc"
config stun-server> set secondary-interface "cluster box 1 interface eth0 ip
                    a"
```

For more information, and for information on setting all stun-server properties, refer to the *Net-Net OS-E – Objects and Properties Reference*.

Configuring Kernel Filtering

Kernel filter rules provide a security mechanism that allows or denies inbound traffic on OS-E IP interfaces. The filter controls access to resources on the enterprise servers based on source IP address and/or subnet, source port, and protocol. When the OS-E processes kernel rules, it first interprets deny rules, then allow rules. In this way, you can deny a subnet access, and then allow specific endpoints.

The OS-E acts on kernel rules before the other, higher level rules such as DOS policy rules. This stops traffic from known problems early, tying up fewer processing resources.

CLI Session

The following CLI session creates and enables a deny rule named *evil-badguy* from source IP address 215.200.40.8, source port 56, over UDP.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
config ip boston1> config kernel-filter
config kernel-filter> config deny-rule evil-badguy
Creating 'deny-rule evil-badguy'
config deny-rule evil-badguy> set admin enabled
config deny-rule evil-badguy> set source-address/mask 215.200.40.8/24
config deny-rule evil-badguy> set source-port 56
config deny-rule evil-badguy> set protocol udp
```

Configuring Messaging

Messaging is the mechanism from which the OS-E system communicates with other systems in the cluster. Messaging sets up a listening socket on an interface, enabling the interface to receive messaging traffic and participate in clustering and media partnering.

In a cluster, the master looks through the configurations of all OS-E systems to find out which interface is used for messaging. (If multiple interfaces are configured, the master only communicates with one—the first it finds.) The master then communicates with the identified interface to share configuration and data.

In media partnering, you configure a specific IP address (on a different box) as a partner. On the box that owns that IP address, you need to configure and enable messaging for media partnering to operate.

CLI Session

The following CLI session configures messaging on box 1, interface eth0.

```
NNOS-E> config cluster
config cluster> config box 1
config box 1> config interface eth0
config interface eth0> config ip boston1
config ip boston1> config messaging
config messaging> set admin enabled
config messaging> set certificate vsp tls certificate name
config messaging> set port 13002
config messaging> set protocol tls
```

For detailed information on OS-E clusters and media partnering, refer to the *Net-Net OS-E – System Installation and Commissioning Guide*.

Post-Installation Tasks

This section explains security configuration to complete after OCASC is installed.

Changing Default Passwords

OCASC is installed with default passwords. Change those passwords as soon as possible.

Chapter 3: Implementing OCASC Security

This chapter explains the OCASC security features.

Denial of Service Policies

This section describes how you can perform Denial of Service (DOS) queries to the OS-E database, and how to create policies that prevent DOS attacks to the OS-E system.

Denial of Service Prevention Overview

A Denial of service (DOS) attacks is usually a flood of meaningless network traffic from a sender who intends to disrupt or totally disable services at a network destination. The OS-E system provides transport-layer, SIP-message, and URL policy definitions to detect DOS attacks. Queries allow you to sort and view incoming and outgoing traffic to better define policies. Policies determine if a packet is attacking the OS-E, and if so, the configured action is applied to that traffic. These tools quickly identify and shutout useless traffic, limiting any damage that might be caused by DOS attacks.

The OS-E uses the integrated database to record all packets that are transmitted or received by the system. The records are stored in specific tables that the OS-E can then access for queries and for policy execution. Activities through the transport layer, such as file transfers and SNMP walks, are stored in the transport layer table. The transport engine accesses that table for transport-level queries and policies. For each entry, the table records the following information from the TCP header:

- Remote IP address
- Remote port
- Local port
- Protocol

The SIP table contains entries for all SIP-related activities. The table includes all TCP-header information, as well as fields in the SIP header.

DOS Policies

The DOS transport, SIP, and URL policies use condition list setting to determine the point at which activity is considered to be part of a DOS attack and what action is to be taken. The action taken by the OS-E depends on whether the attack was identified by the transport, SIP, or URL policy. A policy fires at the frequency defined by the **period** property, scanning the database over the course of the last period looking for matches to the policy.

Transport Policy Operations

Transport policies are based on the TCP, UDP, and IP headers. The OS-E keeps a count of each time a policy match occurs. When the count exceeds the allowable threshold set in the condition list, the OS-E creates a dynamic rule, based on the criteria that resulted in a DOS attack declaration, in the kernel filter. Packets matching the pattern defined in the filter are dropped. The kernel rule keeps a count of dropped packets for later comparison.

The advantage to transport filters is that packets are dropped as soon as they enter the box, requiring very little processing. The disadvantage is that the filter is only set based on TCP header fields, which limits the flexibility. For more detailed policy, you should set SIP policy.

SIP Policy Operations

The SIP policy operates similarly to transport policy, but operates on the SIP header. The OS-E maintains a record of each hit to the policy, and the statements of the condition list define when that data constitutes an attack. Unlike the transport filter, however, the SIP policy comparison is executed in the application layer of the OS-E software. With the SIP policy, you configure the action to take as a result of exceeding the threshold settings. Because you have many more fields to include as part of your filter, you have greater granularity in the filter design. The upper-layer processing, however, requires more CPU cycles.

URL Policy Operations

The URL policy operates on URL regular expressions. All action is taken based solely on the URL. When a URL meets the criteria defined, the implicit action is to drop the packet(s). The URL policy detects when the same URL has gone through the OS-E a specified number of times over a specified number of seconds. When the OS-E detects excessive appearances of a URL (for example, someone SPIMing your network with an ad for their Web site or a virus that self-propagates via links in IMs), it blocks future IMs containing that same URL, regardless of who the IMs appear to come from.

How DOS Policy Object Properties Work Together

When configuring a DOS policy, you:

1. Define the filter criteria with a condition list

2. Select the columns that you want to observe with a select statement.

The sort criteria are applied to the entire message database. Using one or more set condition statements, you define which packets will be considered further. For example, if your condition statement sets remote-ip to match 1.2.3.4. and sets the local-port equal to radius, all packets that originated from IP address 1.2.3.4 and are destined for the RADIUS port, are copied into a new table.

