PolicyGuru®

Meta-Policy Controller

v3.0.2

User Guide

DOC-UG-P302-2021-0911

About SecureLogix

For more than 20 years, SecureLogix has profiled, tracked and defended customers against the schemes and threats plaguing unified communications networks. We’ve developed patented technology and assembled the most skilled team in the industry to monitor and protect some of the world's largest and most complex contact centers and voice networks.

We're not the largest IT vendor; we're the one with the start-up agility and decades of unrivaled enterprise experience. The one that is there when you need us, with superhero level support.

For more information about SecureLogix and its products and services, visit us on the Web at *https://securelogix.com/*.

Corporate Headquarters:
SecureLogix Corporation
13750 San Pedro, Suite 820
San Antonio, Texas 78232
Telephone: 210-402-9669 (non-sales)
Fax: 210-402-6996
Email: *info@securelogix.com*
Website: *https://www.securelogix.com*

Sales:
Telephone: 1-800-817-4837 (North America)
Email: *sales@securelogix.com*

Customer Support:
Telephone: 1-877-SLC-4HELP
Email: *support@securelogix.com*
Web Page: *https://support.securelogix.com*

Training:
Telephone: 210-402-9669
Email: *training@securelogix.com*
Web Page: *https://training.securelogix.com*

Documentation:
Email: *docs@securelogix.com*
Knowledge Base: *https://support.securelogix.com*

IMPORTANT NOTICE:

This manual and the software and/or Products described in it are furnished under license with SecureLogix Corporation ("SecureLogix") and may be used only in accordance with the terms of such license.

Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without prior written permission of SecureLogix.

The content of this manual is subject to change without notice. SecureLogix assumes no responsibility or liability for any errors or inaccuracies that may be contained herein or to correct the same.

ETM, We See Your Voice, SecureLogix, SecureLogix Corporation, and the SecureLogix Emblem are registered trademarks and registered service marks of SecureLogix Corporation in the U.S.A. and other countries. PolicyGuru is a registered trademark of SecureLogix Corporation in the U.S.A. Orchestra One, Call Secure, and Call Defense are trademarks and service marks of SecureLogix Corporation in the U.S.A. All other trademarks mentioned herein are believed to be trademarks of their respective owners.

This product is protected by one or more of the following patents: DE 1,415,459 B1, FR 1,415,459 B1, and GB 1,415,459 B1. U.S. Patents Pending.

PolicyGuru is used herein as shorthand notation to refer to the PolicyGuru® Solution.

Orchestra One is used herein as shorthand notation to refer to the Orchestra One™ Call Authentication Service.

This product also includes code written by other third parties.

Additional details regarding other third party code included in this product, including applicable copyright, legal, and licensing notices, are available in the "licenses" directories included with the third-party code.

Customer Support
for Your SecureLogix® Solution

1-877-SLC-4HELP
(1-877-752-4435)
support@securelogix.com
https://support.securelogix.com

SecureLogix Corporation offers telephone, email, and web-based support. For details on warranty information and support contracts, see our web site at

https://support.securelogix.com

Contents

PolicyGuru® Meta-Policy Controller Introduction

Concepts

The PolicyGuru® Meta-Policy Controller enables the construction and enforcement of customized voice/Unified Communications (UC) network security and business management Rules through a flexible and powerful Business Rule Management System (BRMS) called Business Central. These Rules are defined, managed, and implemented Enterprise-wide from the central, web-based management interface. Using a BRMS framework allows you to create Rules that can handle extremely complex call scenarios both in and out of the Enterprise.

User-defined PolicyGuru Rules specify the criteria by which a call is considered of interest and the action to be taken if call triggers a Rule: allow the call(s), block the call(s), or redirect the call(s) to another number.

Automated notifications can be configured to alert appropriate personnel when a call or a suspect calling pattern triggers a Rule, via a third-party application such as Splunk.

After you define the security, call-access control (CAC), and Call Authentication Rules for your enterprise, you install them on the PolicyGuru Server Applications securing your enterprise SIP Trunks, where they are continuously enforced in real time.

Simple Event Processing (SEP) Policy

In a manner similar to data network firewalls, SEP Policy provides real-time voice/UC application session access control and monitoring on a per-call basis, based on call setup details (source, destination, and direction). The user-defined Rules in the SEP Policy define whether specific calls are to be allowed, blocked, or redirected. Calls that match an SEP Rule specifying call treatment are terminated or redirected at call setup, preserving your network resources for legitimate business calls. For example, you can define a Rule that dictates that all calls from known harassing callers in a Harassing Callers Blacklist are to be terminated before they are set up.

SEP Policy Processing is driven by ENUM requests only. Because the call data available in ENUM queries is limited to events that occur at call setup, SEP Rule criteria include any combination of the following:

- **Blacklist Rules**—Source, Destination, Direction, and timestamp range.
- **Whitelist Rules**—Source, Destination.

The PolicyGuru BRMS GUI provides a **Guided Rule Editor** that includes a robust set of predefined SEP Rule-definition assets for building Rules. The set of defined SEP Rules constitutes the *SEP Policy*.

User-defined *Lists* define the phone numbers/URIs to which SEP Rules apply; these are identified as *Blacklists* and *Whitelists*. You can manually add entries to these Lists, or, for large Lists, you can import the entries from a file. Each List can contain either one to many individual phone numbers/URIs, or one to many Regular Expressions (Regex), which are used to define Ranges and Wildcards (such as all phone numbers/URIs in a certain area code).

Lists and SEP Rules work in conjunction to create and implement either *Blacklist Rules* or *Whitelist Rules*, as described below.

SEP Rule Types

Whitelist Rules—*Whitelist Rules* identify calls that are to be allowed and ignored. *Whitelists* (described below) are used to specify the phone numbers/URIs to which Whitelist Rules apply. These Rules do not explicitly fire—they represent default Allow Rules— and therefore cannot be alerted on, preserving processing resources for true calls of interest. This is especially valuable in high call volume environments, such as Contact Centers. See "Whitelists" below for details about Whitelists.

Blacklist Rules—*Blacklist Rules* specify actions and alerting for calls matching the Rule: allow the call as originally routed, block the call, or redirect the call to a different destination. Blacklist Rules are the only type of SEP Rule that fires, since Whitelist Rules specify calls to be ignored and allowed. *Blacklists* (described below) are used to specify the phone numbers/URIs to which Blacklist Rules apply.

It is important to note that Blacklist Rules can denote suspicious or malicious calls and provide protection from them by using them in Rules to block or redirect those calls. But they can also be used to create "Watch and Alert" Rules for any key traffic. Or, you can use them to create a Rule to specifically allow certain calls you want, while a subsequent Rule blocks all other calls. For example, this might be valuable in the case of an attack, to ensure network availability for critical calls.

Orchestra One™ Verification Request Rules—Orchestra One Verification Rules are Blacklist Allow Rules for integration with the Orchestra One Call Authentication Service (CAS). See "Defining SEP Policy Rules for Orchestra One™ Verification Requests" on page 50 for more information.

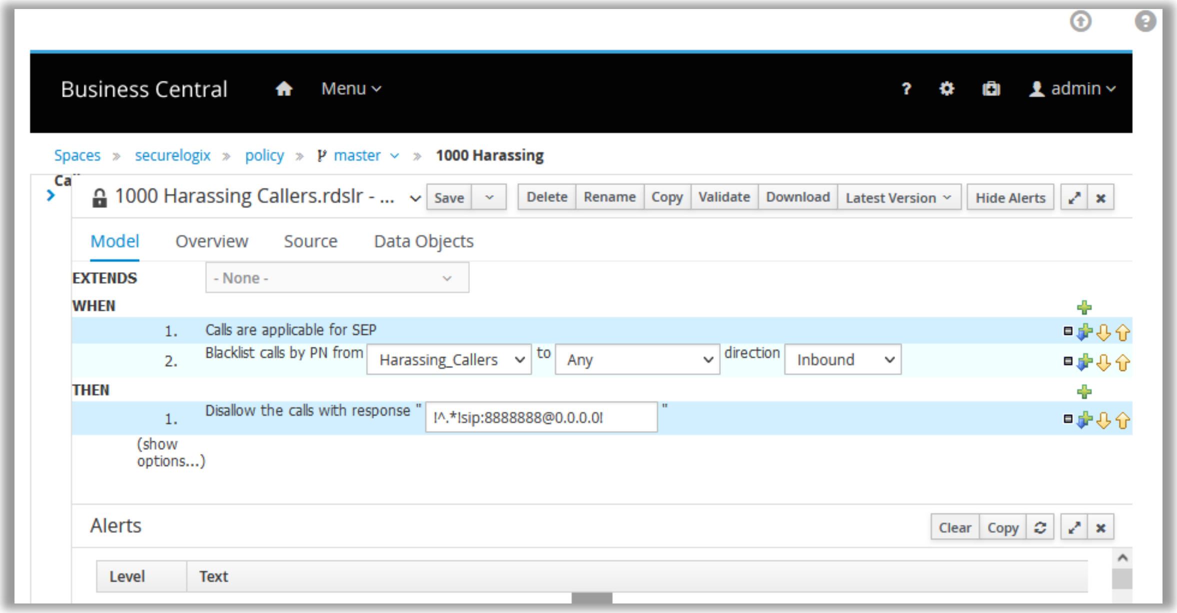
Understanding Lists

PolicyGuru SEP Rules use the following types of Lists to identify the called and calling numbers to which the Rule applies:

- *Whitelists* identify numbers/URIs for calls that are always to be allowed and not treated nor alerted on. They are used in *Whitelist Rules*.
- *Blacklists* contain numbers/URIs that are eligible for policy treatment (Termination or Redirection) and/or alerting. They are used in *Blacklist Rules*.

  As mentioned earlier, it is important to note that the term *Blacklist* does not denote that the numbers/URIs they contain are suspect or disallowed. Rather, it denotes listings that you want to be able to track or take action on.

  For example, you might have a **Harassing Callers** Blacklist of known harassing numbers you want to block and a **Specifically Allowed Callers** Blacklist that you can use in the case of an attack to authorize specific calls, while a subsequent Rule blocks all others.



EXTENDS - None -

WHEN

1. Calls are applicable for SEP
2. Blacklist calls by PN from Harassing_Callers to Any direction Inbound

THEN

1. Disallow the calls with response " !^.*!sip:8888888@0.0.0.0! "

(show options...)

Figure 1: SEP Rule Terminating Numbers in the Harassing_Callers Blacklist

The example SEP Rule in Figure 1 above shows a Rule that blocks phone numbers from the **Harassing_Callers** Blacklist.

Calls are blocked by supplying a regular expression to redirect the call to a nonexistent endpoint, resulting in a **404 Not Found** response. You can choose regular expressions that suit your enterprise practices.

How Lists are Matched in Policy Processing

In SEP Policy processing, values in PN Lists are evaluated via full string matching and can consist of letters and/or digits. This means that, in addition to fully qualified phone numbers, you can specify values such as **INVALID** in a PN List and then configure the Phone Number Normalizer to append that value to certain phone numbers during normalization (for example, those that fail a certain function in a **libphonenumber** library lookup). Phone Numbers in ENUM Requests are processed by the Phone Number Normalizer before Policy processing occurs. Therefore, when that PN List is used in an SEP Rule, called or calling numbers that have been normalized to append **-INVALID** will match the Rule.

Note: See the *PolicyGuru® Meta-Policy Controller Installation and Configuration Guide* for details about phone number normalization/ denormalization in the PolicyGuru Solution.

Similarly, values in Regex Lists are processed as Regex, by determining if any part of the normalized phone number string matches the regular expression as specified in the listing. For example, a Regex value of **badnumber** would match **this_is_a_badnumber!** appended to a normalized number.

Note: You can optionally configure the Phone Number Normalizer to denormalize such phone numbers after Policy processing into a fully qualified format or into the original received format (or any format you choose) for storage in the database. Or you can choose not to denormalize such numbers and store them in the database with the flag appended so that you can generate reports on that flag.

Normalized Phone Numbers

When an ENUM Request is received, the ENUM Server processes the source and destination numbers it contains through a *phone number normalizer* prior to SEP Policy processing. This phone number normalizer is user-configurable. The result is a *normalized* phone number. Phone Number Normalization refers to processing the input from Session Border controllers in such a way that it is relevant to the systems at a given site. This can include, for example, adding or removing country codes, area codes, or exchange, or adding flags, such that the numbers can be processed by SEP Rules. Number normalization occurs prior to Policy processing.

How SEP Rules are Processed

Rules appear in the **Project Explorer** pane of the BRMS GUI in ASCII alphabetical order (1-10, a-z, A-Z). All Whitelist Rules are processed first in the order in which they appear in the **Project Explorer** pane. Next, all Blacklist Rules are processed in the order in which they appear. This is true even if Whitelist and Blacklist Rules are intermixed in the order.

Only one SEP Rule can match a given call. When a call matches a Rule, the Rule fires, and the call is not processed against any subsequent Rules.

Therefore, it is recommended you place Whitelist Rules first, because as noted above, Whitelist Rules are always processed first. To ensure they appear at the top of the list of Rules, use a numbering scheme to organize your Rules, both for processing and management. One such scheme would be to begin the Whitelist Rules with 1000, Blacklist Allow Rules with 2000, and Blacklist Terminate or Redirect Rules with 3000. Increment the next Rule of the same type by 20 to allow for additional Rules to be placed in between if needed.

For example:

1000 – Outbound Exec Calls

3000 – Block Harassing Callers

3020 – Suspected Fraudsters

If a pair of Rules has the same purpose and simply uses different Lists, you might number them sequentially so they always remain together as a group. For example:

3000 – Block Harassing Callers: National Harassing Callers List

3001 – Block Harassing Callers: Customer-Specific List

How Calls are Blocked and Redirected

Calls are blocked by supplying a regular expression (Regex) to redirect the call to a nonexistent endpoint host address, such as !^.*!sip:8888888@0.0.0.0!, resulting in a **404 Not Found** response. You can choose regular expressions that suit your enterprise practices.

Calls are redirected by supplying a regular expression to redirect the call to a different routable endpoint host address than that in the original invite. These Regex values are returned in the PolicyGuru ENUM response after the call data is received in the ENUM query.

As mentioned earlier, calls that match an SEP Rule specifying call treatment are terminated or redirected at call setup, preserving your network resources for legitimate business calls. Mid-call termination is not supported.

SEP Contingency Rules

SEP contingency Rules can be created using multiple **When** and **Then** statements, since only the first two **When** and the first **Then** clause are evaluated in Policy processing. This means you can have alternate criteria defined below the clauses that are evaluated, and then simply change the order of the clauses to change the Rule behavior or effectively disable it without having to delete it, in the event of changing conditions. To disable the Rule, define the first **When** clause such that it will never match a call.

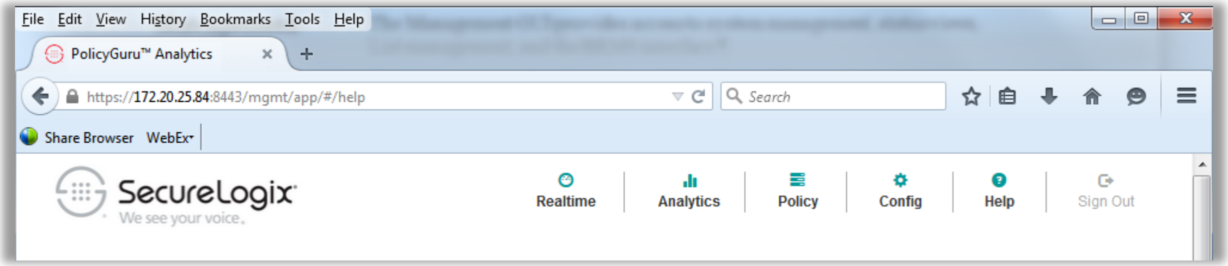
Tour of the Graphical User Interface (GUI)

The PolicyGuru Solution provides a web-based GUI for system and policy management. This GUI comprises two interfaces: the Management GUI and the BRMS GUI.

PolicyGuru® Main Menu

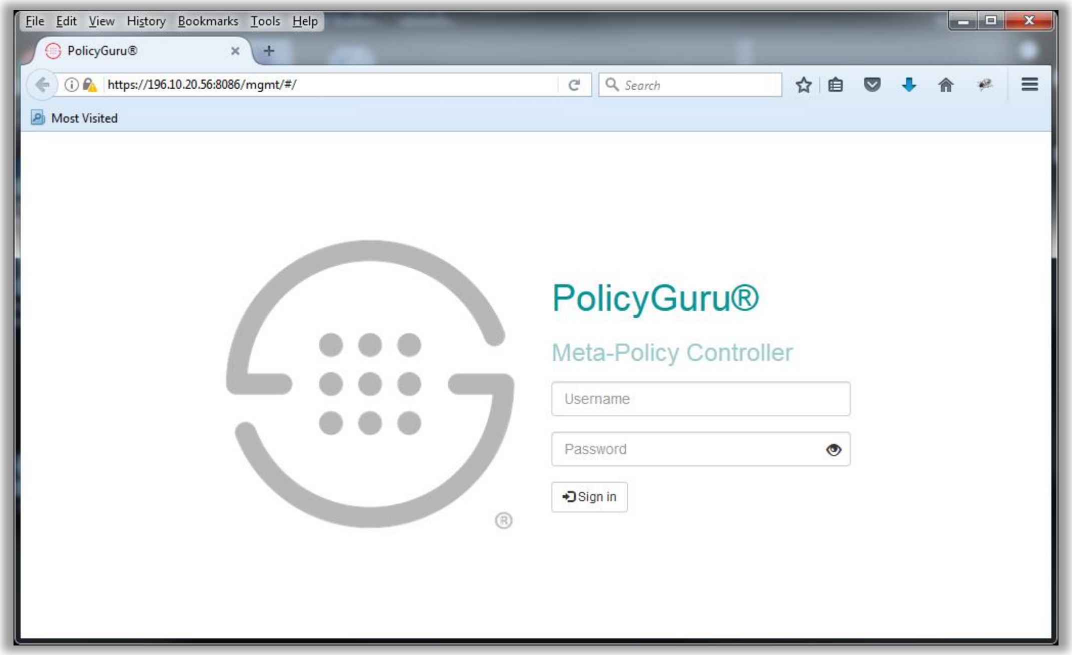
The PolicyGuru main menu shown below provides access for managing all of the features in the solution. This menu is always available, regardless of which tab or feature you are accessing.

(**Note**: The icons available to you may differ if you are using external LDAP, since application permissions are defined by LDAP group memberships, and only icons for those applications for which you have permission are visible to you.) Note that only the icon for the page you are viewing is highlighted; the image below is an illustration to make them more visible in this discussion.



Management GUI

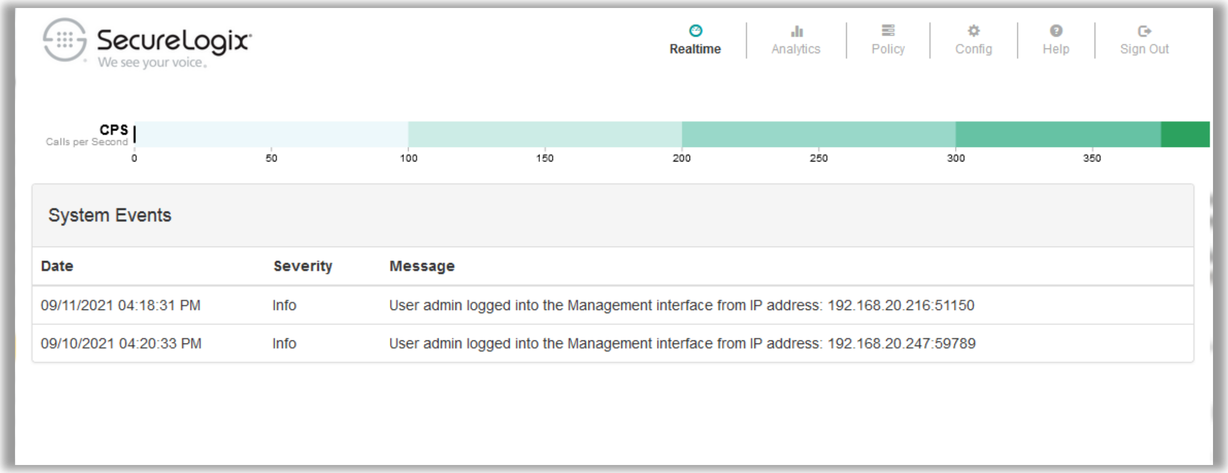
The Management GUI provides access to system management, status views, Analytics reporting, List management, and the Policy interface. The login screen to the Management GUI is shown below.



Note: By default, the GUI uses a self-signed SHA-256-SSL certificate. You can import a customer certificate. See “Importing a Customer Certificate” in the *PolicyGuru® Meta-Policy Controller System Administration Guide* for instructions.

Realtime Tab

The **Realtime** tab shown below appears when you first log in to the Management Interface. It displays real-time system events and current calls per second (CPS) based on ENUM data. (**Note**: The screen that appears when you log in may differ if you are using external LDAP, since application permissions are defined by LDAP group memberships.)



System Events

Date	Severity	Message
09/11/2021 04:18:31 PM	Info	User admin logged into the Management interface from IP address: 192.168.20.216:51150
09/10/2021 04:20:33 PM	Info	User admin logged into the Management interface from IP address: 192.168.20.247:59789

Analytics Tab

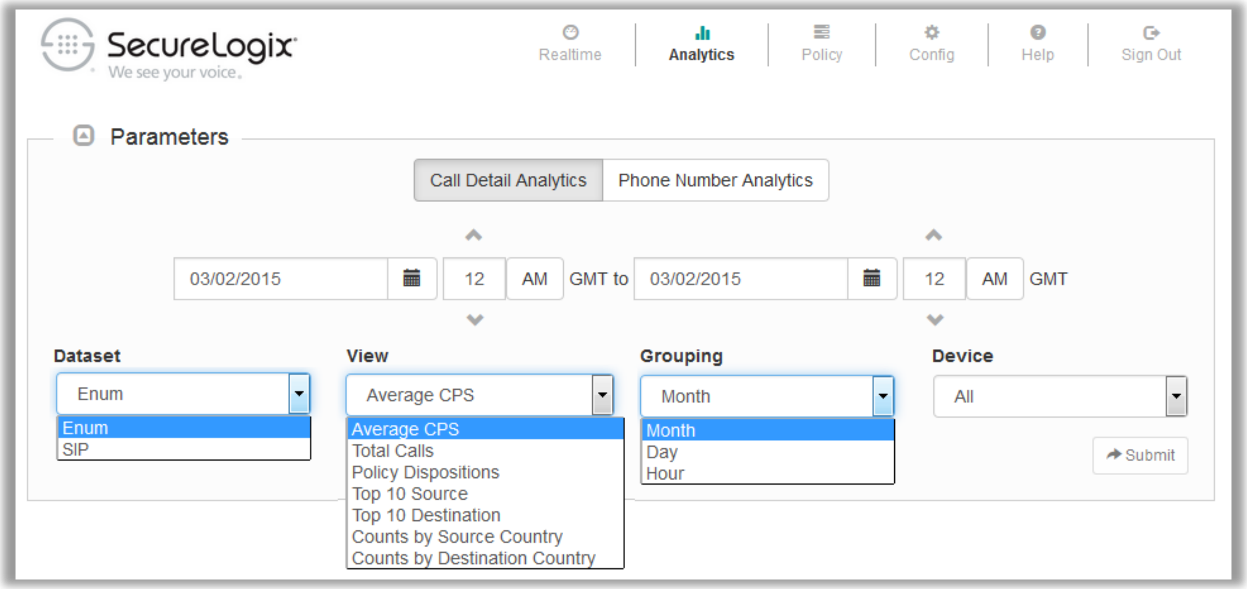
The **Analytics** tab provides a predefined set of drill-down Analytics views that afford a graphical representation of call events. Two types of Analytics views are available: Call Detail Analytics and Phone Number Analytics.