Select the columns from the header fields from which you want your final result set built from. So from the table created above, you may want to compare the local-ip and remote-ip to determine the count of potentially questionable packets. It is the result of this final compare that is measured against the threshold you set to determine the next action.

The following example is an aggregation created at the transport level from selecting remote-ip, remote-port, and protocol. (DOS SIP policy works in the same manner.)

The resulting table looks as follows:

Table 6: An aggregation created at the transport level

Timestamp	remote-ip	remote-port	protocol
04:44:00	1.2.3.4	100	UDP
04:44:06	10.10.10.10	200	UDP

04:44:13	1.2.3.4	100	UDP
04:44:18	1.2.3.4	150	UDP
04:44:22	1.2.3.4	100	UDP
04:44:27	1.2.3.4	100	UDP
04:44:35	1.2.3.4	100	UDP

Additionally, this sample transport policy has the following properties:

- threshold=4
- period=30

From the table, the system keeps a count of instances of remote-ip/remote-port/protocol occurrences for each time period. The count for the table above would look as follows:

Table 7: Count of instances of remote-ip/remote-port/protocol occurrences for each time period

Occurrence	Count	Notes
1.2.3.4/100/UDP	4	The fifth occurrence fell in a new period. However, the threshold is four, so this constitutes a DOS attack!
1.2.3.4./150/UDP	1	The different port causes this packet to be counted separately.
10.10.10/200/UDP	1	None

If the message queue fills, regardless of whether the period interval has expired, the OS-E immediately executes all DOS policies.

How the Inactivity Timer Works

The inactivity timer is a mechanism for ensuring that when the system denies access to an offending sender, there is a time at which the prohibition expires. Once the sender has ceased tripping the kernel rule, the dynamic rules created in response to the configured threshold being crossed, the duration of the inactivity timer determines when the sender can resume communications.

Prior to each interval, the transport engine checks the kernel rule counter and caches the data. At the next interval, the system compares the current value to the cached value. If the counter has gone up, the system overwrites the previous cached value with the current counter. If the value is the same, which indicates that no new packets have been caught by the policy, then the system checks the inactivity time stamp. Once the inactivity timer times out, the kernel rule is deleted.

Using Operators and Regular Expressions

The OS-E uses some predefined relational operators for building conditions lists and predicate statements with elements of the same type. For example, use these operators to define ranges or compare values for equality or inequality. Your statements form logical expressions to determine choice, such as inclusion or exclusion, and sometimes action.(For enumerated lists, IP addresses, ports, and regular expressions, you use match and exclude statements.) The operators are as follows:

- eq=equal to
- ne=not equal to
- gt=greater than
- lt=less than
- ge=greater than or equal to
- le=less than or equal to

A regular expression is a formula for matching strings that follow some pattern. Many of the conditions and predicates require a regular expression entry. The OS-E uses PERL-compliant regular expressions.

Go to one of the following Web sites for complete instructions on forming regular expressions:

- http://www.perl.com/doc/manual/html/pod/perlre.html
- http://www.oreilly.com/catalog/regex/
- http://www.oreilly.com/catalog/regexppr/

Setting DOS Security Levels in the Net-Net OS-E Management System

You can assign security levels to DOS policies using the OS-E Management System Web. The security levels are:

- None — DOS policies are not configured.
- Low security
- Medium security
- High security

Caution: Changing the DOS security level removes all DOS policies from the configuration file.

To change the DOS security level, perform the following steps:

1. From the OS-E Management System, click on **VSP** from the menu tree.

2. Select **Set DOS security level.**

3. Set the security level to **None**, **Low**, **Medium**, or **High**.

All policies have an inactivity timeout that determines how long a DOS rule remains in effect once the DOS attack stops. DoS rules remain in effect for as long as the DoS attack persists. All the above policies are set to 300 seconds, except the URL policy which is set to 1,000,000 seconds.

Low security — Set DOS Security to "Low"

Transport policies

- **LowSecurity_remoteIP**—If an IP address is detected more than the 30,000 times over a 30-second period, then the transport DoS engine will block it.

- **LowSecurity_subnet**—If IP addresses within the same 255.255.255.0 subnet are detected more than 60,000 times over a 30 second period, then the transport DOS engine blocks the entire subnet. This policy excludes packets directed at the SNMP port.
- **LowSecurity_sip**—Same as LowSecurity_subnet, but only considers packets directed at the SIP port 5060, detected 4000 times over a 30 second period.
- **LowSecurity_sip_tls**—Same as LowSecurity_subnet, but only considers packets directed at SIP TLS port 5061, detected more than 600 times over a 30 second period.
- **LowSecurity_snmp**—Same as LowSecurity_subnet, but only considers packets directed at SNMP port 161, detected more than 4000 times over a 30 second period.
- **LowSecurity_https**—Same as LowSecurity_subnet, but only considers packets directed at HTTPS port 443, detected more than 200 times over a 30 second period.
- **LowSecurity_safetynet**—If IP addresses within the same 255.255.255.0 subnet are detected more than 50,000 times over a 5 second period, then the transport DoS engine blocks the entire subnet.

SIP Policies

- **LowSecurity_remoteIP _alert**—If the same IP address is detected more than 200 times over a 10 second period, an alert in the event log is generated. The packets are not blocked.
- **LowSecurity_fromUser**—If the same "From" user is detected more than 1000 times over a 10 second period, the SIP DoS engine blocks it.

Medium Security — Set DOS Security to “Medium”

Same policies as Low security, plus the following:

Transport Policies

- **MediumSecurity_remotePort**—If an IP address/remote port is detected more than 800 times over a 15-second period, the transport DoS engine blocks it. This policy excludes packets directed at the SNMP port.

SIP Policies

- **MediumSecurity_SPIM**—If the same "From" user is detected more than 200 times over a 10-second period with the SIP MESSAGE request method, the SIP DOS engine blocks it. This policy blocks IM spam.
- **MediumSecurity_SPIT**—If the same "From" user is detected more than three times over a 10-second period with the INVITE request method, the SIP DOS engine blocks it. This policy operates on autodial, telemarketing calls.
- **MediumSecurity_socket_timeout**—If the same remote IP address is detected more than 40 times over a 10 second period, and if the connection resulted in a TCP connection timeout, the SIP DOS engine blocks it. This prevents a DOS attack in which the attacker opens TCP sockets, but does not use them.