Call Detail Analytics

Call Detail Analytics provides drill-down reporting views. You can choose to view call details for either SIP data derived from the Meta-Data Probe or ENUM call processing data.

Call Detail Analytics views for ENUM data (as shown in the following figure) include:

- Average CPS
- Total Calls
- Policy Dispositions
- Top 10 Source
- Top 10 Destination
- Counts by Source Country
- Counts by Destination Country



03/02/2015 12 AM GMT to 03/02/2015 12 AM GMT

Dataset: Enum

View: Average CPS

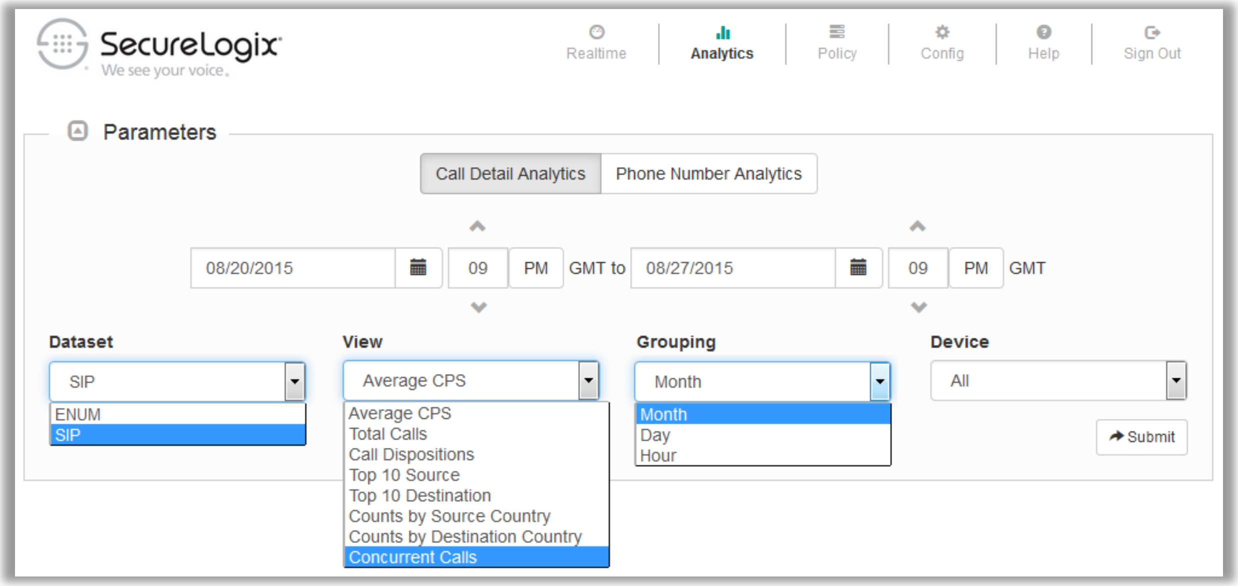
Grouping: Month

Device: All

Figure 2: ENUM Call Detail Analytics Options

Call Detail Analytics for SIP data derived by the Metadata Probe (as shown in the following figure) include:

- Average CPS
- Total Calls
- Call Dispositions
- Top 10 Source
- Top 10 Destination
- Counts by Source Country
- Counts by Destination Country
- Concurrent Calls

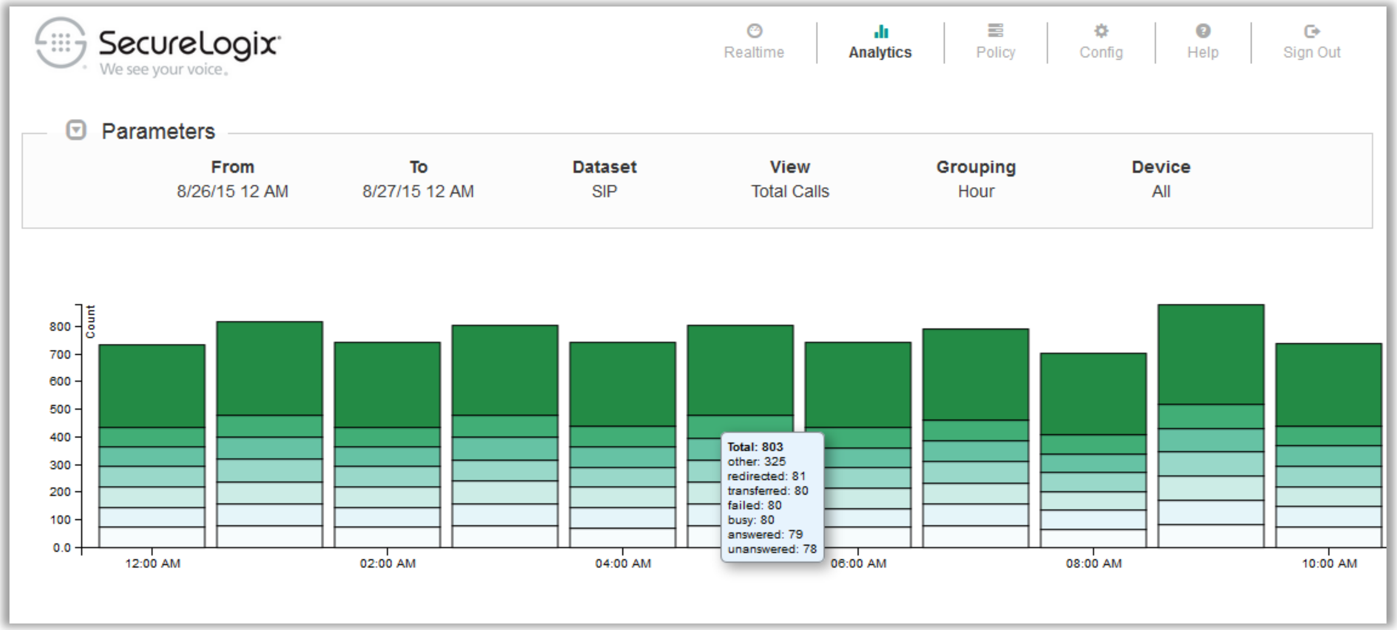


08/20/2015 09 PM GMT to 08/27/2015 09 PM GMT

Dataset	View	Grouping	Device
SIP	Average CPS	Month	All
ENUM SIP	Average CPS Total Calls Call Dispositions Top 10 Source Top 10 Destination Counts by Source Country Counts by Destination Country Concurrent Calls	Month Day Hour	

Figure 3: SIP Call Detail Analytics Options

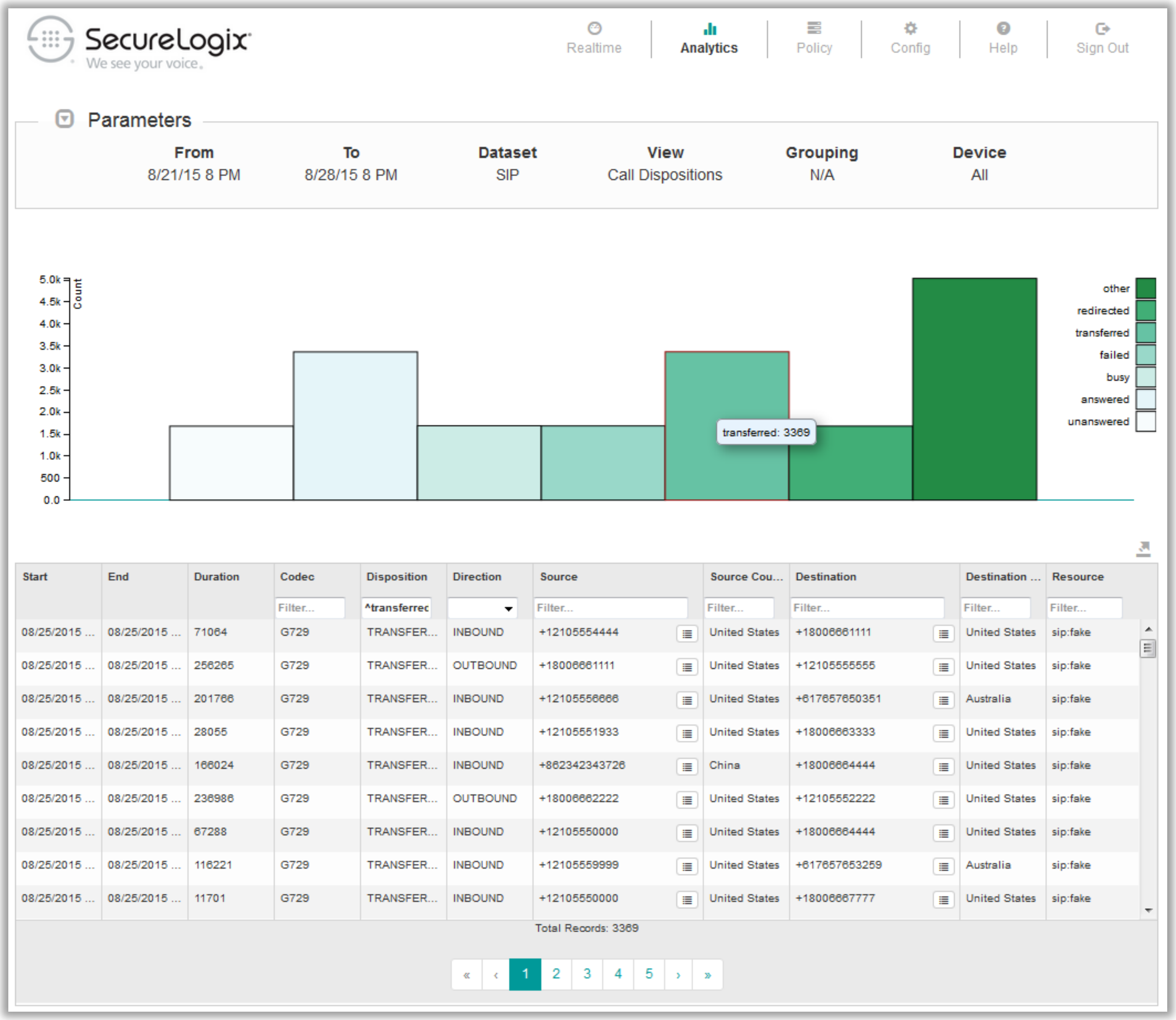
When you select criteria and click **Submit**, a chart view of the resulting analysis appears. When you hover your mouse cursor over the data in the display, details about the data appear as an onscreen tooltip.



From	To	Dataset	View	Grouping	Device
8/26/15 12 AM	8/27/15 12 AM	SIP	Total Calls	Hour	All

Figure 4: SIP Call Detail Analytics—Total Calls

Clicking the display provides a drill-down view of the selected information, as shown in Figure 5, which illustrates the SIP Call Details **Call Dispositions** Analytics view:



Parameters

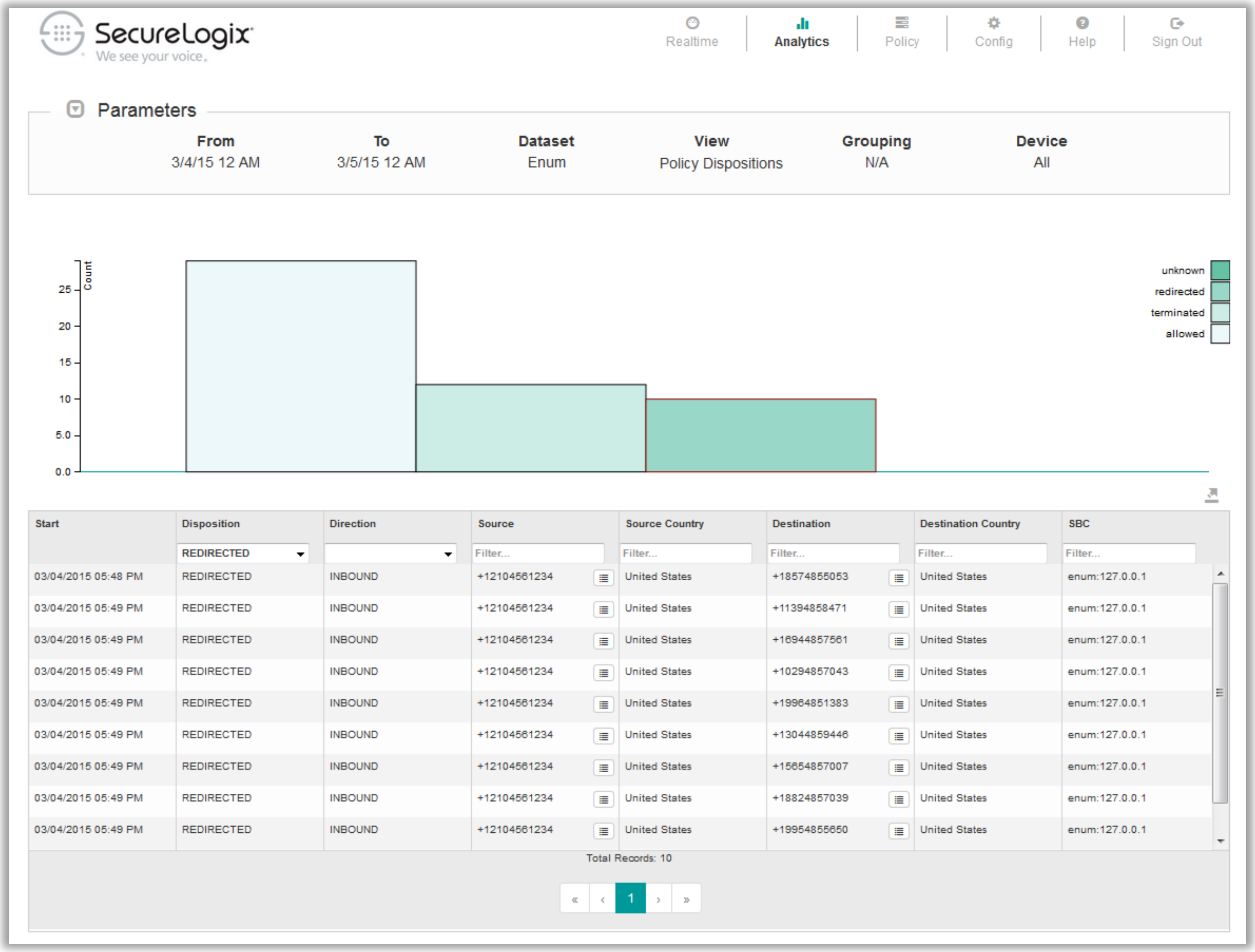
From	To	Dataset	View	Grouping	Device
8/21/15 8 PM	8/28/15 8 PM	SIP	Call Dispositions	N/A	All

Start	End	Duration	Codec	Disposition	Direction	Source	Source Cou...	Destination	Destination ...	Resource
			Filter...	^transferred		Filter...	Filter...	Filter...	Filter...	Filter...
08/25/2015 ...	08/25/2015 ...	71064	G729	TRANSFER...	INBOUND	+12105554444	United States	+18006661111	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	256265	G729	TRANSFER...	OUTBOUND	+18006661111	United States	+12105555555	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	201766	G729	TRANSFER...	INBOUND	+12105556666	United States	+61765760351	Australia	sip:fake
08/25/2015 ...	08/25/2015 ...	28055	G729	TRANSFER...	INBOUND	+12105551933	United States	+18006663333	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	166024	G729	TRANSFER...	INBOUND	+862342343726	China	+18006664444	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	236986	G729	TRANSFER...	OUTBOUND	+18006662222	United States	+12105552222	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	67288	G729	TRANSFER...	INBOUND	+12105550000	United States	+18006664444	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	116221	G729	TRANSFER...	INBOUND	+12105559999	United States	+617657653259	Australia	sip:fake
08/25/2015 ...	08/25/2015 ...	11701	G729	TRANSFER...	INBOUND	+12105550000	United States	+18006667777	United States	sip:fake

Total Records: 3369

« ‹ 1 2 3 4 5 › »

Figure 5: SIP Call Detail Analytics—Call Dispositions



Parameters

From	To	Dataset	View	Grouping	Device
3/4/15 12 AM	3/5/15 12 AM	Enum	Policy Dispositions	N/A	All

Start	Disposition	Direction	Source	Source Country	Destination	Destination Country	SBC
	REDIRECTED		Filter...	Filter...	Filter...	Filter...	Filter...
03/04/2015 05:48 PM	REDIRECTED	INBOUND	+12104561234	United States	+18574855053	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+11394858471	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+16944857561	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+10294857043	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+19964851383	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+13044859446	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+15654857007	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+18824857039	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+19954855650	United States	enum:127.0.0.1

Total Records: 10

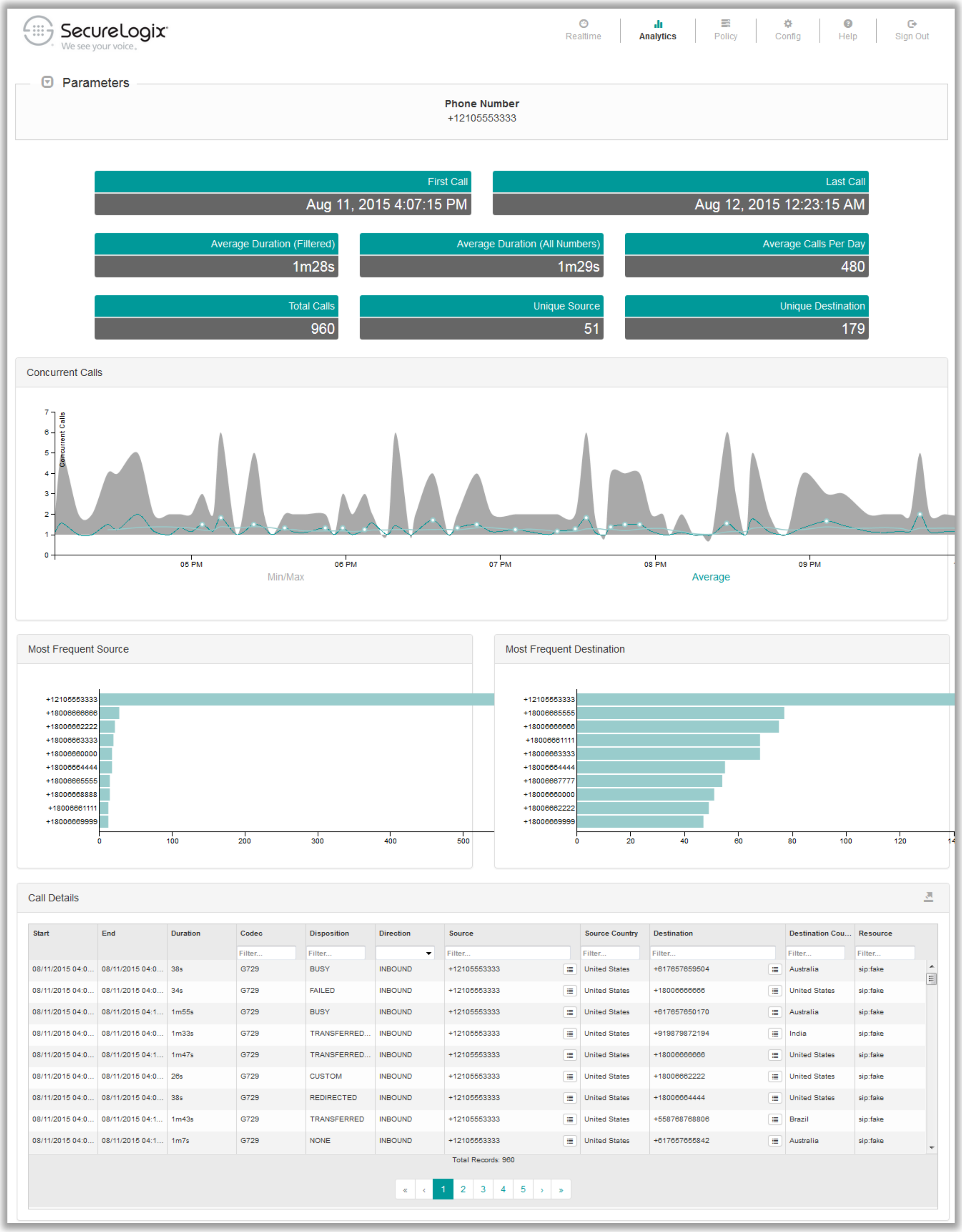
Figure 6: Drill-Down Analytics – Policy Dispositions Call Details

The Analytics application provides more than a view of the information; it also allows you to take action regarding the called and calling numbers provided in the call details. From this drill-down detail view, you can click an icon next to any source or destination and add it to a Whitelist or Blacklist, as appropriate, without needing to toggle between screens.

Note: If you add numbers to Lists used in installed SEP Rules, the Policy must be installed again to effect the changes on the ENUM Server(s) that are enforcing the SEP Policy.

Phone Number Analytics

Phone Number Analytics provides a means for forensic investigation of a specific phone number of interest or a set of phone numbers (such as all of those with +210555) derived from Metadata Probe (SIP) data. You simply type the number of interest in the **Phone Number** field and click **Submit**. All call information for that number is provided in graphs, charts, and tables below the **Search** field. This option is shown in the following figure.