URL Policies

- **MediumSecurity**—If the same embedded URL within SIP IM messages is detected more than 20 times over 60 seconds, messages with that URL are dropped. This policy blocks the spread of downloaded viruses that generate IMs to propagate the virus to recipients configured in the address book

High Security — Set DOS Security to ‘High”

Same policies as Medium security, plus the following:

SIP Policies

- **HighSecurity_bad_headers**—If the same remote IP address is detected more than 50 times over a 10 second period with a "bad" SIP header, the SIP DOS engine will block it.
- **HighSecurity_policy_rejected**—If the same "From" user was rejected by the session policy configuration more than 500 times over a 10 second period, the SIP DoS engine will block it.

URL Policies

The High Security URL policy is the same as the **Medium Security** URL policy

Sample DOS Configuration

This section provides sample CLI sessions that configure policies that capture DOS attacks. For detailed information on using the actual **set** commands to define DOS queries and policies, refer to the *Net-Net OS-E – Objects and Properties Reference.*

Configuring DOS Policies in the CLI

The VSP **dos-policies** object allows you to define the transport, SIP, and URL policy condition lists, which define the point at which activity is determined to be part of a DOS attack and what action is to be taken.

Transport Policy

The properties you set in the transport policy object define the “rules” for applying the condition-list to the transport table. At the transport level, the OS-E can filter on data based on fields of the TCP header. These fields are **remote-port** and **protocol**.

The **transport-policy** specifies the following:

- The **remote-port** and **protocol** fields to examine (**select**).
- The frequency (**period** of time in seconds) between checks to the DOS database.
- The **threshold** on the number of matches that must be found in the database over the configured period of time before a DOS policy performs a filtering action.
- The **condition-list** property references the criteria for choosing which packets then get run through the **select** screening to build the final result set.

CLI Session

```
NNOS-E> config vsp policies
```

```
config policies> config dos-policies
config dos-policies> config transport-policy 1
config transport-policy 1> set description “Filter out bad guys”
config transport-policy 1> set admin enabled
config transport-policy 1> set select remote-port+protocol
config transport-policy 1> set threshold 50
config transport-policy 1> set period 45
config transport-policy 1> set condition-list "vsp policies dos-policies
                           transport-condition-list 1"
Creating 'vsp\policies\dos-policies\transport-condition-list 1'
config transport-policy 1> return
config dos-policies> config transport-condition-list 1
config transport-condition-list 1> set operation OR
config transport-condition-list 1> set condition remote-ip 10.10.10.10
config transport-condition-list 1> set condition remote-port 2525
```

SIP Policy

The properties in the **sip-policy** object define the “rules” for applying the **condition-list** to the SIP table. The SIP policy specifies the following:

- The TCP/UDP/IP/SIP header fields to examine (**select** property).
- The frequency (**period** of time in seconds) between checks to the DOS database.
- The **threshold** on the number of matches that must be found in the database over the configured period of time before a DOS policy performs a filtering action.
- The **action** to take on closing “bad” calls; filter or alert.

  —**Filter** discards all future calls that match the policy

  —**Alert** allows the packets to pass, but generates a log event.
- The **inactivity-period** removes a filtering action due to inactivity if the configured time setting expires.
- The **condition-list** reference adds the criteria for choosing which packets then get run through the **select** screening to build the final result set.

CLI Session

```
NNOS-E> config vsp policies
config policies> config dos-policies
config dos-policies> config sip-policy 1
config sip-policy 1> set description “Filter out INVITEs from 1.2.3.4”
config sip-policy 1> set admin enabled
config sip-policy 1> set select from-user
config sip-policy 1> set threshold 50
config sip-policy 1> set period 45
config sip-policy 1> set inactivity-period 180
config sip-policy 1> set condition-list "vsp policies dos-policies sip-
condition-list 1"
config sip-policy 1> return
config dos-policies> config sip-condition-list 1
config sip-condition-list 1> set operation AND
config sip-condition-list 1> set condition from-user exclude hal
config sip-condition-list 1> set condition result match policy-discard
config sip-condition-list 1> set condition header match .*dave.*
```

URL Policy

The properties in the **url-policy** object define the “rules” for applying the condition list to the URL table. The URL policy specifies the following:

- The frequency (**period** of time in seconds) between checks to the DOS database.
- The **threshold** on the number of matches that must be found in the database over the configured period of time before a DOS policy performs a filtering action.
- The **condition-list** list property reference defines which URL entries to examine, or those URL entries not to examine.

CLI Session

```
NNOS-E> config vsp policies
config policies> config dos-policies
config dos-policies> config url-policy 1
config url-policy 1> set description “Filter out IM virus”
config url-policy 1> set admin enabled
config url-policy 1> set threshold 5
config url-policy 1> set period 45
config url-policy 1> set condition-list "vsp policies dos-policies url-
condition-list 1"
config url-policy 1> return
config dos-policies> config url-condition-list 1
config url-condition-list 1> set url-condition match *buddy*
```

Examining the DOS Packet History

The OS-E Management System **Call Logs** tab allows you to view the packet history in the DOS database for which there are configured DOS policies.

Administering the DOS Database

The **master-services** database and dos-defense objects allows you to administer the DOS database on a host device. You can configure the time of day and interval when the database is purged of old entries.

For detail information on administering the DOS database, refer to the *Net-Net OS-E – System Administration Guide*.

Managing DOS Policy Results

There are several mechanisms for observing the effectiveness of your DOS policy configuration:

- The system generates an SNMP trap and a log message each time a DOS policy detects a DOS attack.
- DOS status providers:

  — show dos-collection

  — show dos-database-entry

  — show dos-query-status

  — show dos-recent-sip-from-user

  — show dos-recent-sip-ip

  — show dos-recent-sip-port

  — show dos-recent-transport-ip

  — show dos-recent-transport-port

— **show dos-rules**

— **show dos-sip-counters**

— **show dos-sip-summary**

— **show dos-transport-counters**

— **show dos-transport-summary**

— **show dos-url-counters**

Configuring Secure Trunking Networks

This section provides information on configuring secure network trunks. A secure trunk uses TLS on the signaling stream and SRTP on the media stream between the OS-E systems in enterprise and service provider networks.

Sample Secure Trunking Networks

The following image illustrates a sample service provider network using the OS-E systems in three branch offices. Traffic between each OS-E node uses TLS and SRTP encryption on the signaling and media streams. It assumes that the phones in this network do not support SRTP. (See the note at the end of the next section.)

Call Traversal In the Secure Trunk

Starting at the top of the illustration in the following image:

1. SIP phone calls over a TLS transport from subscribers enter the secure trunk at the OS-E@San Jose, where the inbound call signaling stream is decrypted using the OS-E's inbound session configuration.

2. Using the outbound session configuration, the OS-E@San Jose then reencrypts the TLS signaling stream, and then encrypts the SIP media stream using SRTP. This creates the secure trunk as the SIP call session traverses the Internet to the call destination at OS-E@Boston.

3. When the SIP call reaches the OS-E@Boston, the inbound session configuration decrypts the TLS and SRTP call streams.

4. The outbound session configuration at OS-E@Boston then reencrypts the SIP signaling stream to TLS before forwarding the call to the destination.

Note: Currently, most SIP phones do not support SRTP. Therefore, media streams outside of the secure trunk are sent "in the clear" to the call destination when received by the service provider. However, you can configure the outbound session configuration to perform SRTP encryption and offer it the destination phone. If the phone does not support the SRTP, then RTP is used to deliver the media stream.

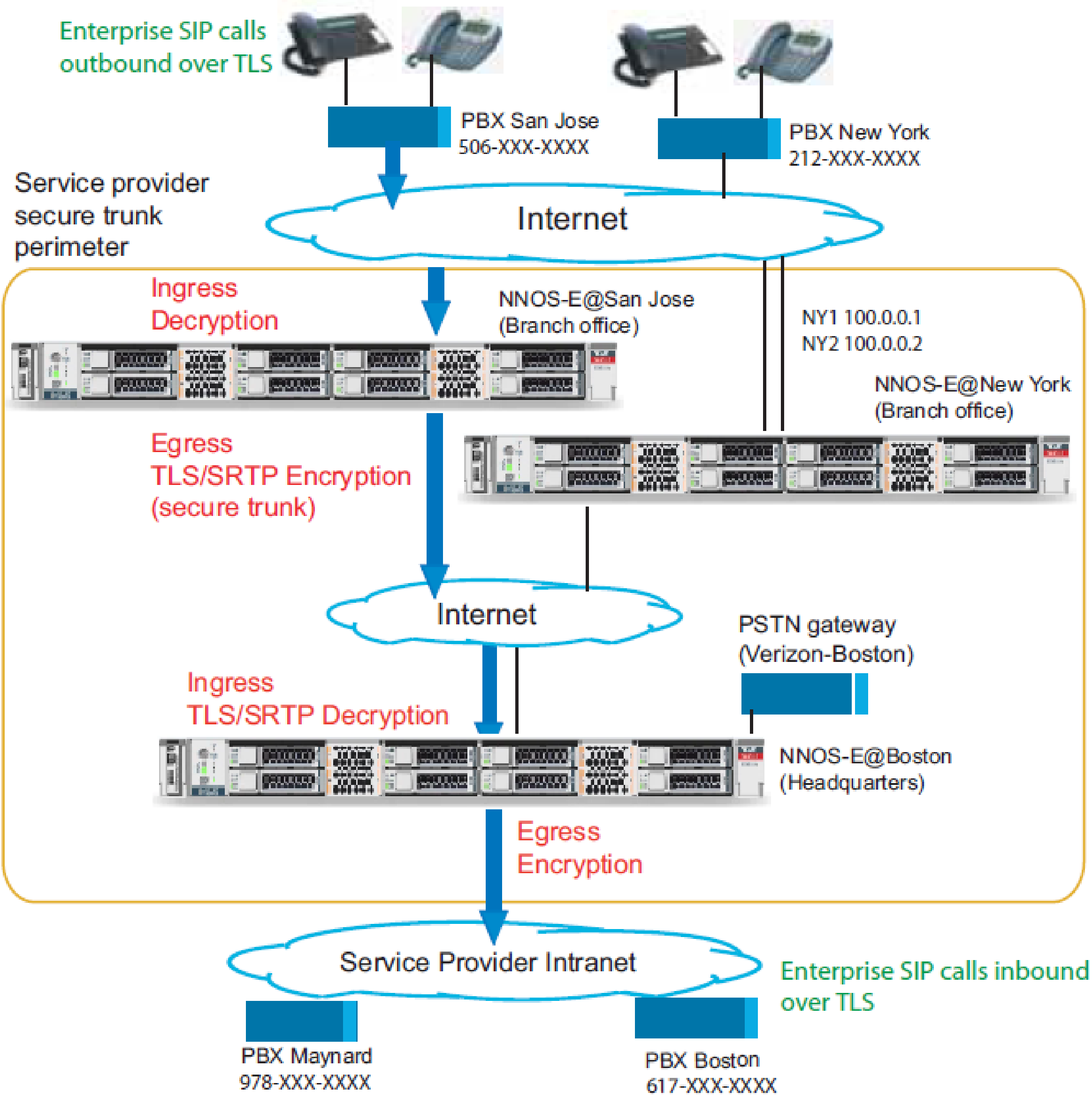

Enterprise SIP calls outbound over TLS
PBX San Jose
506-XXX-XXXX
PBX New York
212-XXX-XXXX
Service provider secure trunk perimeter
Internet
Ingress Decryption
NNOS-E@San Jose (Branch office)
NY1 100.0.0.1
NY2 100.0.0.2
NNOS-E@New York (Branch office)
Egress TLS/SRTP Encryption (secure trunk)
Internet
PSTN gateway (Verizon-Boston)
Ingress TLS/SRTP Decryption
NNOS-E@Boston (Headquarters)
Egress Encryption
Service Provider Intranet
Enterprise SIP calls inbound over TLS
PBX Maynard
978-XXX-XXXX
PBX Boston
617-XXX-XXXX

Figure 16: Call traversal in the secure trunk

Configuration Steps

There are several steps that you need to perform to configure a secure trunking network.