Parameters

Phone Number
+12105553333

First Call	Last Call
Aug 11, 2015 4:07:15 PM	Aug 12, 2015 12:23:15 AM

Average Duration (Filtered)	Average Duration (All Numbers)	Average Calls Per Day
1m28s	1m29s	480

Total Calls	Unique Source	Unique Destination
960	51	179

Concurrent Calls

Most Frequent Source

Most Frequent Destination

Call Details

Start	End	Duration	Codec	Disposition	Direction	Source	Source Country	Destination	Destination Cou...	Resource
			Filter...	Filter...		Filter...	Filter...	Filter...	Filter...	Filter...
08/11/2015 04:0...	08/11/2015 04:0...	38s	G729	BUSY	INBOUND	+12105553333	United States	+617657659504	Australia	sip:fake
08/11/2015 04:0...	08/11/2015 04:0...	34s	G729	FAILED	INBOUND	+12105553333	United States	+18006666666	United States	sip:fake
08/11/2015 04:0...	08/11/2015 04:1...	1m55s	G729	BUSY	INBOUND	+12105553333	United States	+617657650170	Australia	sip:fake
08/11/2015 04:0...	08/11/2015 04:0...	1m33s	G729	TRANSFERRED...	INBOUND	+12105553333	United States	+919879872194	India	sip:fake
08/11/2015 04:0...	08/11/2015 04:1...	1m47s	G729	TRANSFERRED...	INBOUND	+12105553333	United States	+18006666666	United States	sip:fake
08/11/2015 04:0...	08/11/2015 04:0...	26s	G729	CUSTOM	INBOUND	+12105553333	United States	+18006662222	United States	sip:fake
08/11/2015 04:0...	08/11/2015 04:0...	38s	G729	REDIRECTED	INBOUND	+12105553333	United States	+18006664444	United States	sip:fake
08/11/2015 04:0...	08/11/2015 04:1...	1m43s	G729	TRANSFERRED	INBOUND	+12105553333	United States	+558768768806	Brazil	sip:fake
08/11/2015 04:0...	08/11/2015 04:1...	1m7s	G729	NONE	INBOUND	+12105553333	United States	+617657655842	Australia	sip:fake

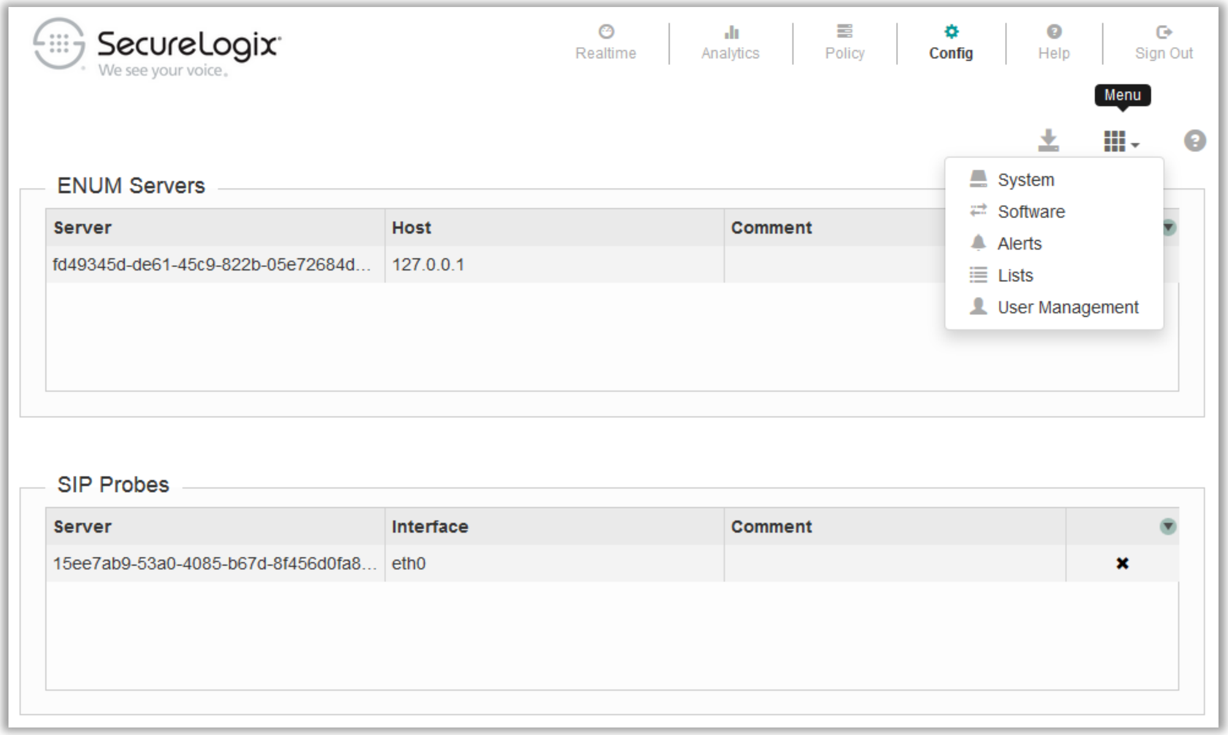
Total Records: 960

Figure 7: Phone Number Analytics

Config Tab

The **Config** tab, shown in Figure 8, provides access to system configuration items, including:

- List management.
- User management.
- A view of which ENUM Servers are connected to this Mediation Server and a field defining the IP address for connections to them.
- A view of which Metadata Probes are connected to this Mediation Server and a field defining the Ethernet port for connections to them.



ENUM Servers

Server	Host	Comment
fd49345d-de61-45c9-822b-05e72684d...	127.0.0.1	

SIP Probes

Server	Interface	Comment	
15ee7ab9-53a0-4085-b67d-8f456d0fa8...	eth0		×

Figure 8: Config Tab

Note: The **Software** option is not used.

Policy Tab

The **Policy** tab provides access to the BRMS Interface from which you manage SEP Policies. Refer to “BRMS GUI” on page 18for an illustration of the BRMS GUI. More information about the BRMS interface is provided later .in this document.

Help Link

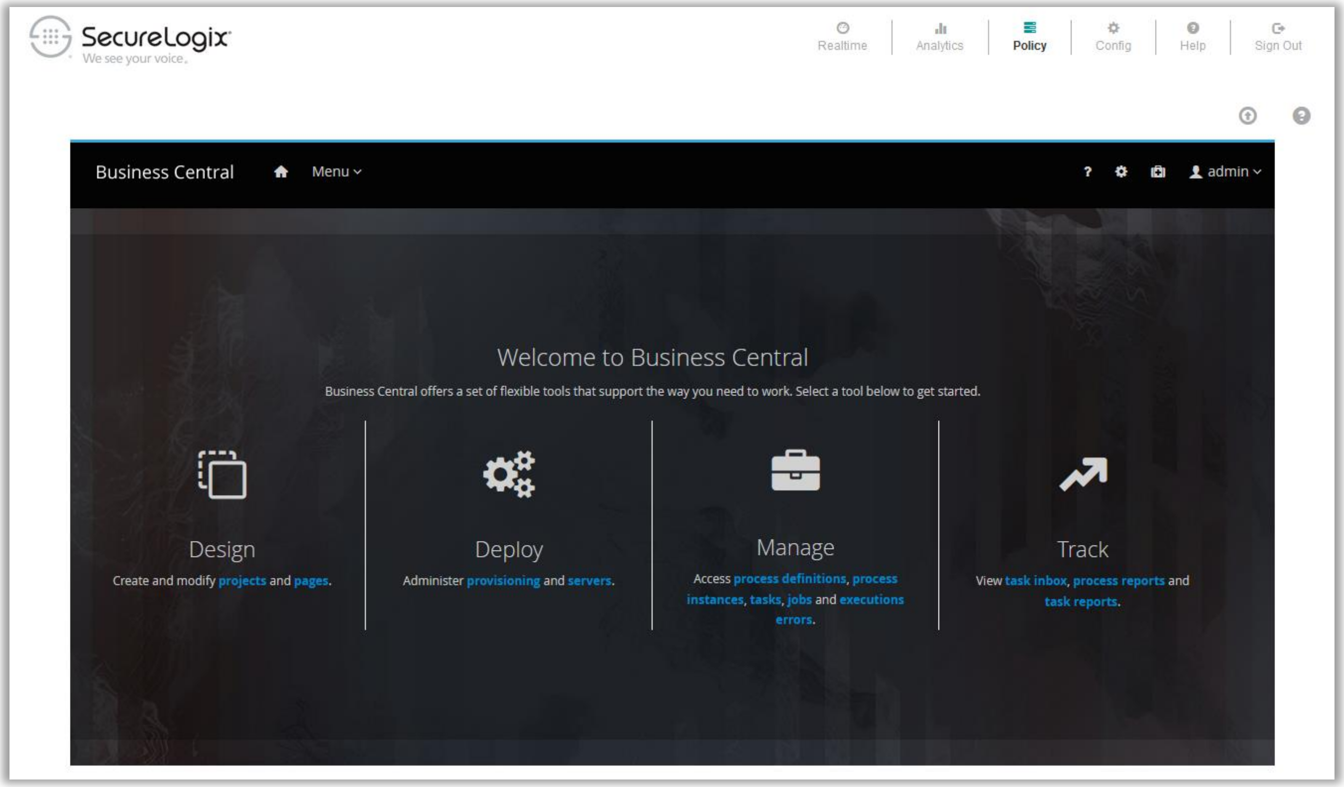
The **Help** link provides contact information for Customer Support and copyright, legal, and licensing notices.

Sign Out Link

The **Sign Out** link signs you out of the PolicyGuru Web Application.

BRMS GUI

When you click **Policy** on the main menu, the following GUI appears from which you access the Policies.



Welcome to Business Central

Business Central offers a set of flexible tools that support the way you need to work. Select a tool below to get started.

Design: Create and modify projects and pages.

Deploy: Administer provisioning and servers.

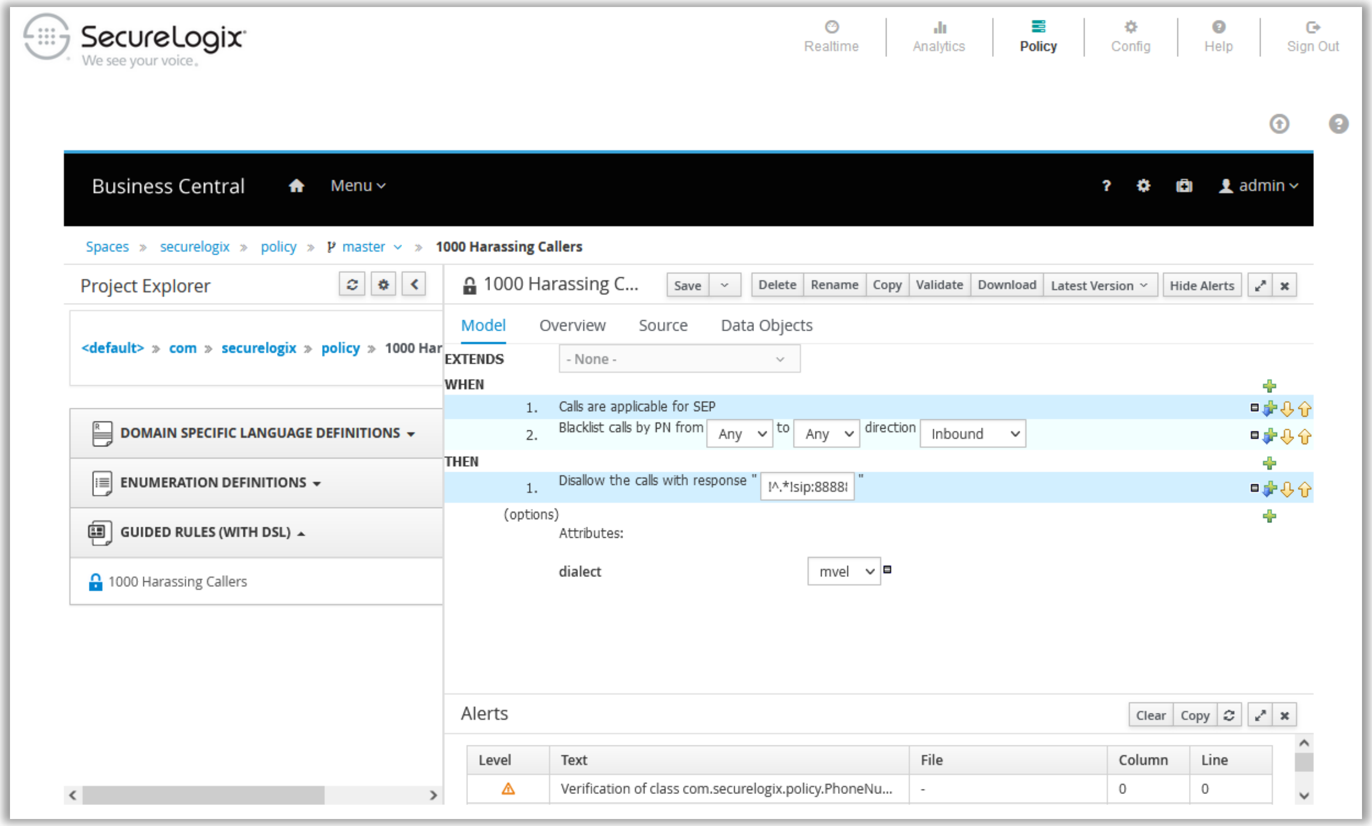
Manage: Access process definitions, process instances, tasks, jobs and executions errors.

Track: View task inbox, process reports and task reports.

Figure 9: BRMS Initial GUI

BRMS Project Authoring GUI

Policy Management is accomplished in the Business Central GUI. This GUI provides a **Project Explorer** pane, a **Rule Editor** pane in which you define SEP Rules, and a set of menus for accomplishing tasks.



EXTENDS - None -

WHEN

1. Calls are applicable for SEP
2. Blacklist calls by PN from Any to Any direction Inbound

THEN

1. Disallow the calls with response " !^.*!sip:8888 "

(options)

Attributes:

dialect mvel

Level	Text	File	Column	Line
⚠	Verification of class com.securelogix.policy.PhoneNu...	-	0	0

Figure 10: BRMS Rule Editor View

Detailed instructions for defining SEP Rules are provided later in this guide.

Simple Event Processing (SEP) Policy Rules and Lists

Defining SEP Rules and Lists

Phone Numbers and URIs for use in SEP Policy Rules are contained in Lists and are defined through the PolicyGuru Management Interface, not the BRMS Interface. To avoid switching between applications, it is recommended you define the applicable List(s) before creating a new Rule.

Each List can be of the type Whitelist or Blacklist and can contain either phone numbers/URIs or regular expressions (Regex), but not both. Regex are used to define ranges and wildcards.

IMPORTANT: In this release, newly added Lists are not automatically available in the Rule editor. To work around this issue each time you manually define or import new Lists::

- Open the **SEP NGP Integration** file in the **Asset** list. Do not make any changes to the file, but do click **Save**. Re-saving this file will prompt the system to requery the available Lists and make them available for subsequent Rule file editing. If a Rule file is already open for editing, close it first and then open the Rule for editing again to update the List drop-downs.

How Lists Are Used In Rules

The **Source** and **Destination** field of a Rule can each include only one List.

When you use Lists in a Rule, they must be of the same type as the Rule. If you specify both **Source** and **Destination**, both Lists must also be of the same type. That is:

- A given Blacklist Rule can contain one of the following means to specify the source(s) and destination(s) to which it applies:
  - PN Blacklist to PN Blacklist, or to **Any**
  - Regex Blacklist to Regex Blacklist, or to **Any**
  - **Any** to PN Blacklist or Regex Blacklist.
  - **Any** to **Any**. This could be used as a Catchall Rule at the end of the Rule list.
- A given Whitelist Rule can contain one of the following means to specify the source(s) and destination(s) to which it applies:
  - PN Whitelist to PN Whitelist, or to **Any**
  - Regex Whitelist to Regex Whitelist, or to **Any**
  - **Any** to PN Whitelist or Regex Whitelist

  - **Any** to **Any**. Such a Rule causes every call to be whitelisted. It could be used as means of “turning off” or “avoiding” all Blacklist Rules. It can be thought of (and used) as a way of disabling Policy processing without actually removing or manually disabling all of the Blacklist Rules.

Managing Lists

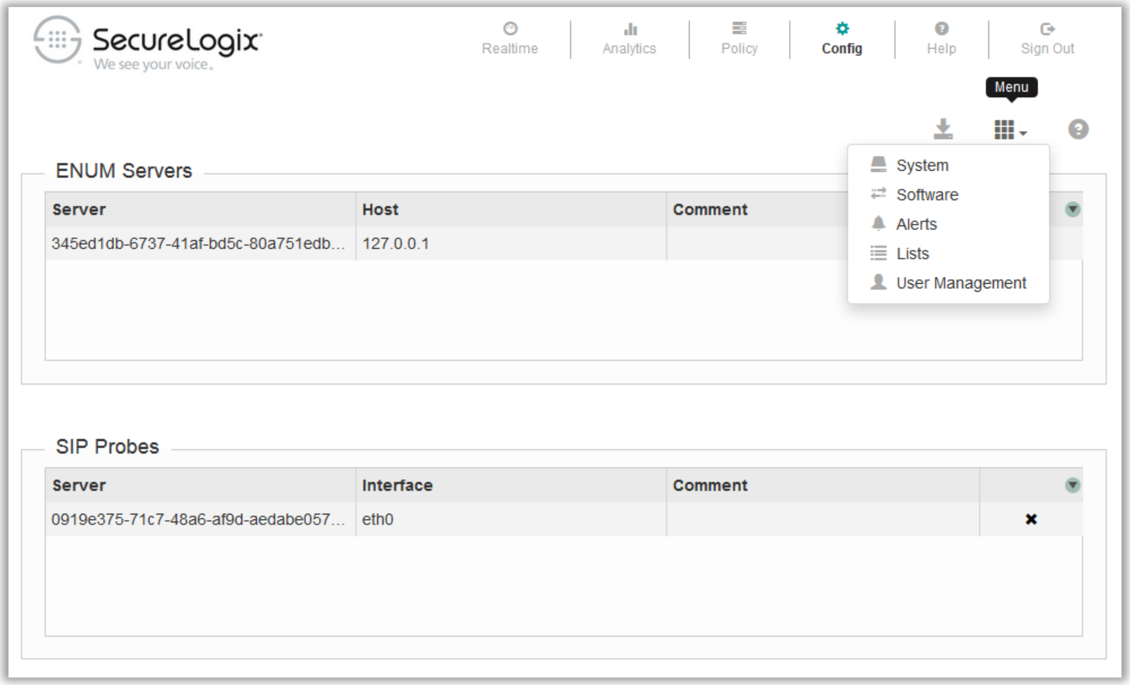
As mentioned earlier, two types of Lists are available:

- **PN Lists**—The values in PN Lists are evaluated via full string matching. PN Lists can contain individual phone numbers/URIs and/or “flag’ strings. Flag strings are character strings that match flags the Phone Number Normalizer has been configured to append or prepend to specified phone numbers received in ENUM Queries. Phone numbers are normalized based on the configuration in the Phone Number Normalizer, prior to Policy processing. This means, for example, that numbers that fail a certain function based on a **libphonenumber** library look up could have a string such as **-INVALID** appended during normalization. You can then create a listing containing the value **INVALID**. When a flagged phone number containing **INVALAD** is compared to a Rule containing this List, the Rule will fire.
- **Regex Lists**—Each listing in a Regex List is specified as a Regex Values in Regex Lists are processed as Regex, by determining if any part of the normalized phone number string matches the regular expression as specified in the listing. In addition to using Regex to define specific number patterns or ranges, you can also use Regex to match on the flagging capability of the Phone Number Normalizer, as described in the bullet above. For example, a Regex value of **badnumber** would match a flag of **this_is_a_badnumber!** appended to a normalized number.

Manually Defining a List

To define a List

1. On the PolicyGuru main menu, click **Config**. The **Config** screen appears.
2. Click the **Configuration Menu** icon and then click **Lists**.



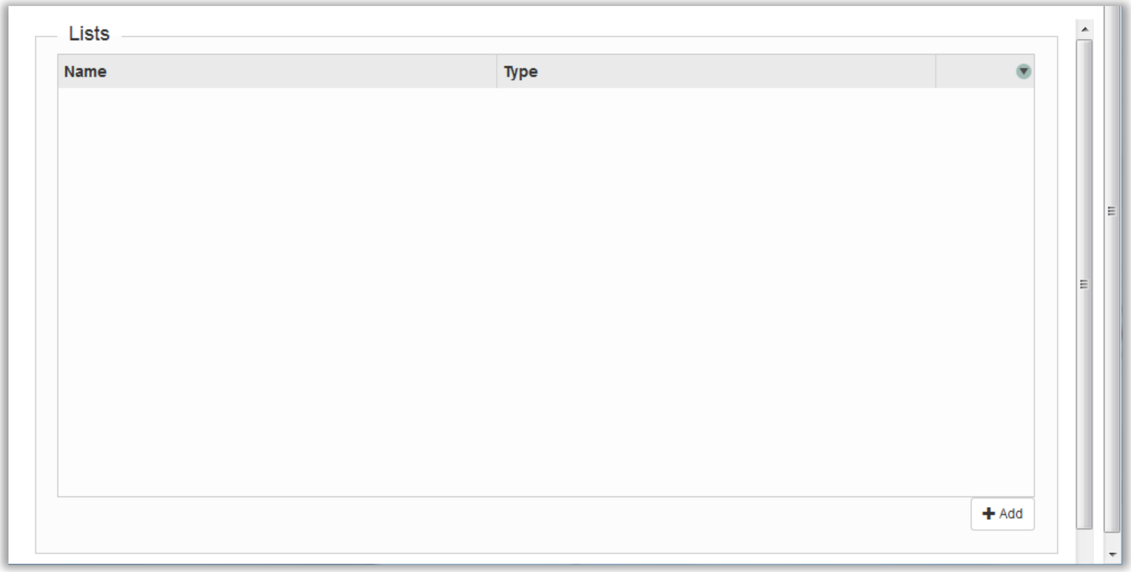
ENUM Servers

Server	Host	Comment
345ed1db-6737-41af-bd5c-80a751edb…	127.0.0.1	

SIP Probes

Server	Interface	Comment	
0919e375-71c7-48a6-af9d-aedabe057…	eth0		✖

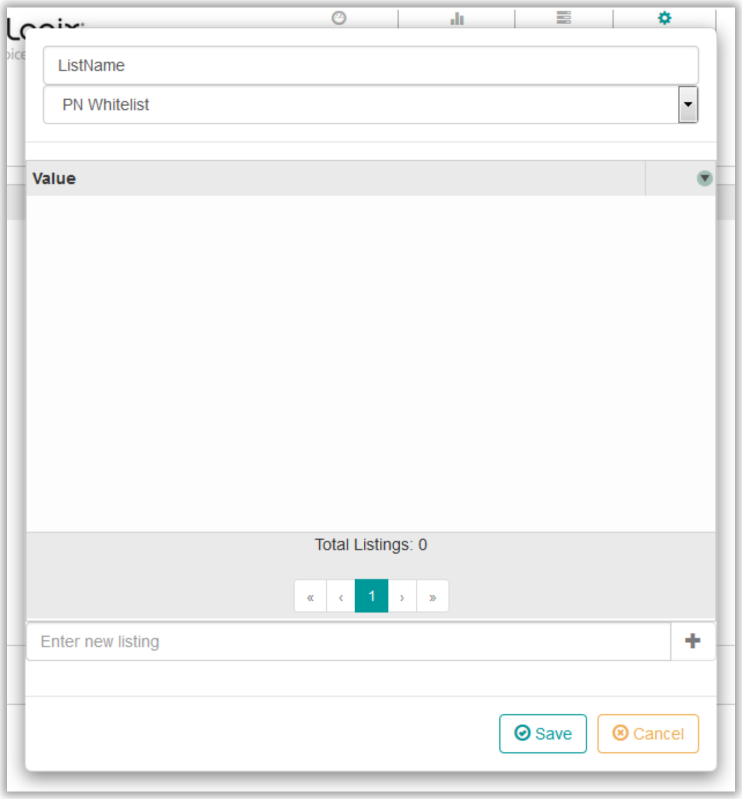
3. The **Lists** screen appears. Click **Add** at the bottom right.



Lists

Name	Type

4. The **List** dialog box appears.



5. In the **ListName** field, type a unique name to identify this list. This is the name by which the list is identified in Rules. The List name can contain any alphanumeric and special characters EXCEPT spaces, double quotes, parenthesis, periods, forward slash, back slash, pound sign, question mark, comma, or percent sign and can be up to 255 characters in length.

   For example, type: `Harassing_Callers`.

Tip: You may want to append _SRC to the name of Lists intended for the **Source** field and _DST to those intended for the **Destination** field of Rules.