1. Configure the enterprise servers and connections. These are the SIP gateways involved in the routing of SIP calls to their destination SIP servers.

2. Configure the server-pools.

3. Configure the SIP connections — the SIP PBXs communicating with the OS-E you are configuring.

4. Configure dial-plan routes to SIP gateways and connections.

5. Configure the inbound and outbound session configuration entries for the encryption and decryption policies, and then apply the entries to the SIP gateways and SIP connections. There are the ingress and egress points for encryption and decryption.

6. Configure the carriers and trunk groups.

Net-Net OS-E Encryption/Decryption Policies

For SIP gateways and SIP connections, you need to define encryption and decryption policies to the inbound (ingress) and outbound (egress) SIP sessions. The following table lists the encryption methods used on the ingress and egress sessions and the encryption or decryption policy applied to each session. In a secure trunk, the SIP media and signaling streams are encrypted using TLS (on signaling) and SRTP (on media).

Table 8: Encryption and decryption policies

Server type	Encryption methods	Encryption/decryption policy (default names)
SIP gateway	IN-ENCRYPTION (on OS-E ingress call direction)	NNOS-EDecryptPolicy
SIP gateway	OUT-ENCRYPTION (on OS-E egress call direction)	NNOS-EEncryptPolicy
SIP connection	IN-ENCRYPTION (on OS-E ingress call direction)	GenericDecryptPolicy
SIP connection	OUT-ENCRYPTION (on OS-E egress call direction)	GenericEncryptPolicy

How the OS-E Performs Encryption and Decryption

Encryption is symmetric on inbound and outbound sides of the OS-E, regardless of the call direction. This mean that encryption and decryption always occurs to that side, or it does not. This is determined at call setup time.

In the following image, the red arrows on the left indicate encrypted media, and blue arrows on the right indicate plaintext media. The OS-E performs the encryption and decryption to make the SIP session either encrypted or plaintext.

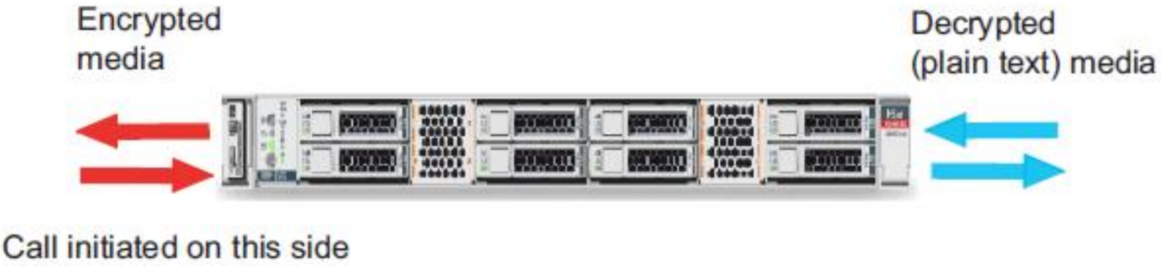


Figure 17: Encryption and decryption performed in OCASC

The **in-encryption** policy causes the left side to be encrypted and decrypted. The SIP phone call initiated on the left must propose encryption in the Session Description Protocol (SDP) as well as provide the encryption key. If the in-encryption mode is set to require and the phone does not send it, the OS-E rejects the call. The in-encryption mode must be set to allow or require, as proposed by the SIP phone.

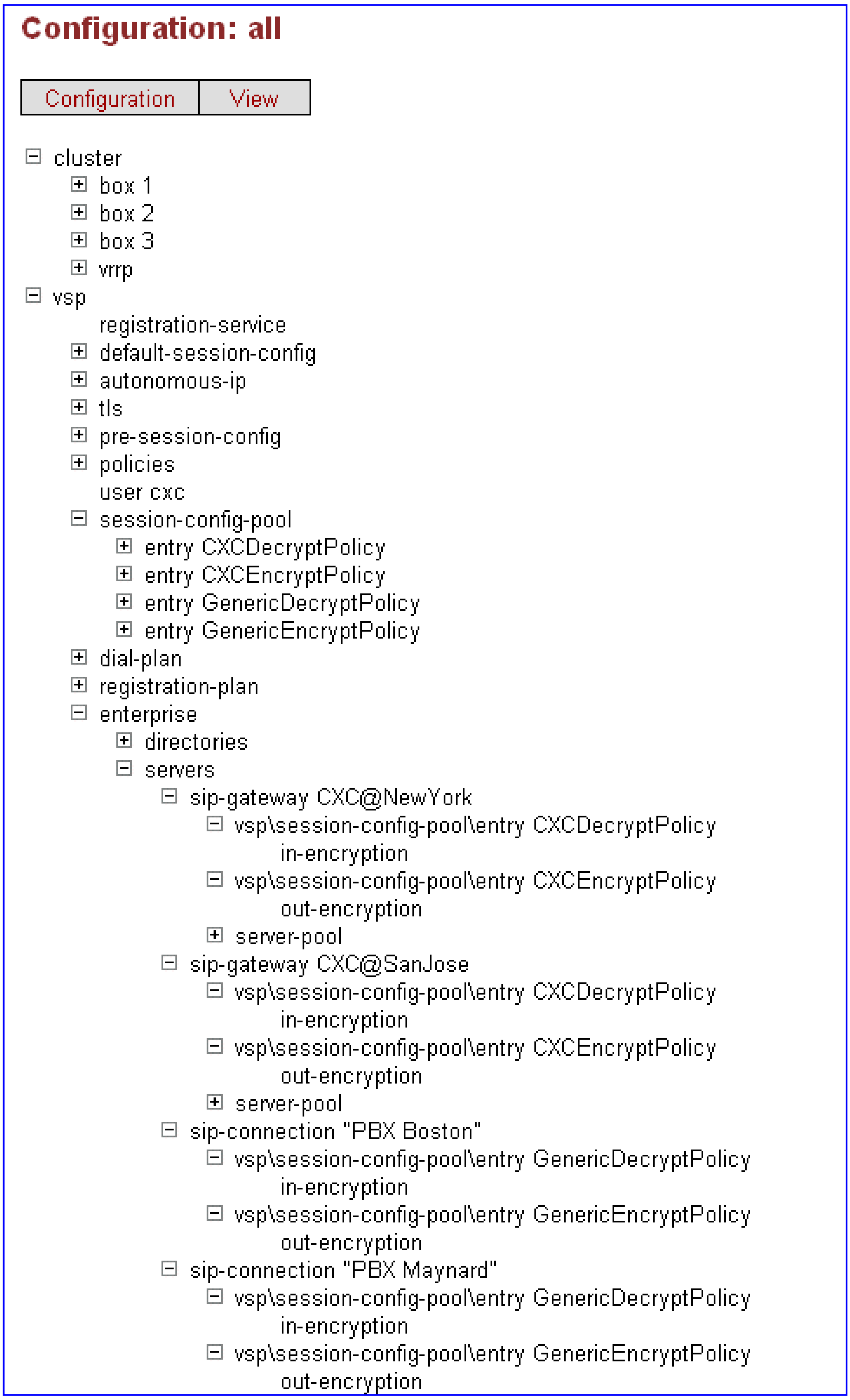
The **out-encryption** policy determines whether we do encryption/decryption on the right side. If the out-encryption mode is set to offer or require, the OS-E proposes encryption in the SDP sent to the right side. If the right side does not want encryption, the destination phone may send the OS-E an SDP that does not contain encryption, therefore rejecting the call. If the OS-E is set to require encryption and the received SDP has no encryption, the OS-E rejects the call. If the OS-E is set to offer encryption and the received SDP has no encryption, the OS-E allows the call to proceed without encryption or decryption on the right side, as shown in the above image.