6. Click the down arrow in the field below the **ListName** field and select the type of List you are defining:

   - **PN Whitelist**—Contains single or multiple phone numbers/URIs. They are only available for use in Whitelist Rules.
   - **PN Blacklist**—Contains single or multiple phone numbers/URIs. They are only available for use in Blacklist Rules.
   - **Regex Whitelist**—Contains regular expressions representing Ranges or Wildcards. They are only available for use in Whitelist Rules.
   - **Regex Blacklist**—Contains regular expressions representing Ranges or Wildcards. They are only available for Blacklist Rules.

   IMPORTANT: You cannot mix Regex and phone numbers in the same List. PN Lists can include only phone numbers/URIs(and strings that are processed like phone numbers via full string matching), not regular expressions. Regex Lists can include only regular expressions, not phone numbers/URIs.

Note: See also “How Lists are Matched in Policy Processing” on page 9 for important details on what Lists can contain and how they are processed.

IMPORTANT: You <u>cannot change the List type</u> after you save the List. Ensure that you have the correct type selected before you save the List.

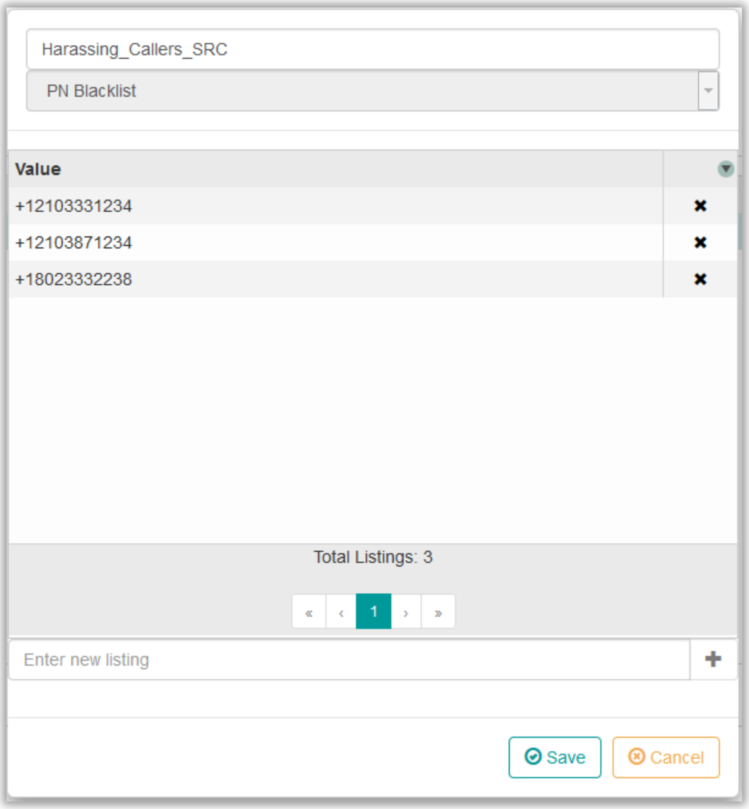
7. In the **Enter new listing** field, type the phone number/URI or regular expression.

Note: You cannot edit an entry once is has been added. If it contains an error, retype it as a new entry and then delete the mistaken one.

- Phone numbers are typically fully qualified, of the form **`+cclocalnumber`** (i.e.,+12104029669) without spaces or punctuation. The normalized numbers seen by the ENUM Server must match the List entries for a Rule to fire on that number. This means they must match the normalized formats defined by your Phone Number Normalizer script.
- See "List Regex Examples" on page 27.

8. Click the PLUS SIGN (**+**)next to the entry or press ENTER to add it to the List. Repeat for each phone number/URI or regular expression that applies to this List. The entries appear in the **Value** field and are searchable and sortable.

TIP: To search for a Listing, click the green arrow to the right of **Value**; to sort the List, click the gray area of the **Value** heading.



Harassing_Callers_SRC

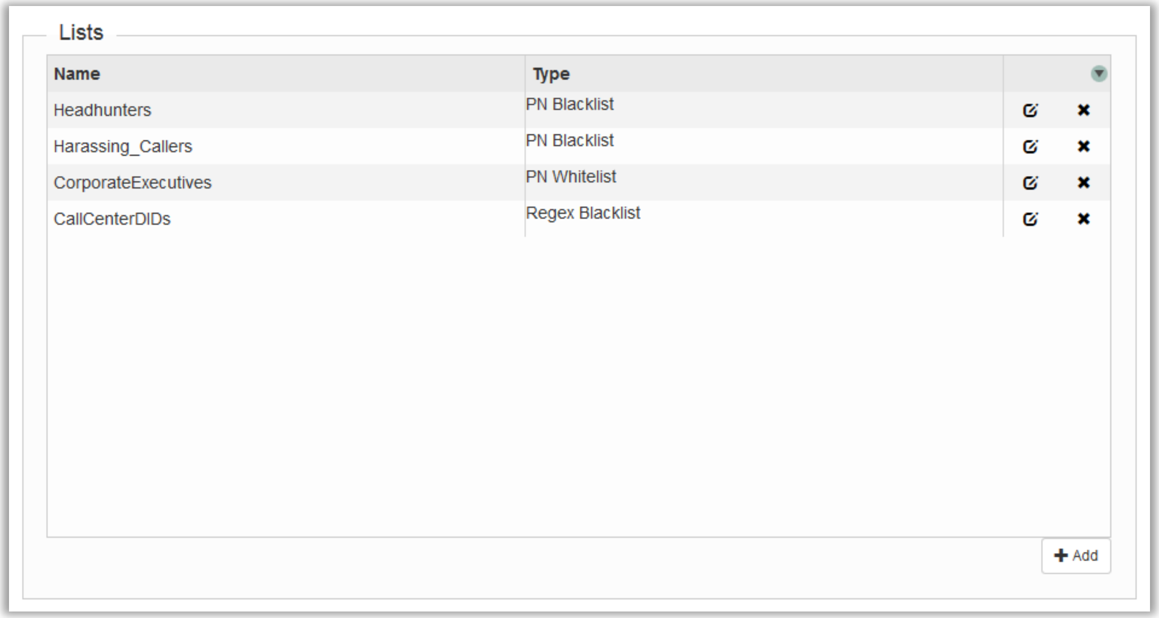
PN Blacklist

Value	
+12103331234	✖
+12103871234	✖
+18023332238	✖

Total Listings: 3

Enter new listing

9. When you have added all of the entries, click **Save**. The List appears in the **Lists** dialog box and is available in SEP Policy Rules.



Lists

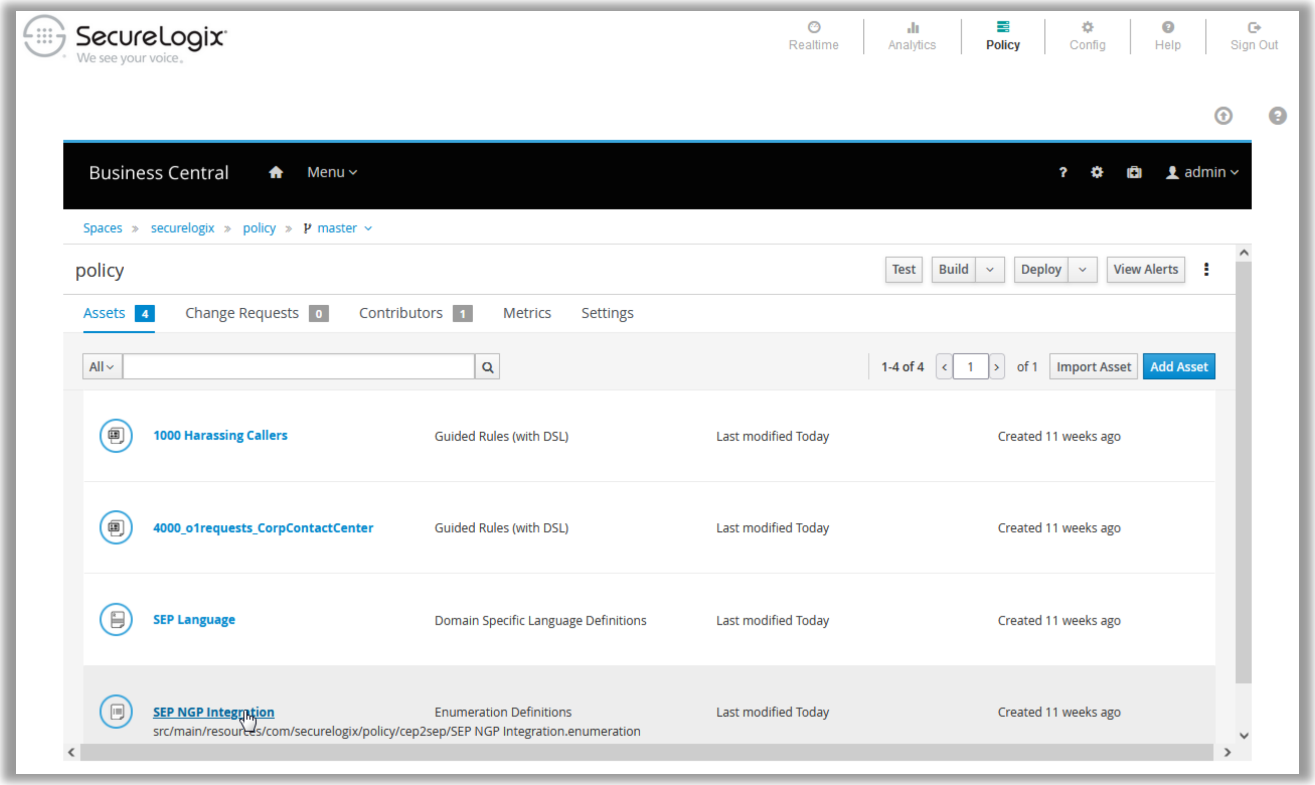
Name	Type	
Headhunters	PN Blacklist	
Harassing_Callers	PN Blacklist	
CorporateExecutives	PN Whitelist	
CallCenterDIDs	Regex Blacklist	

Note: While the system appears to allow you to add a duplicate entry to a List, the duplication is resolved when you add it, as evidenced by the count shown. Each entry in a given List is guaranteed unique.

- To search for a List, click the green arrow to the right of the heading row.
- To sort the Lists, by name or type, click in the gray arrow of the applicable heading.
- To edit a List, click the checkmark in its row or double-click the List to open it. **Note**: You can add and remove entries and change the List name, but you cannot edit existing entries or change the List type. To correct an erroneous entry, type the corrected version as a new entry and delete the erroneous one.
- To delete a List, click the **X** in its row. You are prompted to save changes.

IMPORTANT: In this release, newly added Lists are not automatically available in the Rule editor. To work around this issue each time you manually define or import new Lists::

- In Business Central, open the **SEP NGP Integration** file from the **Asset** list of the **securelogix | policy** project. Do not make any changes to the file, but do click **Save**. Re-saving this file will prompt the system to requery the available Lists and make them available for subsequent Rule file editing. If a Rule file is already open for editing, close it first and then open the Rule for editing again to update the List drop-downs.



1000 Harassing Callers	Guided Rules (with DSL)	Last modified Today	Created 11 weeks ago
4000_o1requests_CorpContactCenter	Guided Rules (with DSL)	Last modified Today	Created 11 weeks ago
SEP Language	Domain Specific Language Definitions	Last modified Today	Created 11 weeks ago
SEP NGP Integration src/main/resources/com/securelogix/policy/cep2sep/SEP NGP Integration.enumeration	Enumeration Definitions	Last modified Today	Created 11 weeks ago

List Regex Examples

Below are some example regular expressions for use in Regex Whitelists and Regex Blacklists:

1. Wild Card for Country code 235:

   `^\+235`

2. Wild Card for NNX 766:

   `^\+1[0-9]{3}766`

3. Ranges:

   `^\+1309766[1-9]`
   would match 309 766 1000 to 309 766 9999

Searching for a List

To search for a specific List

1. On the **Lists** screen, click the green arrow **Search** icon to the right of the **Name | Type** header row.
2. In the **Search** field, type any part of the List name.
3. To clear the search, clear the **Search** field.

Sorting the Set of Lists

The set of Lists on the **Lists** screen can be sorted by name or by type.

To sort the Lists screen

- Click the column heading by which you want to sort: **Name** or **Type**.

Searching for Specific Entries in a List

To search for a specific entry in a List

1. On the **Lists** screen, click the click the **Edit** icon at the right of the row containing the List. The **List** dialog box appears.
2. At the right of the **Value** heading row, click the green arrow **Search** icon.
3. In the **Search** field, type the complete entry for which you want to search and then press ENTER. The view is filtered to show only the matching entry.
4. To clear the search, clear the **Search** field and then press ENTER.

Sorting a List

List entries appear by default in ascending order.

To sort a List in ascending or descending order

- In an open List, click the **Value** heading row. Each click toggles the sort order.

Editing a List

Important: When you update a List that is used in an installed Rule, you must click **Commit** to push the changes to the ENUM Server before the changes take effect in the installed Rule.

Note that you can add and delete entries and change the List name, but you cannot edit existing entries nor change the List type.

To edit a List

1. On the **Lists** screen, click the **Edit** icon at the right of the row containing the List you want to edit. The **List** dialog box appears containing the entries in the List.
   - ✖ To delete an entry, click the **Delete** icon at the right of the row containing the entry. You are prompted to save changes.
   - + To add an entry, type it in the **Enter new listing** field and then click the PLUS SIGN (**+**) or press ENTER.
   - To rename a List, type the new name. **Note**: After renaming, reselect the List in any Rules where it is used.
   - ▼ To search for listings, click the down arrow at the right of the **Value** heading row.
   - To sort the Listings, click in the gray area of the **Value** heading. Each click toggles the sort order.
2. After making all of your changes, click **Save**.
3. If you are making changes to one or more Lists used in Policy, you must reinstall the Policy before the changes take effect on the ENUM Servers enforcing the policy. Wait 15 to 30 seconds after clicking **Save**, click the **Policy** tab, and then click the **Commit** icon to push the changes to the ENUM Servers.

View the **Realtime** tab to view the progress of the Policy push. You should see two messages appear.

First, you will see a message similar to the following:

```
08/19/2015 05:40:37 PM   Info    PolicyPublisherServiceBean - Sending COMMIT version 35
```

This message is shortly followed by a message similar to the following for *each* ENUM Server belonging to this Mediation Server:

```
                                 PolicyVersionCheckServiceBean - Successfully updated app 31695fbf-4b7e-
08/19/2015 05:40:37 PM   Info    449e-aa0c-59969f26d3ab_02:3a:52:3f:61:a7 to policy version 35
```

Deleting a List

To delete a List

1. In the **Lists** screen, click the **X** to the right of the row containing the **List** you want to delete.
2. At the upper right, click the **Save** icon. The List is not deleted from the database until you click **Save**.
4. **IMPORTANT** If you delete a List used in SEP Policy, reinstall the SEP Policy for the change to take effect on the ENUM Servers. Wait 15 to 30 seconds after clicking **Save**, click the **Policy** tab, and then click the **Commit** icon to push the changes to the ENUM Servers.

Importing a List

For large Lists, SecureLogix provides utility scripts for importing a new List, adding entries to an existing List, and deleting Lists. You can initially populate Lists by running a script to import entries from a flat file into the database. You can then maintain the List through the GUI. You can also import additional Listings into an existing List. You can import either individual, static Listings or ranges. Range import results in one individual Listing for every number included in the specified range.

Run these scripts on the Mediation Server.

Note: These import scripts are not included in the default installation. Contact SecureLogix Technical Support to obtain a copy of the scripts.

IMPORTANT: In this release, newly added Lists are not automatically available in the Rule editor. To work around this issue <u>each time</u> you manually define or import new Lists::

- Open the **SEP NGP Integration** file in the **Asset** list. Do not make any changes to the file, but do click **Save**. Re-saving this file will prompt the system to requery the available Lists and make them available for subsequent Rule file editing. If a Rule file is already open for editing, close it first and then open the Rule for editing again to update the List drop-downs.

Static List Import

A static import list consists of a text file with one normalized phone number/URI or Regex entry per line.

Defining a Static Import File

A static import List consists of a text file with one normalized phone number/URI or Regex entry per line. These must be in exactly the form you want them to appear in the List, in the same way as with manual entry as described above.

Do not leave any blank lines, add comments, or use tabs in the file. Blank lines and comments are imported as Listings. Ensure that each number appears only once in the List; duplicates cause the script to error out.

Recall that a given List can contain either phone numbers/URIs OR Regex entries, but not both.

You specify the filename of the import file when you run the import script. See “Importing a New Static List” below for instructions.

Importing a New Static List

Static List import imports a set of individual, fully qualified phone numbers or a set of Regex from a text file to create a new List of the applicable type. Run this scripts on the Mediation Server.. See “Defining a Static Import File” above for instructions for creating the file.

To import a static List from a text file

1. Obtain a copy of **createlist_file_2.1.sh**. This script is not included in the default installation. Contact SecureLogix Technical Support to obtain a copy. A sample appears in “Appendix A: List Import Scripts” on page 74.
2. SSH to the Mediation Server and place the script there.
3. Define a text file containing the phone numbers/URIs or Regex you want included in the List. See “Defining a Static Import File” above.
4. SSH to the Mediation Server and place the text file in the directory where you placed the script.
5. Execute the script from a command line. Usage (typed on one line) is:

```
./createlist_file_2.1.sh <server> <list name> <list type> <list file> [batch size]
[username] [password]
```

Where:

server—Mediation Server IP Address.

listname—The name to give the List in the GUI.

- **IMPORTANT**: **listname** can contain any alphanumeric and special characters EXCEPT spaces, double quotes, parenthesis, periods, forward slash, back slash, pound sign, question mark, comma, or percent sign and can be up to 255 characters in length.

listtype—A digit denoting the type of List to create:

0 – PN Whitelist

1 – PN Blacklist

2 – Regex Whitelist

3 – Regex Blacklist

listfile—The filename name of import file.

batch size—(*optional*) the number of listings to submit at a time. If omitted, the batch size defaults to 100.

username—(*optional*) The username for PolicyGuru application login. If omitted from the command line, you will be prompted for the username during script execution.

password—(*optional*) The password for the PolicyGuru application username you supply. If omitted from the command line, you will be prompted for the password during script execution.

6. When script execution completes, the new List appears in the **Lists** screen in the Management GUI.

Range List Import

Range imports to a List are useful if you have one or more large, contiguous ranges of phone numbers to import. Range import results in one individual Listing for every number included in the specified range. Note that with this script, the Listing created as the beginning of the range will end with some number of zeros and the Listings will increment from there based on your supplied values. Examples are provided below. Run this script on the Mediation Server.

Importing a New List from a Range

To import a Range to a List

1. Obtain a copy of the **createlist_range_2.1.sh** file. This script is not included in the default installation. Contact SecureLogix Technical Support to obtain a copy. A sample appears in “Appendix A: List Import Scripts” on page 74.
2. SSH to the Mediation Server and place the script there.
3. Execute the script from a command line. Usage (typed on one line) is:

```
./createlist_range_2.1.sh <server> <list name> <list type> <prefix> <value length>
<count> [batch size] [username] [password]
```

Where:

server—Mediation Server IP Address..

listname—The name to give the List in the GUI.

- **IMPORTANT**: **listname** can contain any alphanumeric and special characters EXCEPT spaces, double quotes, parenthesis, periods, forward slash, back slash, pound sign, question mark, comma, or percent sign and can be up to 255 characters in length.

listtype—A digit denoting the type of List to create:

0 – PN Whitelist

1 – PN Blacklist

2 – Regex Whitelist

3 – Regex Blacklist

prefix—The starting digits of each range entry (for example, +121055500.)

valuelength—The number of additional characters to be added to the end of the specified prefix, for example, 2. Using the example value above and a count of 100, this would denote adding the digits 00-99 to create a range of 100 listings from +12105550000 to +12105550099.

count—The number of List entries to create (for example, 100).

batch size—(*optional*) the number of listings to submit at a time. If omitted, the batch size defaults to 100.

username—(*optional*) The username for PolicyGuru application login. If omitted from the command line, you will be prompted for the username during script execution.

password—(*optional*) The password for the PolicyGuru application username you supply. If omitted from the command line, you will be prompted for the password during script execution.

4. When script execution completes, the new List appears in the **Lists** screen in the Management GUI.

Importing Listings Into An Existing List

You can import additional Listings from a text file into an existing List. See "Defining a Static Import File" on page 30 for instructions for creating the file. Run this script on the Mediation Server.

To import Listings into an existing List

1. Obtain a copy of the **addtolist_file_2.1.sh** file. This script is not included in the default installation. Contact SecureLogix Technical Support to obtain a copy. A sample appears in "Appendix A: List Import Scripts" on page 74.
2. SSH to the Mediation Server and place the script there.
3. Execute the script from a command line. Usage (typed on one line) is:

```
./addtolist_file_2.1.sh <server> <list name> <list file> [batch size] [username]
[password]
```

Where:

server—Mediation Server IP Address.

listname—The name of the existing List that the Listings will be added to.

listfile—The filename name of import file.

batch size—(*optional*) the number of listings to submit at a time. If omitted, the batch size defaults to 100.

username—(*optional*) The username for PolicyGuru application login. If omitted from the command line, you will be prompted for the username during script execution.

password—(*optional*) The password for the PolicyGuru application username you supply. If omitted from the command line, you will be prompted for the password during script execution.

5. When script execution completes, the Listings in the file have been added to the existing List. .
6. If the List is used in SEP Policy, reinstall the SEP Policy for the change to take effect on the ENUM Servers.

Delete List Script

The Delete List script is a utility to enable you to easily delete a large List you are no longer using. Attempting to delete a very large List using the GUI can time out before the operation succeeds.

To delete a large List

1. Obtain a copy of the **deletelist_2.1.sh** file. This script is not included in the default installation. Contact SecureLogix Technical Support to obtain a copy. A sample appears in “Appendix A: List Import Scripts” on page 74.
2. SSH to the Mediation Server and place the script there.
3. Execute the script from a command line. Usage is:

```
./deletelist_2.1.sh <server> <list name> [batch size] [username] [password]
```

Where:

server—Mediation Server IP Address.

listname—The name of the existing List to be deleted from the system.

batch size—(*optional*) the number of listings to submit at a time. If omitted, the batch size defaults to 100.

username—(*optional*) The username for PolicyGuru application login. If omitted from the command line, you will be prompted for the username during script execution.

password—(*optional*) The password for the PolicyGuru application username you supply. If omitted from the command line, you will be prompted for the password during script execution.

4. If the List you deleted was being used in installed SEP Rules and you are replacing it with a new List, edit any SEP Policy Rules the List you deleted was used in, add the new List to the Rules, and reinstall the Policy.

First-Time Business Central Policy Setup

At this time, the setup of Business Central and SEP Policy is a manual process. This will be carried out automatically during installation in a subsequent release.

Before setting up Business Central SEP policy, the following Error message will be displayed in the **Realtime** System Events. Also, before this setup, the **securelogix | policy** project will not be present in the Business Central GUI.