Applying Encryption and Decryption Policies

You configure encryption and decryption modes when you configure SIP gateways and SIP connections. In the OS-E Management System, go to the Policies section on the vsp/enterprise/servers/sip-gateway or vsp/enterprise/servers/sip-connection page and specify the **inbound-session-config-entry** and **outbound-session-config entry** by making a selection from the session-config-pool.

Open the session-config-pool and create entries for the in-encryption and out-encryption properties. Specify the encryption mode and encryption type as required for the ingress and egress call streams. In-encryption modes include: **disable**, **allow**, and **require**.Out-encryption modes include: **none**, **offer**, **require**, and **follow**.

The following image illustrates the configuration tree showing the encryption and decryption policies and where they are applied under each SIP gateway and SIP connection.



Configuration: all

Configuration | View

```
⊟ cluster
    ⊞ box 1
    ⊞ box 2
    ⊞ box 3
    ⊞ vrrp
⊟ vsp
      registration-service
    ⊞ default-session-config
    ⊞ autonomous-ip
    ⊞ tls
    ⊞ pre-session-config
    ⊞ policies
      user cxc
    ⊟ session-config-pool
        ⊞ entry CXCDecryptPolicy
        ⊞ entry CXCEncryptPolicy
        ⊞ entry GenericDecryptPolicy
        ⊞ entry GenericEncryptPolicy
    ⊞ dial-plan
    ⊞ registration-plan
    ⊟ enterprise
        ⊞ directories
        ⊟ servers
            ⊟ sip-gateway CXC@NewYork
                ⊟ vsp\session-config-pool\entry CXCDecryptPolicy
                      in-encryption
                ⊟ vsp\session-config-pool\entry CXCEncryptPolicy
                      out-encryption
                ⊞ server-pool
            ⊟ sip-gateway CXC@SanJose
                ⊟ vsp\session-config-pool\entry CXCDecryptPolicy
                      in-encryption
                ⊟ vsp\session-config-pool\entry CXCEncryptPolicy
                      out-encryption
                ⊞ server-pool
            ⊟ sip-connection "PBX Boston"
                ⊟ vsp\session-config-pool\entry GenericDecryptPolicy
                      in-encryption
                ⊟ vsp\session-config-pool\entry GenericEncryptPolicy
                      out-encryption
            ⊟ sip-connection "PBX Maynard"
                ⊟ vsp\session-config-pool\entry GenericDecryptPolicy
                      in-encryption
                ⊟ vsp\session-config-pool\entry GenericEncryptPolicy
                      out-encryption
```

Figure 18: Configuration tree showing the encryption and decryption policies

Admission Control

This section covers OS-E admission control for SIP INVITE, SIP REGISTER, and TLS sessions. Admission control allows you to limit calls to the OS-E that might otherwise heavily consume OS-E memory and storage resources.

Call Admission Control

OS-E call admission control (CAC) allows you to control or limit the number of calls to and from various SIP devices. Call admission control, which only applies to SIP INVITE traffic, operates at the following levels:

- VSP (operating on all calls to the OS-E system.)
- SIP gateways (carrier and enterprise)
- Trunk groups
- Calling groups
- User agents (location)

Call admission control prevents malfunctioning or improperly-configured devices from consuming critical resources that impact the performance of the OS-E system. Call routing loops, for example, can be prevented with call admission control since it is not always possible for the OS-E to detect certain types of loops (per RFC 3261) when the next-hop device is a back-to-back user agent (B2BUA).

VSP Control

By default, VSP call admission control is *disabled*. When *enabled*, you can limit the total number of calls that can be processed by the OS-E at any time. Use the **call-admission-control** property to enable or disable call admission control.

CLI Session

```
config> config vsp admission-control
config admission-control> set call-admission-control enabled
```

Call control limits are optimized for the specific platform on which you are running the OS-E with a default setting of *automatic*. You can display the automatic values with the **show automatic-settings** status provider.

```
NNOS-E> show automatic-settings
name value
---- -----
cac-max-calls 7500
cac-max-calls-in-setup 1500
cac-max-number-of-tls 3000
cac-max-tls-in-setup 425
cac-min-calls-in-setup 10
max-number-of-sessions 7500
max-routes 1048576
stack-socket-event-threads-max 4
stack-socket-threads-max 4
stack-worker-threads 4
```

The current settings can be viewed with the **show call-admission-control** command.

```
NNOS-E> show call-admission-control
                                    name: default
                  call-admission-control: disabled
                               max-calls: 7500
                      max-calls-in-setup: 1500
                      min-calls-in-setup: 10
        calls-in-setup-dynamic-threshold: 1500
                        cpu-monitor-span: 20 seconds
                    cpu-monitor-interval: 10 seconds
                         average-sip-cpu: 0 %
                calls-high-cpu-threshold: 90 %
                 calls-low-cpu-threshold: 50 %
                           current-calls: 0
                  current-calls-in-setup: 0
                              most-calls: 0
                     most-calls-in-setup: 0
                       max-calls-dropped: 0
          max-calls-dropped-last-logging:
max-calls-in-setup-dropped-this-interval: 0
max-calls-in-setup-dropped-last-interval: 0
              max-calls-in-setup-dropped: 0
```

Limiting Based On Calls

The following VSP **admission-control** settings restrict the number of calls that can be processed by the VSP.

- **cac-max-calls**—The maximum number of allowed concurrent calls.
- **cac-max-calls-in-setup**—The maximum number of allowed calls in the setup stage.

Limiting Based On CPU

The following VSP settings calculate a dynamic threshold so that the OS-E rejects calls based on the CPU usage. The initial dynamic threshold value is the **cac-max-calls-in-setup** setting.

- **cpu-monitor-span**—The number of seconds over which the OS-E calculates the total system CPU average. At the end of the span, the average value is compared to the call CPU thresholds to determine whether to modify the dynamic threshold. The longer the span, the fewer the changes to the thresholds. A shorter span will result in reaction to brief CPU activity spikes.
- **cpu-monitor-interval**—The frequency in seconds over which the OS-E calculates the total system CPU average for the last span.
- **calls-high-cpu-threshold**—When this threshold is reached, the dynamic threshold value decreases by 10% but never goes below the **cac-min-calls-in-setup** setting.
- **calls-low-cpu-threshold**—When this threshold is reached, the dynamic threshold value increases by 16% if the average CPU is less than the low threshold and by 4% if the less than the high threshold.
- **cac-min-calls-in-setup**—This lowest possible value of the dynamic threshold.
- **call-response-code-at-threshold**—The response code sent when a request was rejected because the dynamic threshold was reached.
- **call-response-string-at-threshold**—The response string sent when a request was rejected because the dynamic threshold was reached.

Server Control

You can configure the OS-E to perform call admission control under the following server objects:

- **vsp/enterprise/servers/sip-gateway** *name*/**server-pool/serve**r *name*
- **vsp/carriers/carrier/gateway** *name*
- **vsp/carriers/carrier/gateway** *name*/**trunk-group** *name*
- **vsp/calling-group/group** *name*

Each of these server objects can be set to limit inbound or outbound calls based on absolute values or estimated bandwidth. The OS-E currently keeps track of inbound and outbound calls using the same counters, so inbound calls may affect emission control and the reverse.

You can enable or disable admission (inbound) or emission (outbound) calls with the respective server object properties. You can view the current settings with the following commands:

- **show sip-server-cac**
- **show gateway-cac**
- **show trunk-cac**
- **show calling-group-cac**

Limiting Based On Calls

The following settings restrict the amount of calls that can be sent or received from a server.

- **max-number-of-concurrent**—The maximum number of allowed concurrent calls.
- **max-calls-in-setup**—The maximum number of allowed calls in the setup stage.
- **call-rate-limiting** (secondary)—The number of calls allowed during a given period of time. For example, if the *calls-per-interva*l setting is 60 and the *smoothing-interval* setting is 1, the OS-E allows 60 calls/second. Once that limit is reached, the OS-E attempts to hunt for another server. If no servers are found, the OS-E rejects the call with the specified *result-code* and *result-string*.

—*calls-per-interval*—The maximum number of calls allowed in this period.

—*smoothing-interval*—The period when the OS-E allows a burst of calls without rejection.

—*result-code*—The response code sent when a request is rejected.

—*result-string*—The response string sent when a request is rejected.

Limiting Based On Bandwidth

The following setting restricts the bandwidth to and from a server. The bandwidth is an estimate based on the CODEC negotiated in the SDP. You can view the estimated CODEC bandwidth with the **show codec-info** status command.

- **max-bandwidth**—The maximum bandwidth that can handled by this server. When set to unlimited (default), the bandwidth is limited only by the physical links or processing engine.

User-Agent Control

You can configure the OS-E to perform call admission control for individual User-Agents (UAs). These settings are found under the **session-config/location-call-admission-control** configuration object.

When a UA first registers, the values are copied to the location-cache entry. The session configuration can be from either the default-session-config or any session-config entry pools that are associated with this UA during the registration process. You can view the current setting with the **show location-cache-cac** status command.

- **max-number-of-concurrent**—The maximum number of allowed concurrent calls.
- **max-calls-in-setup**—The maximum number of allowed calls in the setup stage.
- **call-rate-limiting**—The number of calls allowed in a certain period of time. For example, if the *calls-per-interval* setting is 60 and the *smoothing-interval* setting is 1, the OS-E allows 60 calls/second. Once that limit is reached, the OS-E attempts to hunt for another server. If no servers are found, the OS-E rejects the call with the specified *result-code* and *result-string*.

—*calls-per-interval*—The maximum number of calls allowed in this period.

—*smoothing-interval*—The period for which we allow a burst of calls without rejecting.

—*result-code*—The response code sent when a request is rejected.

—*result-string*—The response string sent when a request is rejected.

Limiting Based On Bandwidth

The following setting restricts the bandwidth to and from a UA. The bandwidth is an estimate based on the CODEC negotiated in the SDP. You can view the estimated CODEC bandwidth with the **show codec-info** status command.