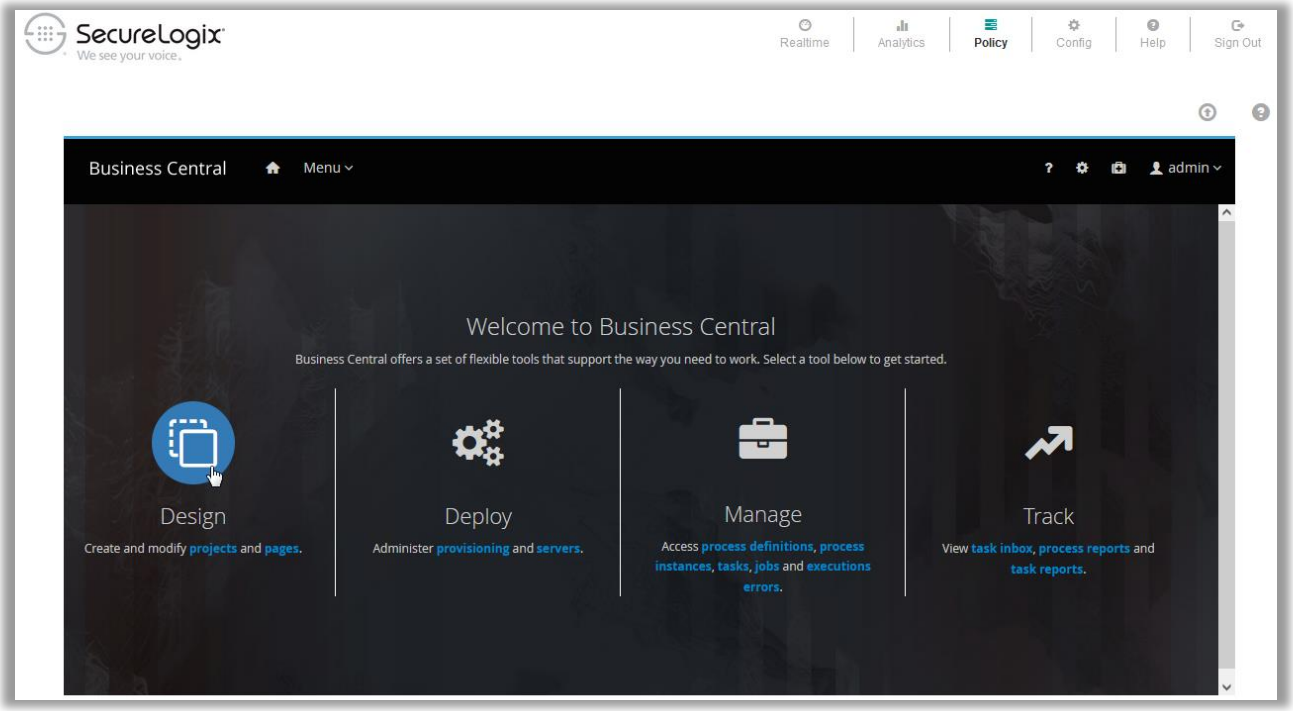
```
DroolsGitSyncronizer: Failed to execute drools
pull from
[/opt/ngp/bin/.niogit/securelogix/policy.git] to
[/opt/ngp/sep_drools] : Invalid remote: origin
```

Perform the steps below to for first-time setup of SEP policy:

1. With the Mediation Server stopped, replace the existing **/opt/ngp/bin/.niogit folder** with the contents of the provided **NIOGIT.tar.gz** file. This is done by untarring the file in the **/opt/ngp/bin folder**. Also untar the provided **M2.tar.gz** file in the **/root folder** (contents will go into the **/root/.m2** folder).

2. Start the Mediation Server, log in to the Web GUI, and click the **Policy** tab. Log in to the Business Central GUI using PolicyGuru System credentials.

3. On the Business Central home page, click the **Design** link.



Welcome to Business Central

Business Central offers a set of flexible tools that support the way you need to work. Select a tool below to get started.

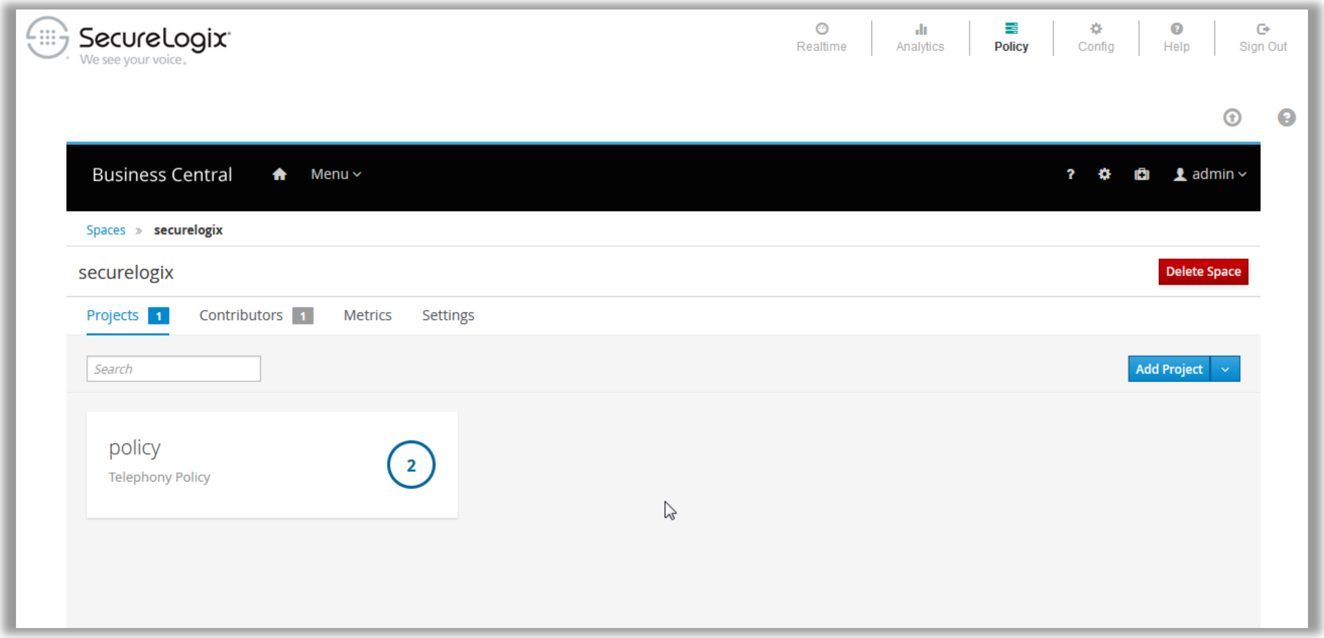
Design
Create and modify projects and pages.

Deploy
Administer provisioning and servers.

Manage
Access process definitions, process instances, tasks, jobs and executions errors.

Track
View task inbox, process reports and task reports.

4. Click **Projects**.

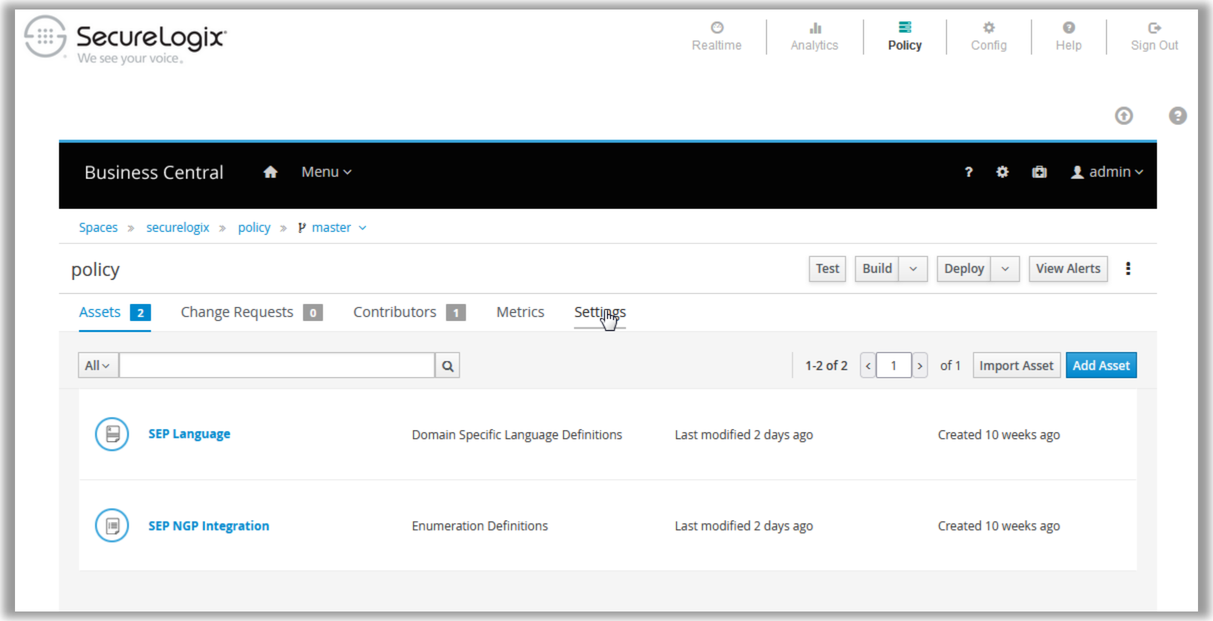


securelogix

policy
Telephony Policy
2

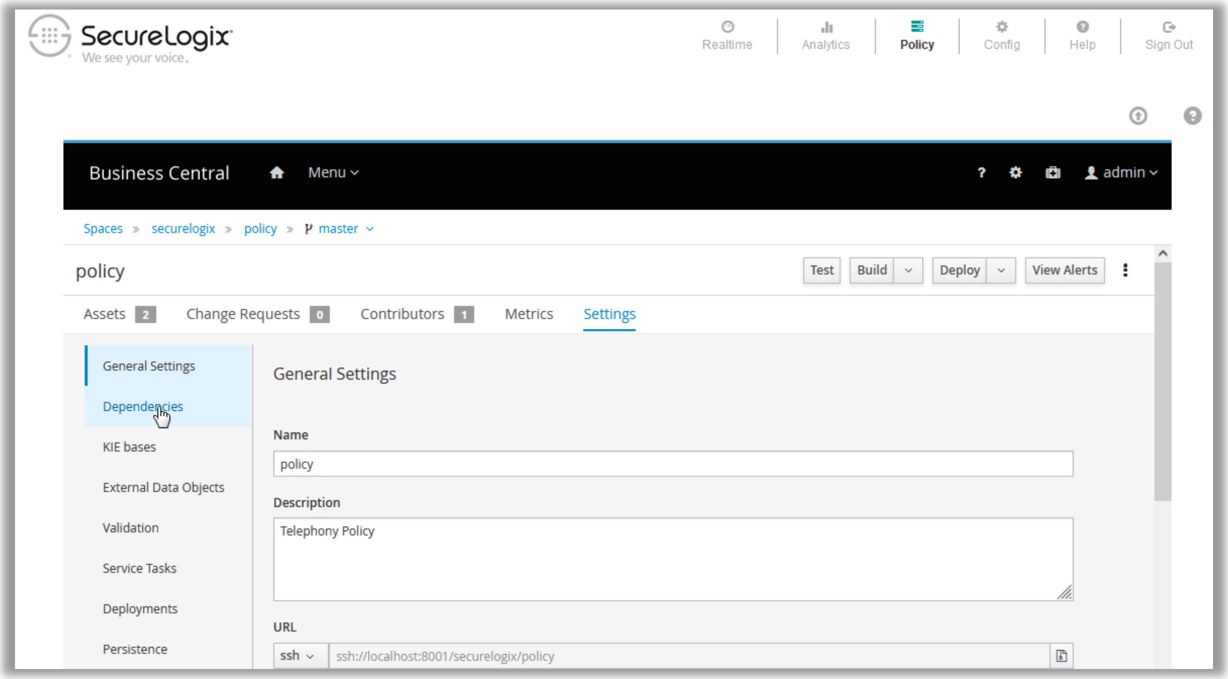
5. By default, you may either see the **MySpace** space or the **securelogix** space.

- For **MySpace**, click the blue **Spaces** heading just under the black menu bar. There are two available spaces: **MySpace** and **securelogix**. Click the **securelogix** space, and then click the **policy** project. Under **policy** is a list of Assets, including the **SEP Language** and **SLC NGP Integration** files.
- For the **securelogix** space, simply click the **policy** project. Under **policy** is a list of Assets, including the **SEP Language** and **SLC NGP Integration** files.



SEP Language	Domain Specific Language Definitions	Last modified 2 days ago	Created 10 weeks ago
SEP NGP Integration	Enumeration Definitions	Last modified 2 days ago	Created 10 weeks ago

6. On the menu bar for the **policy** project, click **Settings**, and then in the menu tree below that on the left side, click **Dependencies**.



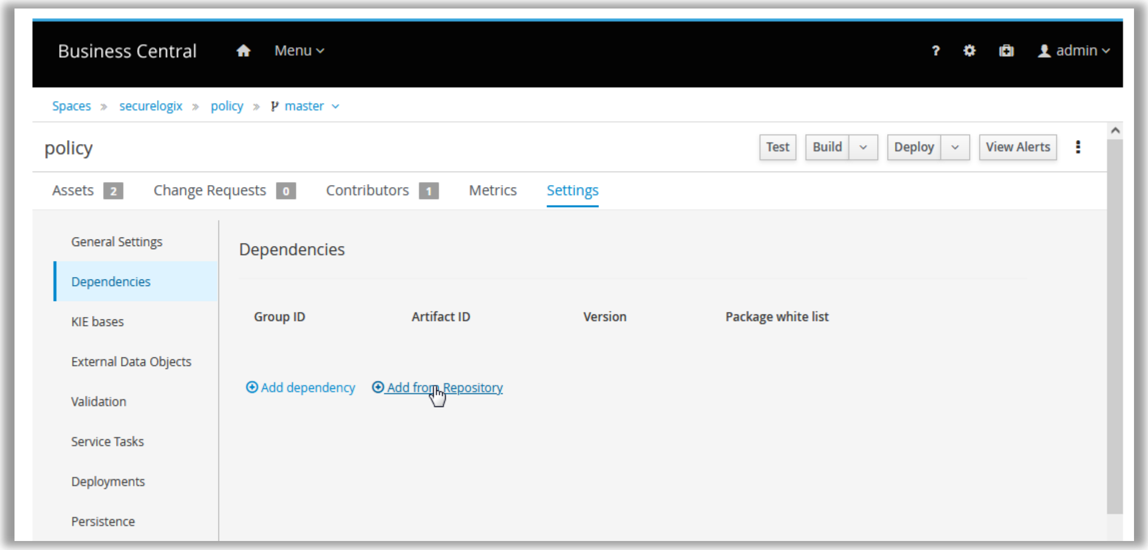
General Settings

Name: policy

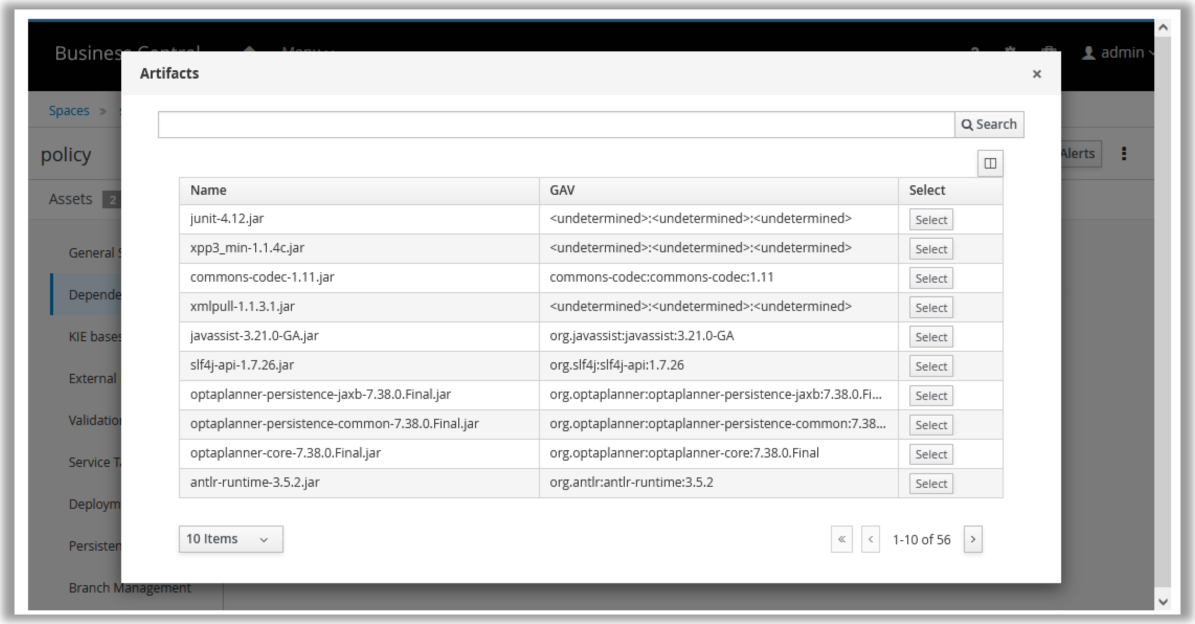
Description: Telephony Policy

URL: ssh ssh://localhost:8001/securelogix/policy

7. The list of **Dependencies** is empty when configuring Business Central the first time. In the **Dependencies** view, click the **Add From Repository** link.

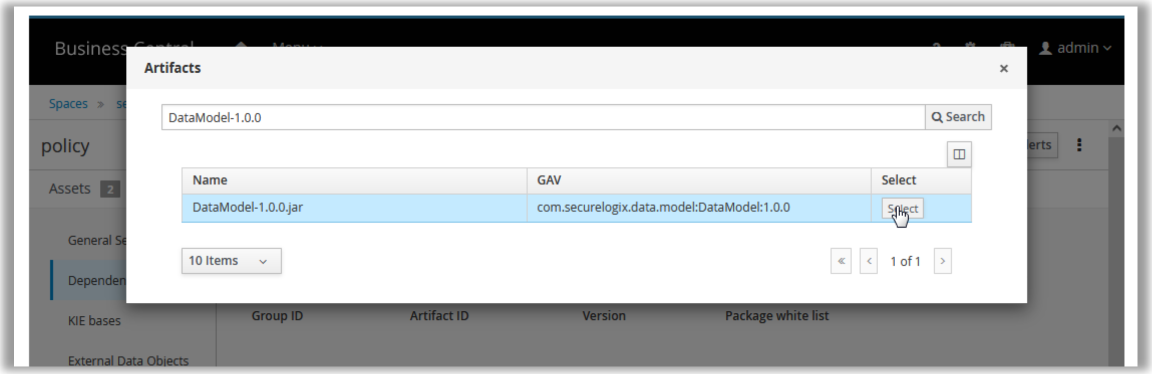


8. A list of Artifacts is displayed.



Name	GAV	Select
junit-4.12.jar	<undetermined>:<undetermined>:<undetermined>	Select
xpp3_min-1.1.4c.jar	<undetermined>:<undetermined>:<undetermined>	Select
commons-codec-1.11.jar	commons-codec:commons-codec:1.11	Select
xmlpull-1.1.3.1.jar	<undetermined>:<undetermined>:<undetermined>	Select
javassist-3.21.0-GA.jar	org.javassist:javassist:3.21.0-GA	Select
slf4j-api-1.7.26.jar	org.slf4j:slf4j-api:1.7.26	Select
optaplanner-persistence-jaxb-7.38.0.Final.jar	org.optaplanner:optaplanner-persistence-jaxb:7.38.0.Fi...	Select
optaplanner-persistence-common-7.38.0.Final.jar	org.optaplanner:optaplanner-persistence-common:7.38...	Select
optaplanner-core-7.38.0.Final.jar	org.optaplanner:optaplanner-core:7.38.0.Final	Select
antlr-runtime-3.5.2.jar	org.antlr:antlr-runtime:3.5.2	Select

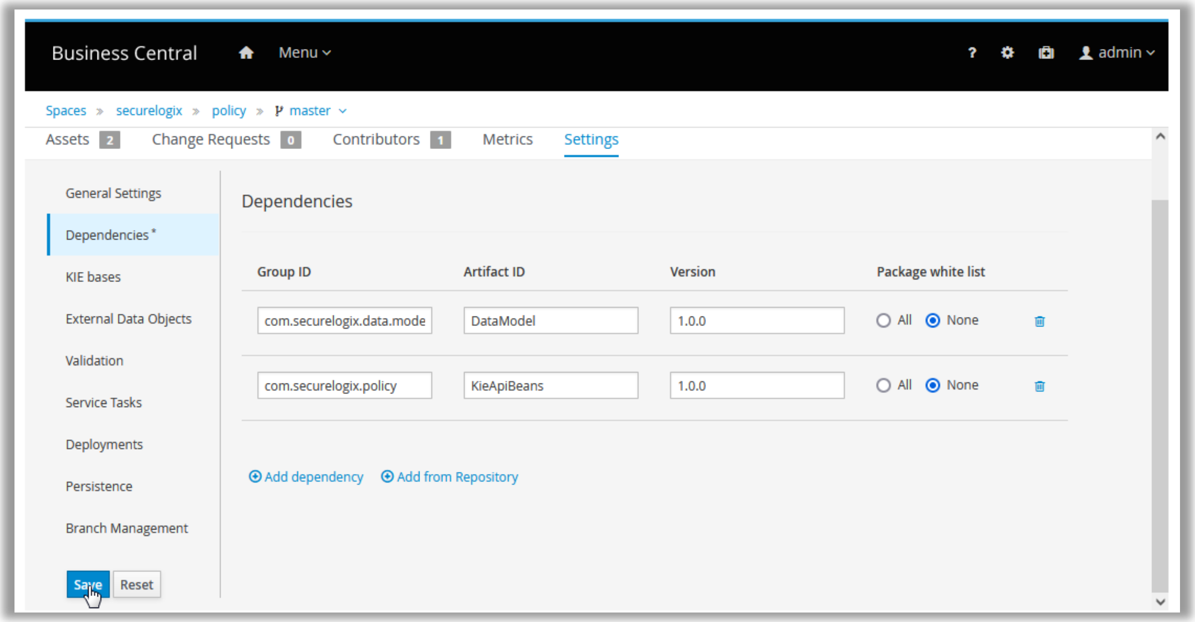
9. In the search bar, search for **DataModel-1.0.0**. This search should find a single artifact with Name **DataModel-1.0.0.jar** and GAV **com.securelogix.data.model:DataModel:1.0.0**. Click the **Select** button for that artifact and it will be added to the **Dependencies** list for the **policy** project.



DataModel-1.0.0

Name	GAV	Select
DataModel-1.0.0.jar	com.securelogix.data.model:DataModel:1.0.0	Select

10. Click **Add From Repository** again, and search for **KieApiBeans-1.0.0**. This search should find a single artifact with Name **KieApiBeans-1.0.0.jar** and GAV **com.securelogix.policy:KieApiBeans:1.0.0**. Click the **Select** button for that artifact and it will be added to the **Dependencies** list for the **policy** project.



Dependencies

Group ID	Artifact ID	Version	Package white list
com.securelogix.data.mode	DataModel	1.0.0	○ All ● None
com.securelogix.policy	KieApiBeans	1.0.0	○ All ● None

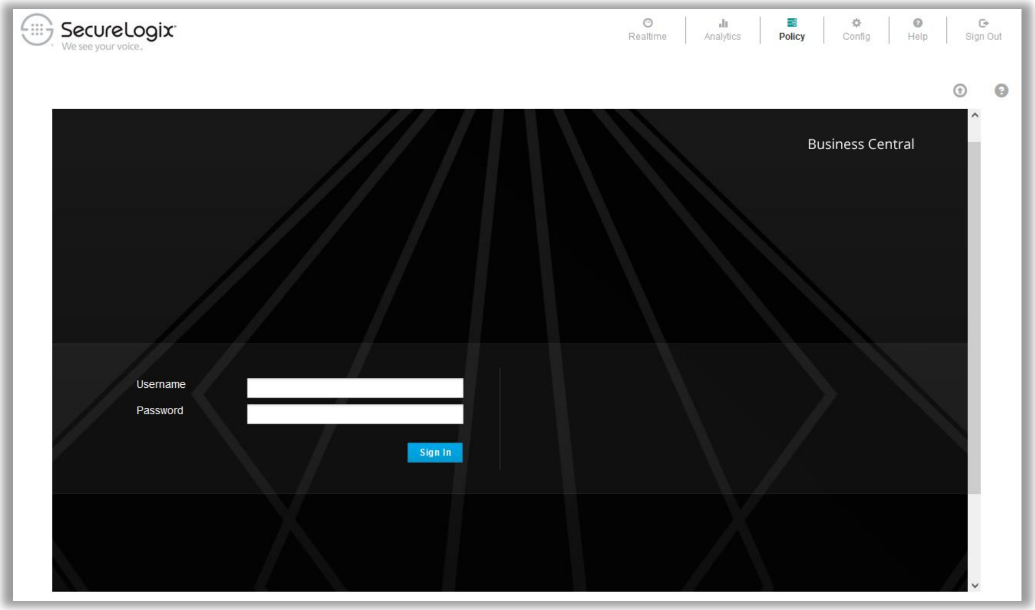
11. **<u>Important/Easy to Miss</u>:** Click the **Save** button at the bottom of the menu tree on the left side (as shown in the screenshot) to actually add those **Dependencies** to the policy project. You are prompted to **Save with Comments**. Optionally type a comment, and then click **Save**.

12. At this point, Business Central and the SEP Policy project are ready to be used for SEP rule creation.

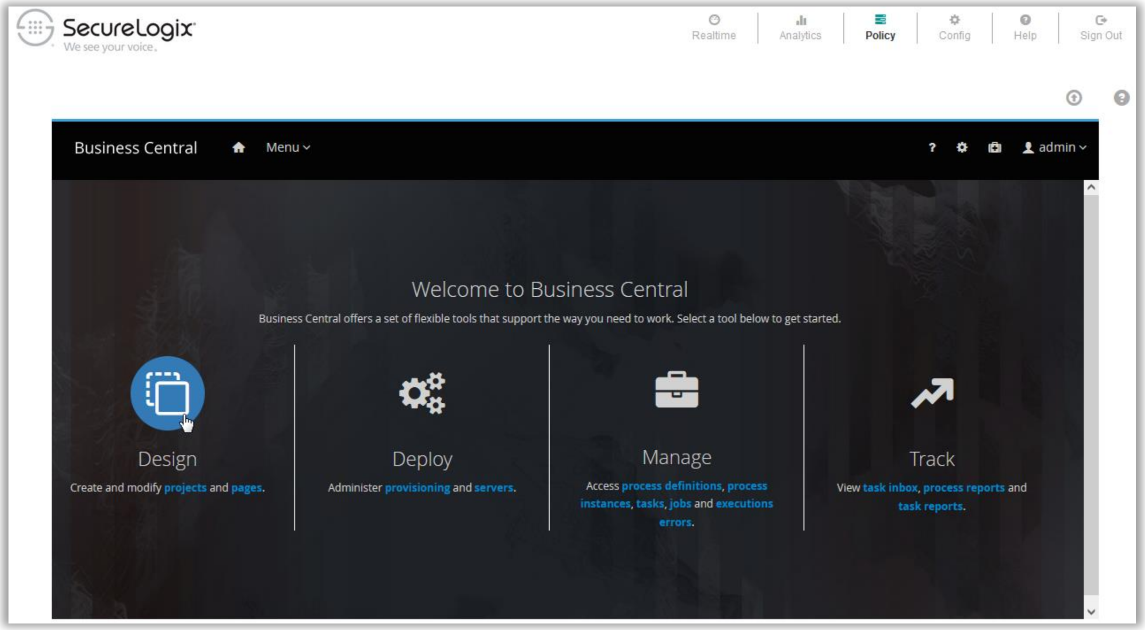
Defining an SEP Call Control Rule

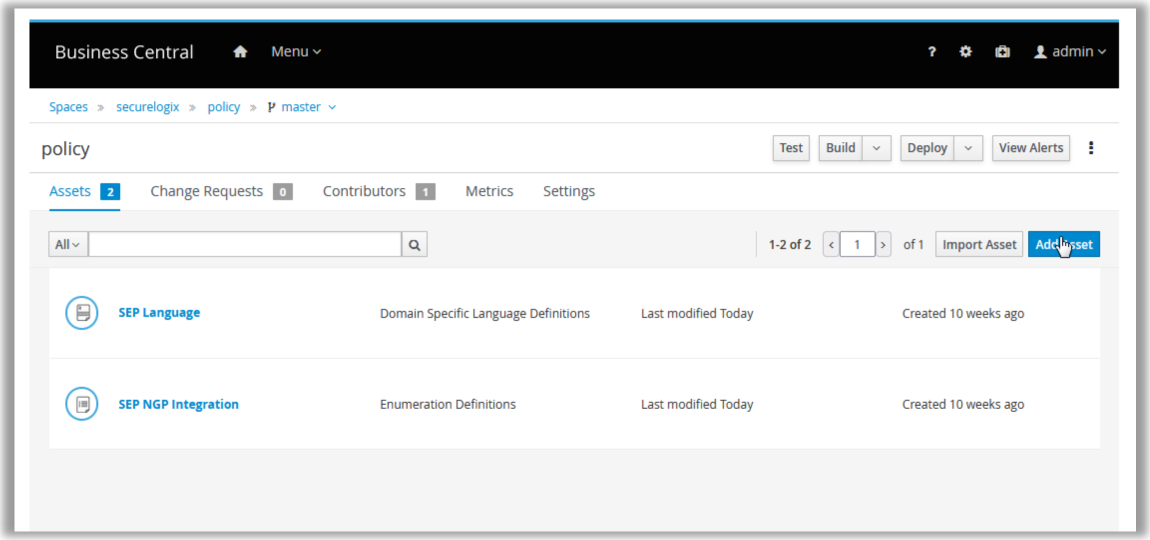
To define an SEP Call Control Rule

1. On the PolicyGuru main menu, click **Policy**.
2. You may be prompted to log in to the Policy GUI. Type your PolicyGuru GUI login credentials and click **Sign in**.



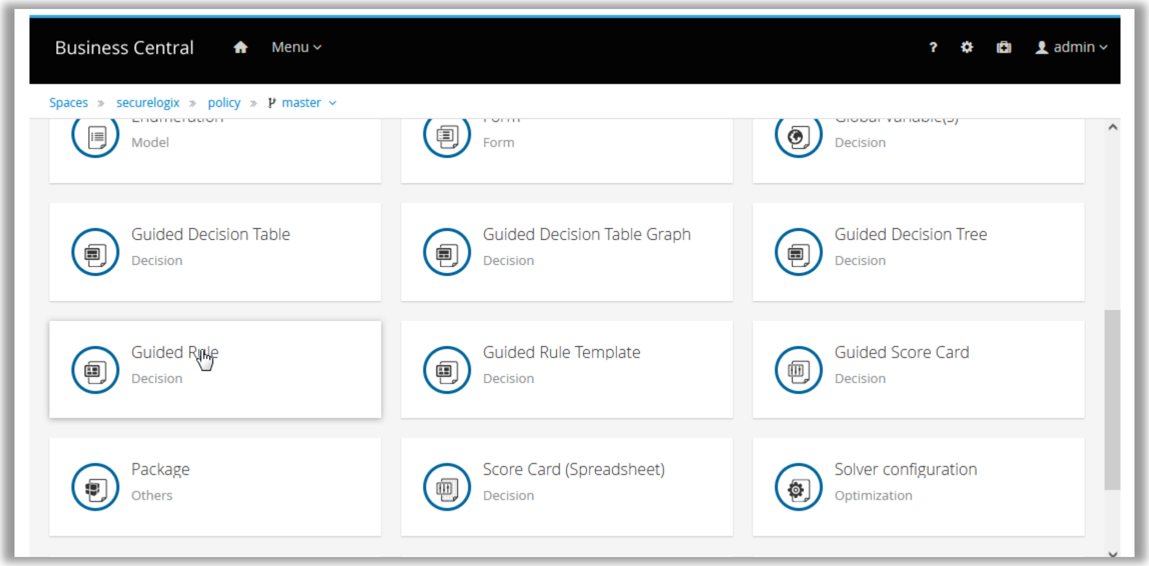
3. In Business Central, click **Design**. navigate to the **securelogix** space, **policy** project, and then click **Assets**. The **SEP Language** and **SLC NGP Integration** files are shown.


SecureLogix
We see your voice.
Realtime
Analytics
Policy
Config
Help
Sign Out
Business Central
Menu
admin
Welcome to Business Central
Business Central offers a set of flexible tools that support the way you need to work. Select a tool below to get started.
Design
Create and modify projects and pages.
Deploy
Administer provisioning and servers.
Manage
Access process definitions, process instances, tasks, jobs and executions errors.
Track
View task inbox, process reports and task reports.

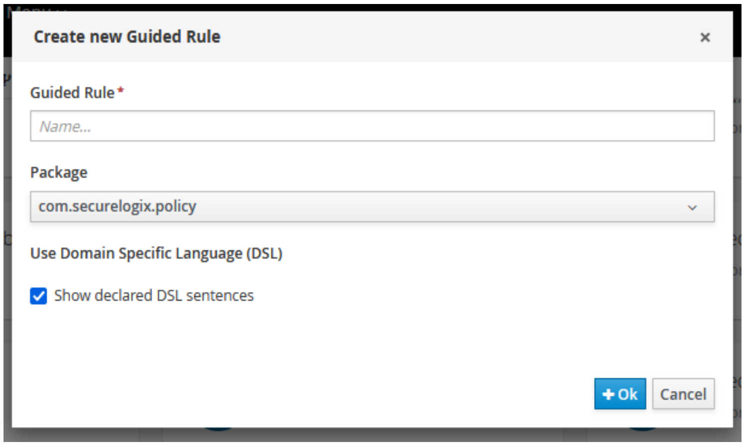


SEP Language	Domain Specific Language Definitions	Last modified Today	Created 10 weeks ago
SEP NGP Integration	Enumeration Definitions	Last modified Today	Created 10 weeks ago

4. Click **Add Asset**.
5. Click **Guided Rule** to add a new rule.



6. The **Create new Guided Rule** dialog box appears.



Create new Guided Rule

Guided Rule*

Name...

Package

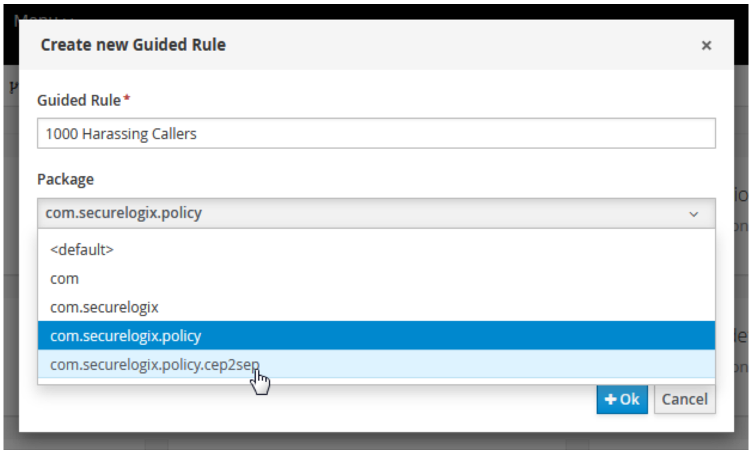
com.securelogix.policy

Use Domain Specific Language (DSL)

☑ Show declared DSL sentences

7. In the **Guided Rule** field, type a name for the Rule. The rules are ordered alphabetically, so choose a name that puts the rule in the expected order in relation to other rules.

8. In the **Package** drop-down list, click **com.securelogix.policy.cep2sep**.



Create new Guided Rule

Guided Rule*

1000 Harassing Callers

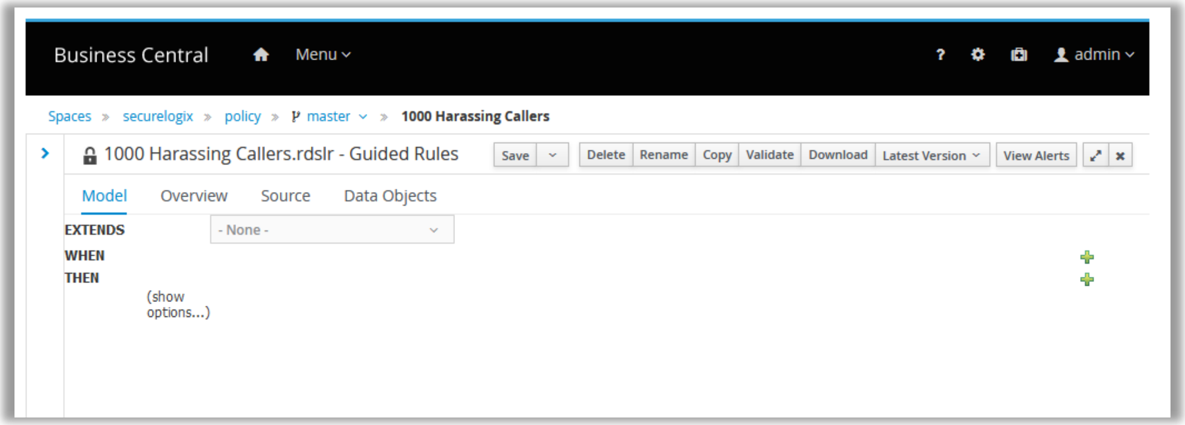
Package

com.securelogix.policy

<default>
com
com.securelogix
com.securelogix.policy
com.securelogix.policy.cep2sep

9. Select **Show declared DSL sentences**.

10. Click **Ok**. The new Rule appears in the Rule editor **Model** view.



Business Central
Menu
?
admin
Spaces » securelogix » policy » master » 1000 Harassing Callers

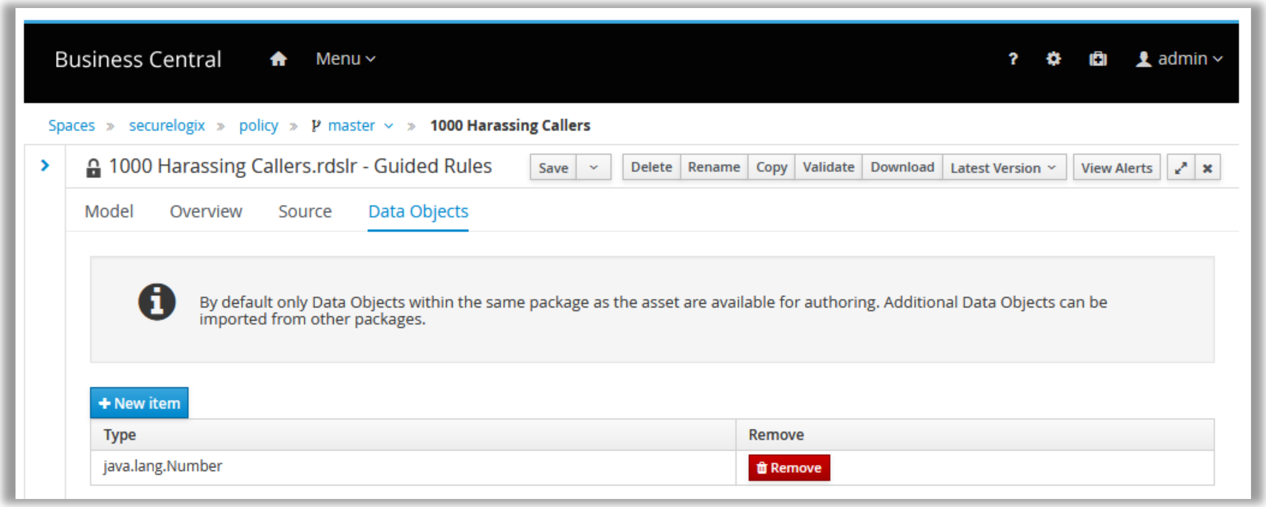
1000 Harassing Callers.rdslr - Guided Rules

EXTENDS - None -

WHEN

THEN

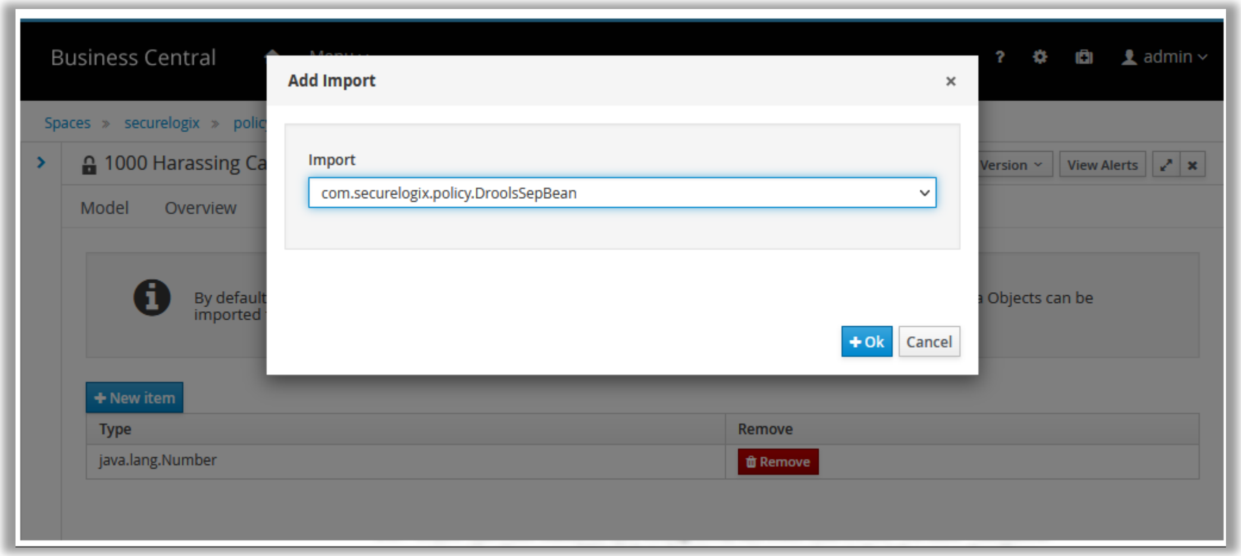
11. Click **Data Objects**.



By default only Data Objects within the same package as the asset are available for authoring. Additional Data Objects can be imported from other packages.

Type	Remove
java.lang.Number	Remove

12. Click **+New item** and then scroll down and click **com.securelogix.policy.DroolsSepBean**. Click **Ok.**

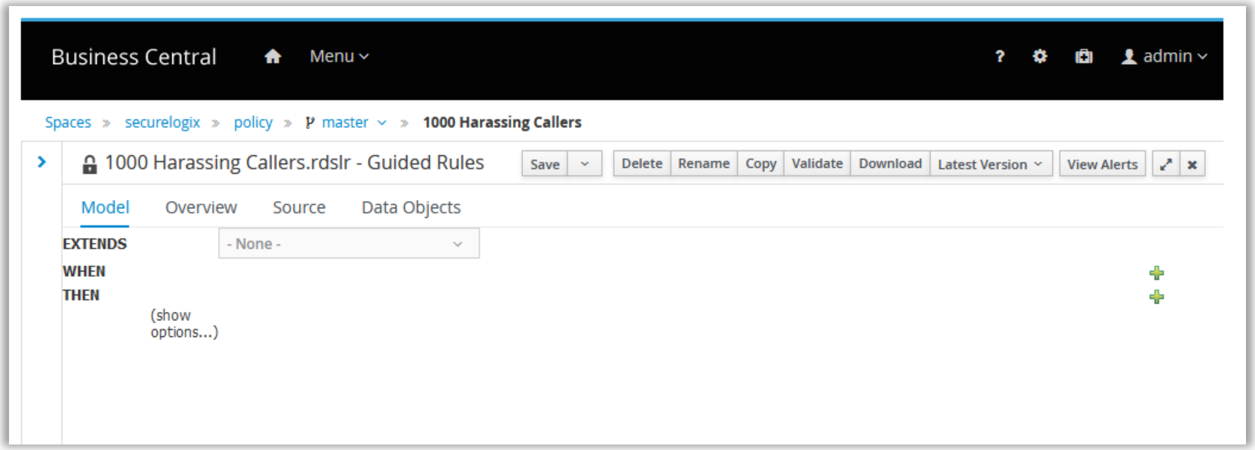


Import: com.securelogix.policy.DroolsSepBean

Type	Remove
java.lang.Number	Remove

13. Click **Model**.

TIP: If the screen is minimized beyond a certain size, the PolicyGuru main menu becomes a menu icon. Click it to access the options.



1000 Harassing Callers.rdslr - Guided Rules

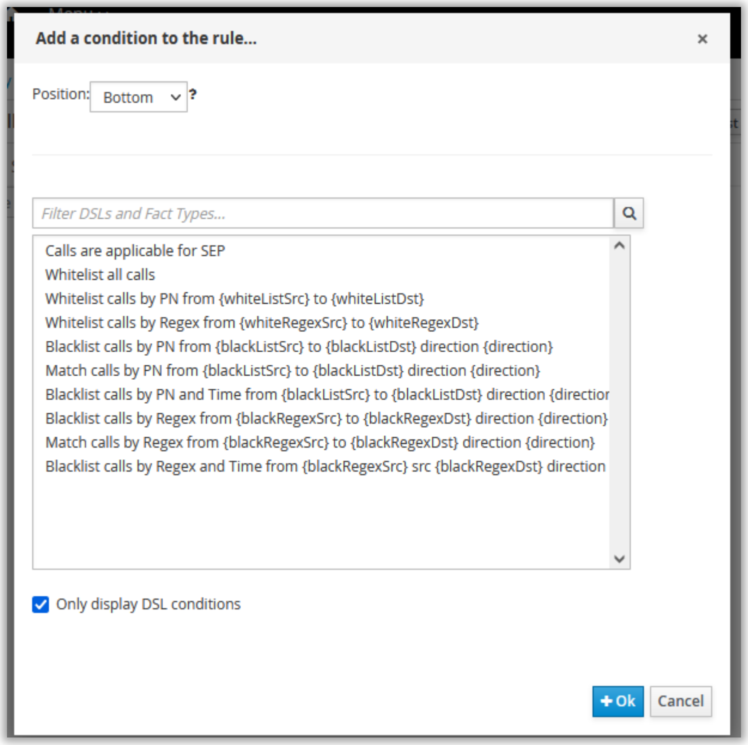
EXTENDS - None -

WHEN

THEN

14. Click the green PLUS SIGN (+) to the right of **When**. The list of **When** conditions defined for SEP Guided Rules appears.

15. Select **Only display DSL conditions**, and then click **Calls are applicable for SEP.** You must always include this option as the first condition for SEP Rules.



Add a condition to the rule...