- **max-bandwidth**—The maximum bandwidth that can handled by this UA. When set to unlimited (default), the bandwidth is limited only by the physical links or processing engine.

Using the Session-Config Override

The OS-E call-admission-control feature is bypassed when the emergency-settings feature in the session-config is enabled. This allows the administrator to associate this session-config to a an emergency dial-plan (for example 911) or to dynamically load it on the session using RADIUS or WSDL when placing a call. These calls still count towards future CAC checks but would not be rejected.

Registration Admission Control

The OS-E registration admission control feature allows you to control or limit the amount of registration to a server. Registration admission control, which applies to SIP REGISTERs, operates at the following levels:

- VSP (operating on all calls to the OS-E system)
- SIP gateways (enterprise)

Registration admission control, when enabled, prevents the OS-E and/or the server from accepting more registrations than it can possibly process.

VSP Control

By default, VSP registration admission control is disabled. When enabled, you can limit the total number of registrations that can be processed by the OS-E at any time. Use the **registration-admission-control** property to enable or disable registration admission control.

CLI Session

```
config> config vsp admission-control
config admission-control> set registration-admission-control enabled
```

Registration control limits are optimized for the specific platform on which you are running the OS-E with a default setting of *automatic*. You can display the automatic values with the **show automatic-settings** status provider.

```
NNOS-E> show automatic-settings
name value
---- -----
cac-max-calls 7500
cac-max-calls-in-setup 1500
cac-max-number-of-tls 3000
cac-max-tls-in-setup 425
cac-min-calls-in-setup 10
max-number-of-sessions 7500
max-routes 1048576
stack-socket-event-threads-max 4
stack-socket-threads-max 4
stack-worker-threads 4
```

The current settings can be viewed with the **show registration-admission-control** command.

```
NNOS-E> show registration-admission-control

                                   name: default
         registration-admission-control: disabled
                      max-registrations: 30000
   pending-registrations-high-watermark: 500
    pending-registrations-low-watermark: 10
pending-registrations-dynamic-threshold: 500
                       cpu-monitor-span: 20 seconds
                   cpu-monitor-interval: 10 seconds
                        average-sip-cpu: 0 %
       registrations-high-cpu-threshold: 90 %
        registrations-low-cpu-threshold: 70 %
                  total-client-bindings: 0
              registrations-in-progress: 0
         registrations-most-in-progress: 0
                 registrations-sessions: 0
            processed-new-registrations: 0
        processed-waiting-registrations: 0
     processed-challenged-registrations: 0
          processed-other-registrations: 0
 suppressed-registrations-this-interval: 0
 suppressed-registrations-last-interval: 0
           suppressed-new-registrations: 0
       suppressed-waiting-registrations: 0
    suppressed-challenged-registrations: 0
            last-register-suppressed-at:
          discarded-other-registrations: 0
```

```
  last-register-discarded-at:
edp-transactions-in-progress: 0
```

Limiting Based On Registers

The following setting restricts the amount of registers that can be processed by the VSP.

- **max-number-of-registrations**—The total number of registrations that can be processed by this VSP.

Limiting Based On CPU

The following settings calculate a dynamic threshold so that the OS-E rejects calls based on the CPU usage. The initial dynamic threshold value is the **pending-registrations-high-watermark.**

- **cpu-monitor-span**—The number of seconds over which the OS-E calculates the total system CPU average. At the end of the span, the average value is compared to the call CPU thresholds to determine whether to modify the dynamic threshold. The longer the span, the fewer the changes to the thresholds. A shorter span will result in reaction to brief CPU activity spikes.
- **cpu-monitor-interval**—The frequency in seconds over which the OS-E calculates the total system CPU average for the last span.
- **registrations-high-cpu-threshold**—When this threshold is reached, the dynamic threshold value decreases by 10% but never goes below **pending-registrations-low-watermark**.
- **registrations-low-cpu-threshold**—When this threshold is reached, the dynamic threshold value increases by 16% if the average CPU is less than the low threshold and by 4% if the less than the high threshold.
- **pending-registrations-low-watermark**—This lowest possible value of the dynamic threshold.
- **pending-registrations-low-watermark**—This highest possible value of the dynamic threshold.

Server Control

You can configure the OS-E to perform registration admission control under the **vsp\enterprise\servers\sip-gateway** *name*\server-pool\server *name*

This object can be set to an absolute number of registrations that it can accept. The admission-control setting on the server or gateway must be set to enabled for registration-admission-control to be active.

You can view the current settings with the **show sip-server-cac** and **show gateway-cac** status commands.

Limiting based on registers

The following settings restrict the amount of registers that can be processed by this server.

- **max-number-of-registrations**—The total number of registrations that this server can handle.
- **max-registrations-in-progress**—The total number of registrations in progress that this server can handle.

TLS Admission Control

The OS-E TLS admission control allows you to control or limit the amount of TLS connections that can be established. The control is applied at the VSP configuration level only.

VSP Control

By default, VSP TLS admission control is disabled. When enabled, you can limit the total number of TLS connections that can be processed by the OS-E at any time using the VSP admission-control object.

- **cac-max-number-of-tls**—The total number of TLS connections that can be established.
- **cac-max-tls-in-setup**—The total number of TLS connections in progress.

TLS admission control limits are optimized for the specific platform on which you are running the OS-E with a default setting of *automatic*. You can display the automatic values with the **show automatic-settings** status command.

```
NNOS-E> show automatic-settings
name value
---- -----
cac-max-calls 7500
cac-max-calls-in-setup 1500
cac-max-number-of-tls 3000
cac-max-tls-in-setup 425
cac-min-calls-in-setup 10
max-number-of-sessions 7500
max-routes 1048576
stack-socket-event-threads-max 4
stack-socket-threads-max 4
stack-worker-threads 4
```

The current settings can be viewed with the **show tls-admission-control** status command.

```
NNOS-E> show tls-admission-control
                       name: default
      tls-admission-control: disabled
                  max-calls: 7500
         max-calls-in-setup: 1500
              current-calls: 0
     current-calls-in-setup: 0
          max-calls-dropped: 0
 max-calls-in-setup-dropped: 0
```