Position: Bottom

Calls are applicable for SEP
Whitelist all calls
Whitelist calls by PN from {whiteListSrc} to {whiteListDst}
Whitelist calls by Regex from {whiteRegexSrc} to {whiteRegexDst}
Blacklist calls by PN from {blackListSrc} to {blackListDst} direction {direction}
Match calls by PN from {blackListSrc} to {blackListDst} direction {direction}
Blacklist calls by PN and Time from {blackListSrc} to {blackListDst} direction {directior
Blacklist calls by Regex from {blackRegexSrc} to {blackRegexDst} direction {direction}
Match calls by Regex from {blackRegexSrc} to {blackRegexDst} direction {direction}
Blacklist calls by Regex and Time from {blackRegexSrc} src {blackRegexDst} direction

☑ Only display DSL conditions

Note: Time of Day in a SEP Rule **When** clause governs when the Rule is active, not the call time.

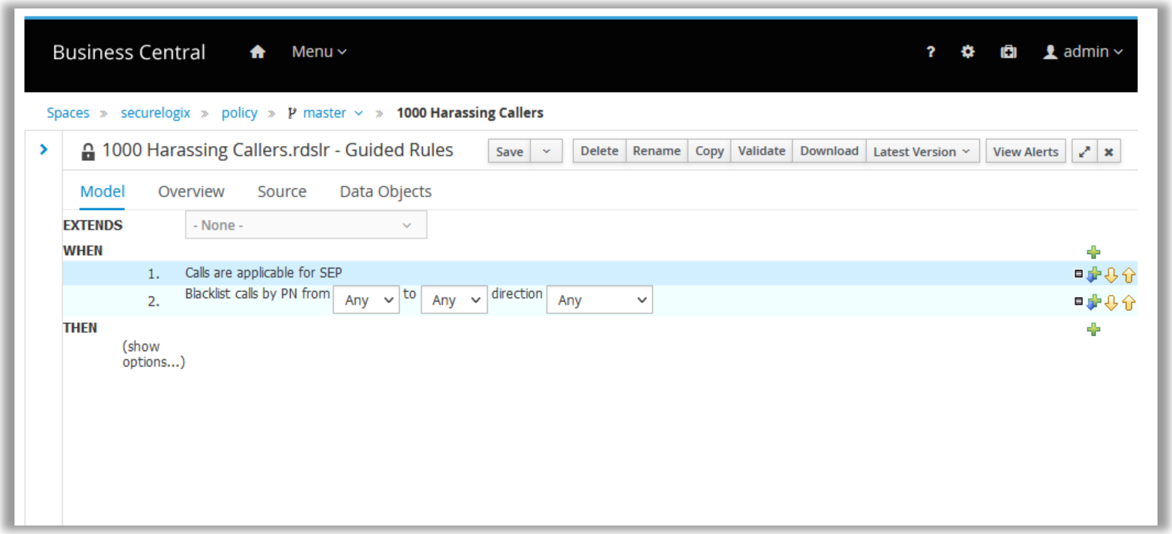
16. Click **OK**. The condition is added as the first **When** condition.

17. Click the green PLUS SIGN (+) to the right of **When** again and add a second **When** condition from the list to define whether this is a Whitelist Rule or a Blacklist Rule and the calls to which the Rule applies.

For example, to terminate harassing inbound calls from Headhunters, you might click **Blacklist Calls by PN from {BlacklistSrc} to {BlacklistDest} direction {direction}**.

IMPORTANT: While you can add additional **When** clauses, only the **When** condition immediately following “Calls are applicable for SEP” is used. If you want to add additional **When** clauses as contingency clauses, you can then change the order of the **When** clauses to change the behavior of the Rule and reinstall the Policy. See “SEP Contingency Rules” on page 11 for more information.

18. Click **OK.**



EXTENDS - None -

WHEN

1. Calls are applicable for SEP
2. Blacklist calls by PN from Any to Any direction Any

THEN

(show options...)

19. Define the **When** fields by clicking the down arrow in the field and selecting the applicable item.

- **Source** and **Destination**—All defined Lists that are of the same type as the Rule and its **When** clause are displayed. For example, a Blacklist Rule using the **When** clause, “Blacklist calls by PN from…” only offers PN Blacklists in the **Source** and **Destination** fields.
- **Direction**—**Inbound**, **Outbound**, or **Any**.

IMPORTANT: In this release, newly added Lists are not automatically available in the Rule editor. To work around this issue each time you manually define or import new Lists::

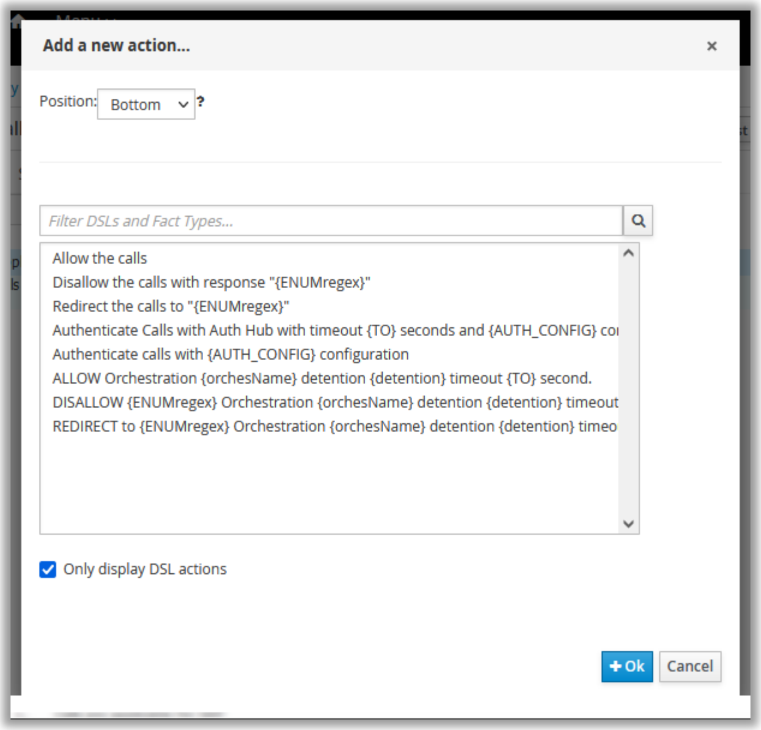
- Open the **SEP NGP Integration** file in the **Asset** list. Do not make any changes to the file, but do click **Save**. Re-saving this file will prompt the system to requery the available Lists and make them available for subsequent Rule file editing. If a Rule file is already open for editing, close it first and then open the Rule for editing again to update the List drop-downs.

Note: You can define a custom DSL option to specify a custom Regex instead of the default for Allow Blacklist Rules. See "Defining a Custom Allow Regex for Blacklist Rules" on page 49.

20. Click the green PLUS SIGN (+) to the right of **Then**, select **Display only DSL actions**, and select the disposition to be applied to calls that trigger this Rule. **Important**: Terminate and Redirect apply only to Blacklist Rules. You can also use Allow with Blacklist Rules. Calls that match a Whitelist Rule are always allowed.

    - To allow matching calls, select **Allow the calls**.
    - To terminate matching calls, select **Disallow the calls with response {ENUMRegex}**.
    - To redirect matching calls, select **Redirect the calls to {ENUMRegex}.**

Note: The disposition containing **Auth Hub** applies only to integration with a specific 3rd party platform. Contact your SecureLogix Sales representative for more information.



Add a new action... ×

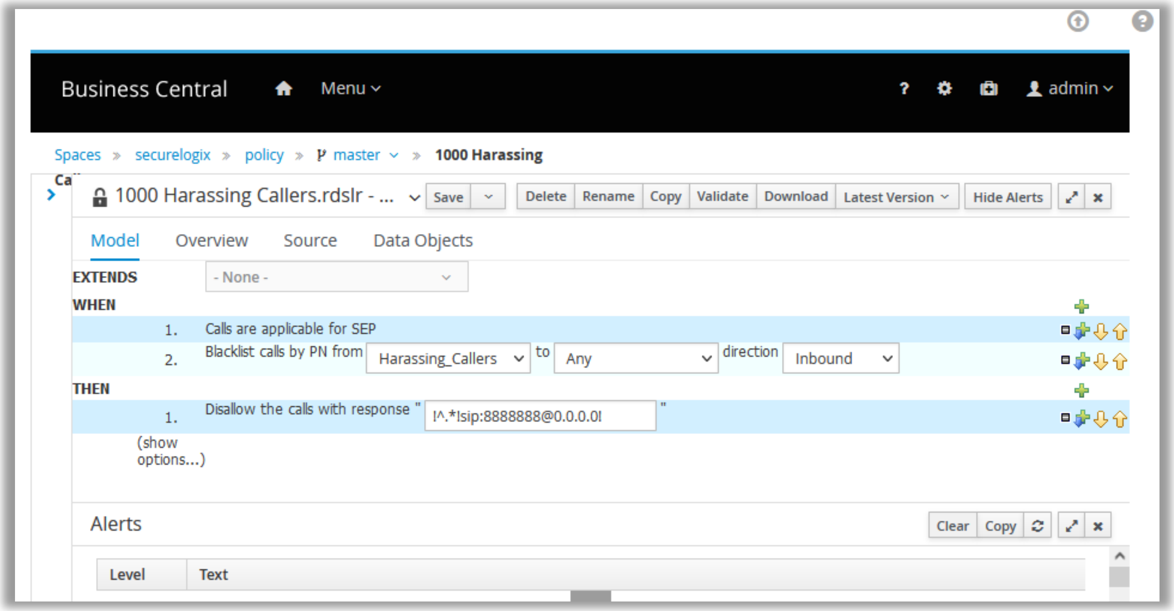
Position: Bottom ?

Filter DSLs and Fact Types...

Allow the calls
Disallow the calls with response "{ENUMregex}"
Redirect the calls to "{ENUMregex}"
Authenticate Calls with Auth Hub with timeout {TO} seconds and {AUTH_CONFIG} coı
Authenticate calls with {AUTH_CONFIG} configuration
ALLOW Orchestration {orchesName} detention {detention} timeout {TO} second.
DISALLOW {ENUMregex} Orchestration {orchesName} detention {detention} timeout
REDIRECT to {ENUMregex} Orchestration {orchesName} detention {detention} timeo

☑ Only display DSL actions

21. Click **OK.**



1000 Harassing Callers.rdslr - ...

EXTENDS - None -

WHEN

1. Calls are applicable for SEP
2. Blacklist calls by PN from Harassing_Callers to Any direction Inbound

THEN

1. Disallow the calls with response "!^.*!sip:8888888@0.0.0.0!"

(show options...)

Alerts

Level	Text

Note: Allow Rules supply the Regex defined for Allow Rules during system configuration, by default: `!a^!guaranteed no replacement!`

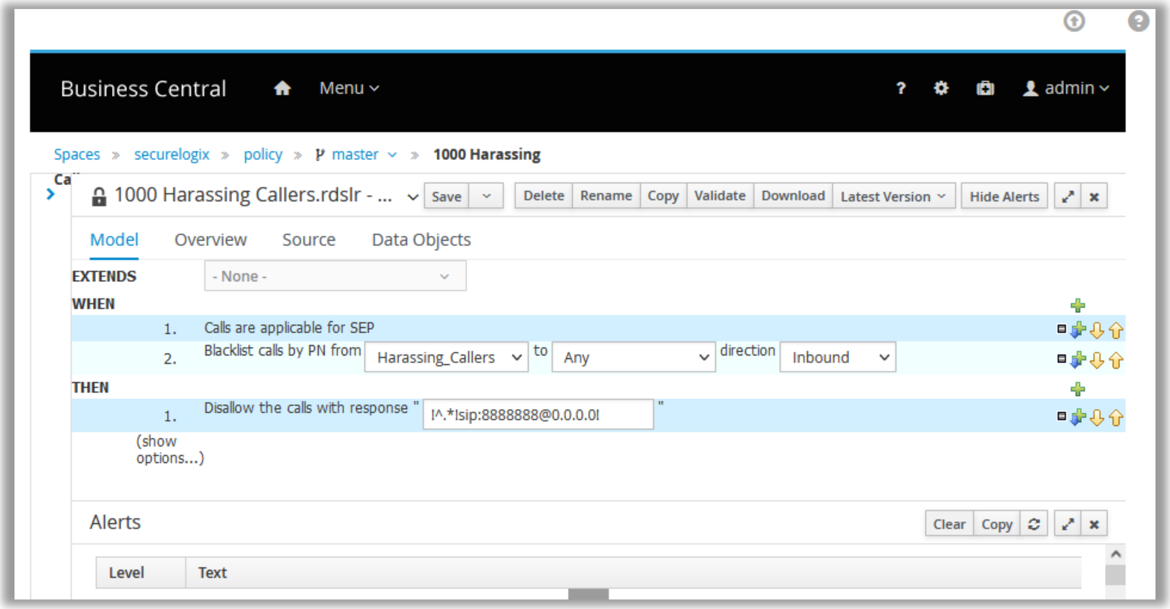
22. If you selected other than the **Allow**… disposition, type the regular expression to be sent. See "Appendix C: Understanding Regular Expressions in Rules" on page 91 for important information about POSIX Regex syntax and "Important Information About Regex in the Guided Rule Editor" on page 47 regarding special character replacement.

23. For example, type: !^.*!sip:8888888@0.0.0.0! for a Termination Rule, which terminates the call by sending it to a nonroutable endpoint, resulting in a **404 Not Found**. The applicable Regex may vary depending on your carrier and SBC configuration.

    IMPORTANT: While you can add additional **Then** clauses, only the first **Then** clause is used in Rule processing. If you want to add additional **Then** clauses as contingency clauses, you can then change the order of the **Then** clauses to change the behavior of the Rule, and then reinstall the Policy. See "SEP Contingency Rules" on page 11 for more information.

24. Leave **show options** set to **mvel** for **Dialect**.

    The illustration below shows a Rule that terminates inbound calls from a Blacklist named **Harassing_Callers** to any destination.



EXTENDS - None -

WHEN

1. Calls are applicable for SEP

2. Blacklist calls by PN from Harassing_Callers to Any direction Inbound

THEN

1. Disallow the calls with response " !^.*!sip:8888888@0.0.0.0! "

(show options...)

25. Click **Validate**. The Rule and the entire Policy is checked for missing imports and other errors. When complete, either a **Validation Successful** message or an error dialog listing errors that must be corrected appears. If errors exist, correct them and then click **Validate** again. You can also refer to the **Alerts** pane at the bottom of the editor for errors.

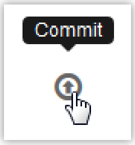
26. Click **Save**. You can optionally click the down arrow and choose to save with a check-in comment.

    - In the **Check in comment** box, type a comment. Check in comments are optional, but they appear in the **Metadata** info of the Rule and can be useful for tracking changes to Rules and the reasons they were made.

27. The Rule is saved but is not currently active on the ENUM Servers. If you are ready to install the Policy with the new Rule, see "Installing or Reinstalling an SEP Policy" below.

Installing or Reinstalling an SEP Policy

To install or reinstall an SEP Policy



1. Wait 15 seconds after clicking **Save** on any Rule and then click the **Commit** icon at the top right above the BRMS main menu.

2. When successful, a **Commit Complete** message appears.

3. View the **Realtime** tab to view the progress of the policy push. A message similar to the following appears:

```
08/27/2015 05:38:47 PM        Info            PolicyPublisherServiceBean - Sending COMMIT version 21
```

4. This message is shortly followed by a message similar to the following for *each* ENUM Server belonging to this Mediation Server:

```
08/27/2015 05:39:06 PM        Info            PolicyVersionCheckServiceBean - Successfully updated app fd49345d-de61-45c9-822b-
                                              05e72684de63_0e:1d:d1:fa:af:70 to policy version 21
```

Information About How an SEP Policy is Reinstalled

When you change an item in a Rule, such as updating a List or adding a Rule to the Policy, and then reinstall a Policy, it pushes down a delta. On the ENUM Server itself, the entire policy is rebuilt to pull in the change. If an error is encountered while pushing or installing the Policy, the ENUM Server sets its Policy version to -1, which then causes the entire Policy to be pushed down from the Mediation Server again.

Important Information About Regex in the Guided Rule Editor

The **Guided Rule Editor** has limitations in parsing certain characters that are frequently used in regular expressions (Regex). The PolicyGuru Solution provides a set of replacement character strings to use instead of these problematic characters.

IMPORTANT: This limitation applies only to Regex in the **Guided Rule Editor**. List entries are not affected.

The table below shows which characters are affected and the replacement character string to use:

Character	Name	Replace with:
(	open parenthesis	&slclb;
)	close parenthesis	&slcrb;
\	back slash	&slcbs;
$	dollar sign	&slcds;

For example:

```
!([0-9]+)\$!sip:\1@10.10.10.10!
```

Must be defined as shown below:

```
!&slclb;[0-9]+&slcrb;&slcbs;&slcds;!sip:&slcbs;1@10.10.10.10!
```

Example Regular Expressions for SEP Rule Responses

IMPORTANT: Rules require POSIX Regex syntax. See “Appendix C: Understanding Regular Expressions in Rules” on page 91 for details.

Below are two sample regular expressions to use in the **Response** field of the **Then** clause in a SEP Redirect or Terminate Rule:

- **Terminate String** —For termination, a host address of 0.0.0.0 results in a **404 Not Found** response .For example:

  ```
  !^.*!sip:8888888@0.0.0.0!
  ```

- **Redirection String**—For redirection, supply the valid host address of the end point that will get the invite. For example:

  ```
  !^.*!sip:2104029669@10.1.1.35!
  ```

  Note: Allow Rules supply the Regex that was defined for Allow Rules during system configuration. By default, the following Regex is used:

  ```
  !a^!guaranteed no replacement!
  ```

See “Defining a Custom Allow Regex for Blacklist Rules” on page 49.

In the above Regex, the clause between the first two exclamation marks is the Regex string that indicates what part of the SIP Request URI is to be matched (^.* means to match everything while a^ means to match nothing). The second clause between the second and third exclamation marks is the replacement string that is used to replace the portion of the SIP Request URI that is matched by the Regex string. In this case, nothing will be matched.

See “Appendix C: Understanding Regular Expressions in Rules” on page 91 for more information about how the ENUM Server uses Regex in Policy enforcement.

Source Link in Rules

When you define the Rule on the **Edit** tab using DSL assets, the editor then generates the required DRL, which is what the decision engine uses to execute the Rule. This generated DRL can be viewed on the **Source** link of the **Policy Editor**, but it cannot be edited there, since it is automatically generated based on the inputs in the editor.

Overview Link in Rules

The **Overview** link of each Rule shows the change history of the Rule since its creation, including who modified it and when. If check-in comments are used, they appear next to each change.

The **Description** field is useful for documenting the purpose and function of the Rule.

A **Discussion** field is also available for comments, such as change requests that have not yet been implemented.

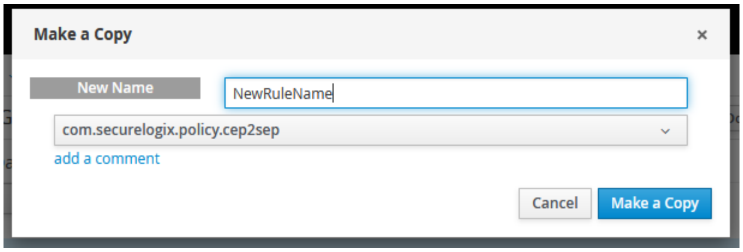
Copying, Deleting, and Renaming SEP Rules

Refer to the following procedures for instructions for copying, deleting, and renaming SEP Rules.

Copying a Rule

To copy an SEP Rule as the basis for a new Rule

1. Open the Rule you want to copy.
2. On the Rule Editor main menu, click **Copy**. The **Make a copy** dialog box appears.



New Name: NewRuleName

com.securelogix.policy.cep2sep

3. In the **New Name** box, type the name for the new Rule.
4. In the **Packages** drop down box, select **com.securelogix.policy.cep2sep**.

   IMPORTANT: If you fail to do this, the Rule will not be editable nor included in SEP Policy.
5. Click **Make a Copy**. The Rule is copied as a new Rule.

6. The Rule you copied remains open in the Editor. Close that Rule by clicking the **x** on the main menu, and then in the **Project Explorer** pane, click the new Rule to open it for editing.

7. Make changes to the new Rule as needed, and then click **Save**.

   Note: When you make a copy of a Rule, an erroneous error message appears in the **Alerts** section of the Rule editor: `[KBase: defaultKieBase]: Rule Compilation error The type Rule_<rule_name> is already defined.` Ignore this error.

8. When you have completed editing all Rules for this session, wait 15 seconds and then click **Commit** to install the edited Rules on the ENUM Servers in your deployment.

Deleting a Rule

To delete a Rule

1. Open the Rule in the Editor and then click **Delete** on the main menu.

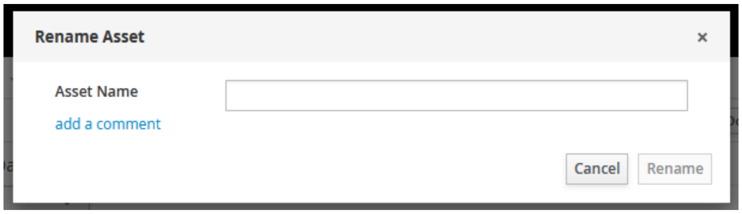
2. When you have completed editing all Rules for this session, wait 15 seconds and then click **Commit** to install the edited Rules on the ENUM Servers in your deployment.

Renaming a Rule

To rename a Rule

1. Open the Rule in the Editor and then click **Rename** on the main menu.

2. The **Rename Asset** dialog box appears.



3. Type the new Rule name and then click **OK**. You can optionally add a comment denoting the reason for the change.

4. When you have completed editing all Rules for this session, wait 15 seconds and then click **Commit** to push the edited Rules to the ENUM Servers in your deployment.

Defining a Custom Allow Regex for Blacklist Rules

By default, PolicyGuru SEP Blacklist Allow Rules use the following Regex: **!a^!guaranteed no replacement!**. You can define a custom Regex to use in Blacklist Allow Rules instead of the default.

To change the Allow Regex for Blacklist Rules

1. Close any open SEP Rules.

2. On the Business Central main menu, click **Policy**.

3. Click **SEP Language**. It opens in the editing pane.

4. Scroll down to the **[then]** entries.

5. Below **[then]Allow the calls=$sep.doRuleAction("ALLOW", "!a^!guaranteed no replacement!")**, add the following line, substituting your preferred Regex for the example Regex shown in red here:

```
[then]AllowREGEX=$sep.doRuleAction( "ALLOW",
"!&slclb;^.*&slcrb;!sip:authorized&slcbs;1@securelogix!" );
```

6. On the **Policy Editor** main menu, click **Save**.
7. On the **Policy Editor** main menu, click the **X** to close the file.
8. **AllowREGEX** now appears in the list of available **Then** conditions for Blacklist Rules. To define SEP Blacklist Rules using the Regex you specified, select **AllowREGEX** instead of **Allow the calls** as the **Then** statement.

Note: The new **AllowREGEX** option is not available in any Rules that were open when you added it until you close and reopen the Rule.

Defining SEP Policy Rules for Orchestra One™ Verification Requests

The PolicyGuru Solution provides an integration that enables the PolicyGuru Mediation Server to send verification requests to the Orchestra One™ Call Verification Service. This feature is called the o1 Agent. Orchestra One™ Call Verification Request Rules specify which calls are to trigger an Orchestra One Request. See the *PolicyGuru® Meta-Policy Controller System Administration Guide* for information on configuring the o1 Agent before using these instructions.

The o1 Agent can be configured to send verification requests for all inbound calls to Orchestra One, or SEP Policy Rules can be defined that specify only certain destination and/or source numbers for which call verification requests are to be sent, such as contact center lines.

Use the procedures below to create and install SEP Rules to trigger selective Orchestra One Requests if your system is configured to use this functionality. See the *PolicyGuru® Meta-Policy Controller System Administration Guide* for more information about the o1 Agent and available configuration modes.

Rule Order Considerations

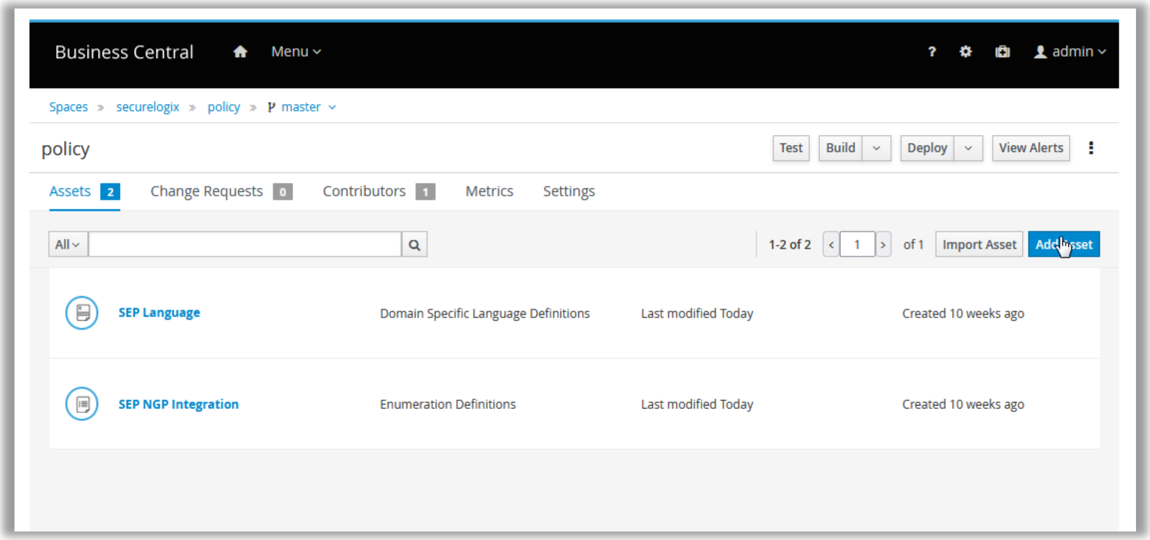
Recall that SEP Rules are processed as follows: All Whitelist Rules in the order in which they appear and then all Blacklist Rules in the order in which they appear. Also recall that the Rules are listed in the Policy in ASCII alphabetical order. As previously discussed, use a numbering scheme to ensure the Rules are listed in the correct order for processing.

Orchestra One Verification Request Rules are Blacklist Allow Rules. They should be placed after all Blacklist Terminate (reject) and Blacklist Redirection Rules.

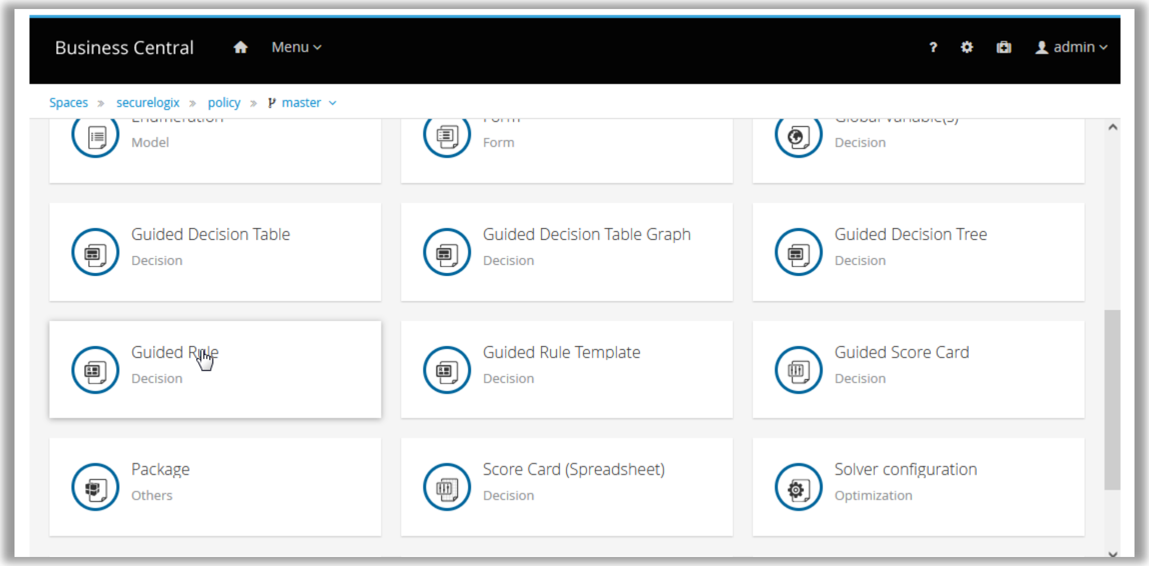
Defining an Orchestra One™ Verification Request Rule

To define an Orchestra One™ Verification Request Rule

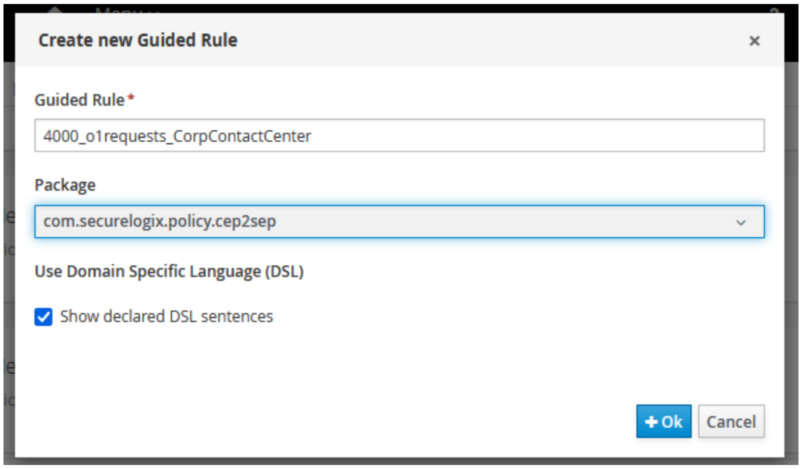
1. In Business Central, click **Design**. navigate to the **securelogix** space, **policy** project, and then click **Assets**. The **SEP Language** and **SLC NGP Integration** files are shown.



2. Click **Add Asset**.
3. Click **Guided Rule** to add a new rule.



4. The **Create new Guided Rule** dialog box appears.


Create new Guided Rule

Guided Rule*: 4000_o1requests_CorpContactCenter

Package: com.securelogix.policy.cep2sep

Use Domain Specific Language (DSL)

☑ Show declared DSL sentences

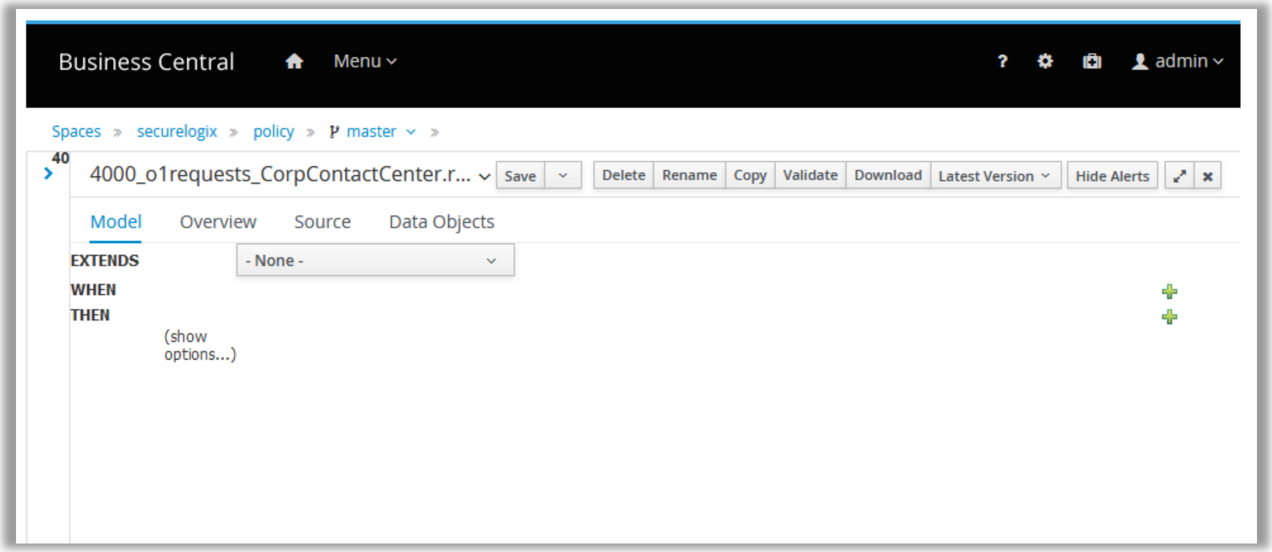
5. Type a name for the Rule that identifies its purpose. For example, type:

   `4000_o1requests_CorpContactCenter`

6. In the **Package** dropdown, select **com.securelogix.policy.cep2sep**.

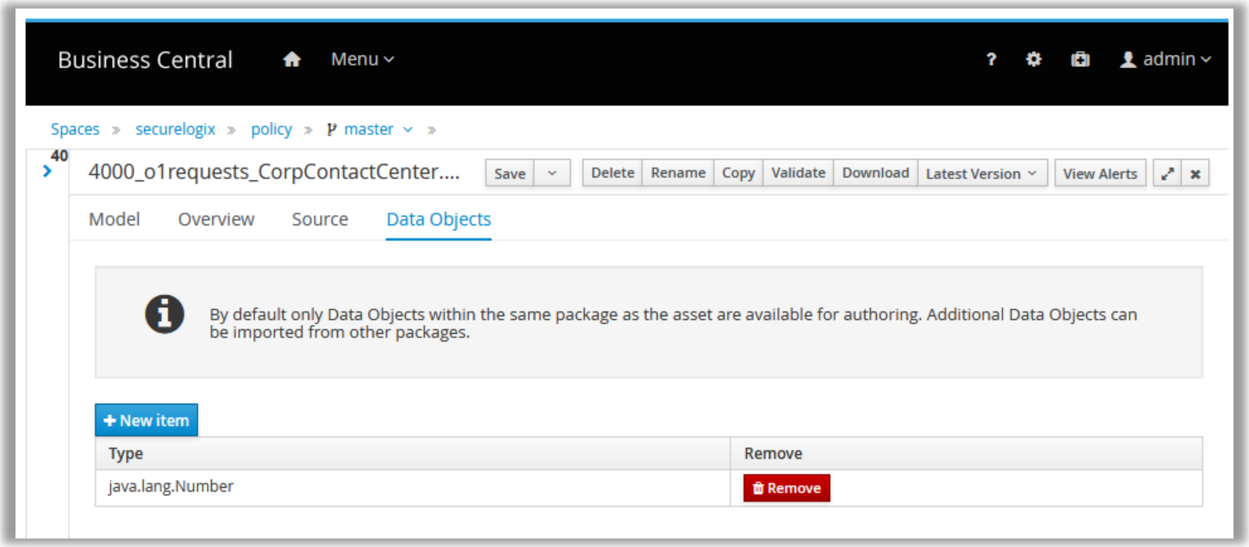
7. Select **Show declared DSL sentences**.

8. Click **OK**. The new Rule appears in the **Project Explorer** tree pane and opens in the **Model** view of the Rule Editor.



EXTENDS: - None -

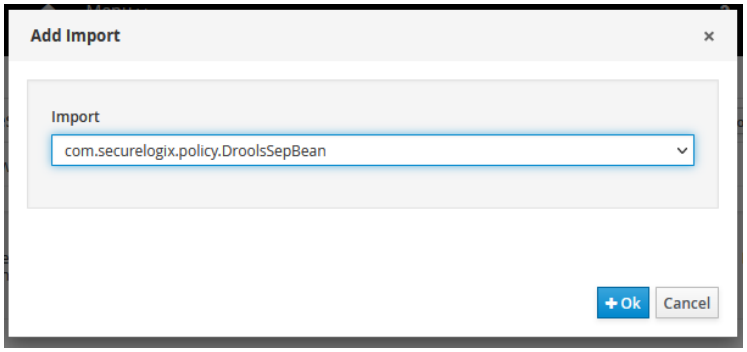
9. Click **Data Objects**.



By default only Data Objects within the same package as the asset are available for authoring. Additional Data Objects can be imported from other packages.

Type	Remove
java.lang.Number	Remove

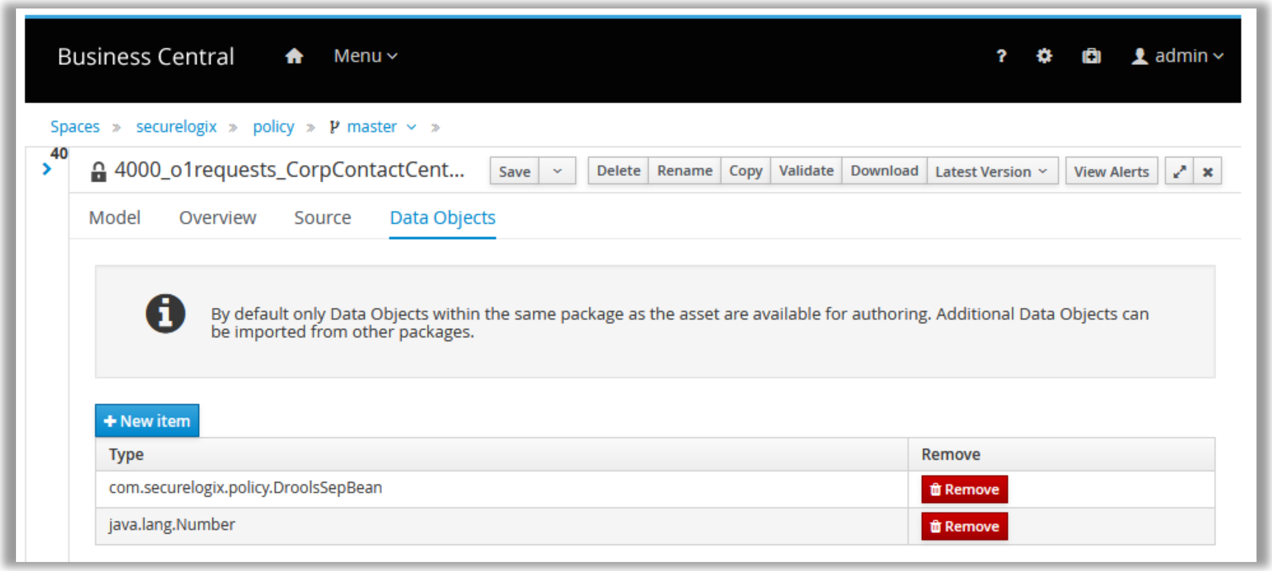
10. Click **New item**. The **Add import** dialog box appears.



Import: com.securelogix.policy.DroolsSepBean

11. In the **Import** box, click the down arrow, scroll down the list, and click **com.securelogix.policy.DroolsSepBean**. (Entries are in alphabetical order. This one is near the bottom of the list.)

12. Click **OK**.

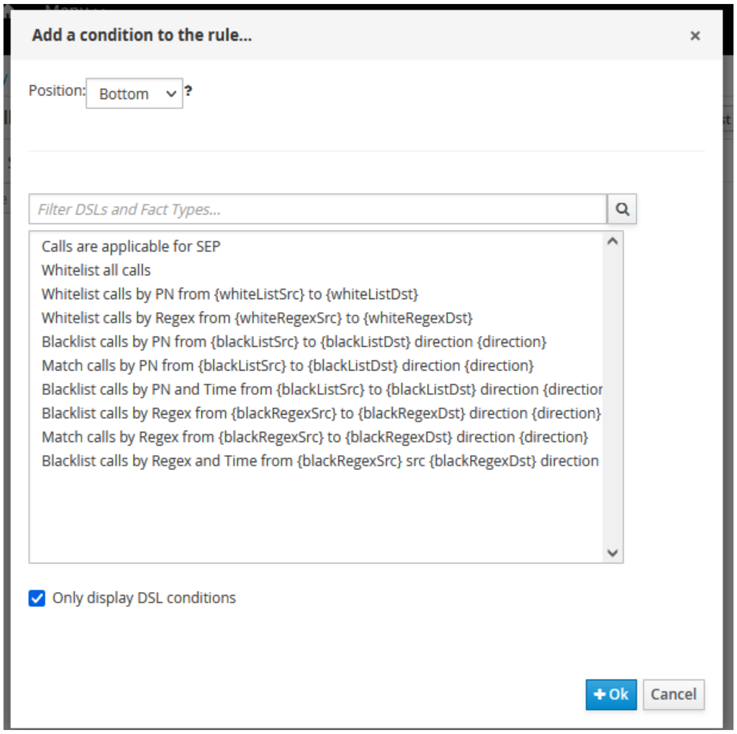


By default only Data Objects within the same package as the asset are available for authoring. Additional Data Objects can be imported from other packages.

Type	Remove
com.securelogix.policy.DroolsSepBean	Remove
java.lang.Number	Remove

13. Click **Model**.

14. Click the green PLUS SIGN to the right of **WHEN**. The **SEP WHEN Conditions** dialog box appears.



Calls are applicable for SEP
Whitelist all calls
Whitelist calls by PN from {whiteListSrc} to {whiteListDst}
Whitelist calls by Regex from {whiteRegexSrc} to {whiteRegexDst}
Blacklist calls by PN from {blackListSrc} to {blackListDst} direction {direction}
Match calls by PN from {blackListSrc} to {blackListDst} direction {direction}
Blacklist calls by PN and Time from {blackListSrc} to {blackListDst} direction {directior
Blacklist calls by Regex from {blackRegexSrc} to {blackRegexDst} direction {direction}
Match calls by Regex from {blackRegexSrc} to {blackRegexDst} direction {direction}
Blacklist calls by Regex and Time from {blackRegexSrc} src {blackRegexDst} direction

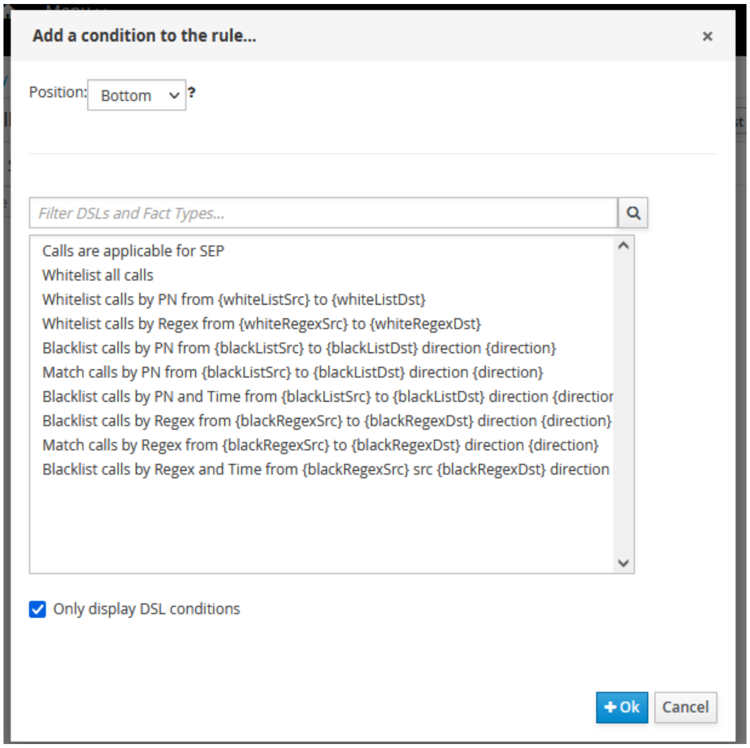
☑ Only display DSL conditions

Note: The **Match calls…**options are also Blacklist options, since they use Blacklists to specify source and destination.

1. Click **Calls are applicable to SEP** and then click **OK**.

2. Click the green PLUS SIGN to the right of **WHEN** again. The **SEP WHEN Conditions** dialog box appears.

3. Click one of the Blacklist options, according to whether you will use Regex Blacklists or a Phone Number Blacklists to specify the called and/or calling numbers to which the Rule applies.



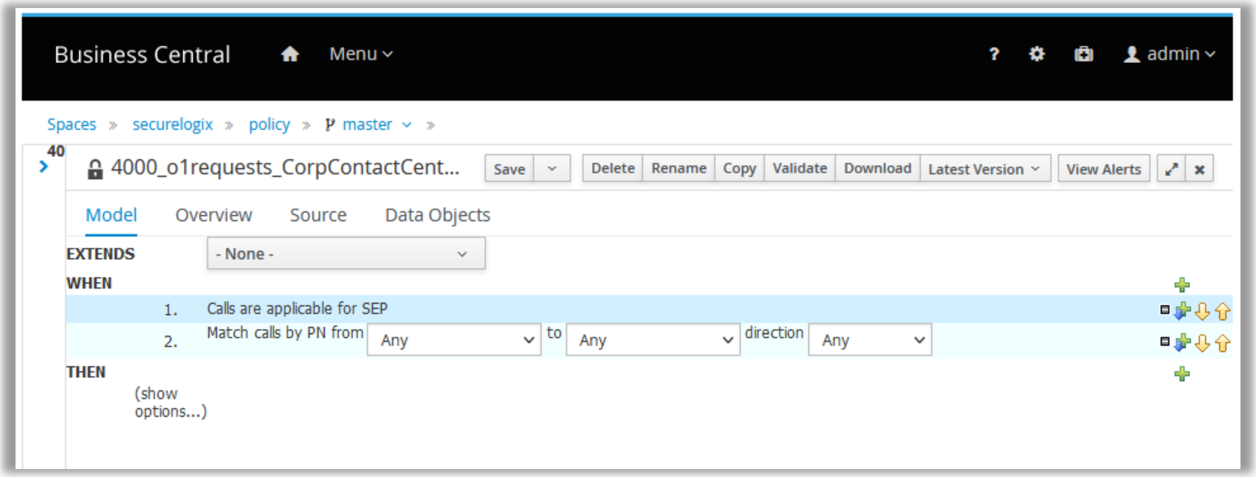
Add a condition to the rule...

Position: Bottom

Calls are applicable for SEP
Whitelist all calls
Whitelist calls by PN from {whiteListSrc} to {whiteListDst}
Whitelist calls by Regex from {whiteRegexSrc} to {whiteRegexDst}
Blacklist calls by PN from {blackListSrc} to {blackListDst} direction {direction}
Match calls by PN from {blackListSrc} to {blackListDst} direction {direction}
Blacklist calls by PN and Time from {blackListSrc} to {blackListDst} direction {directior
Blacklist calls by Regex from {blackRegexSrc} to {blackRegexDst} direction {direction}
Match calls by Regex from {blackRegexSrc} to {blackRegexDst} direction {direction}
Blacklist calls by Regex and Time from {blackRegexSrc} src {blackRegexDst} direction

☑ Only display DSL conditions

4. Click **OK**.



4000_o1requests_CorpContactCent...

EXTENDS - None -

WHEN

1. Calls are applicable for SEP

2. Match calls by PN from Any to Any direction Any

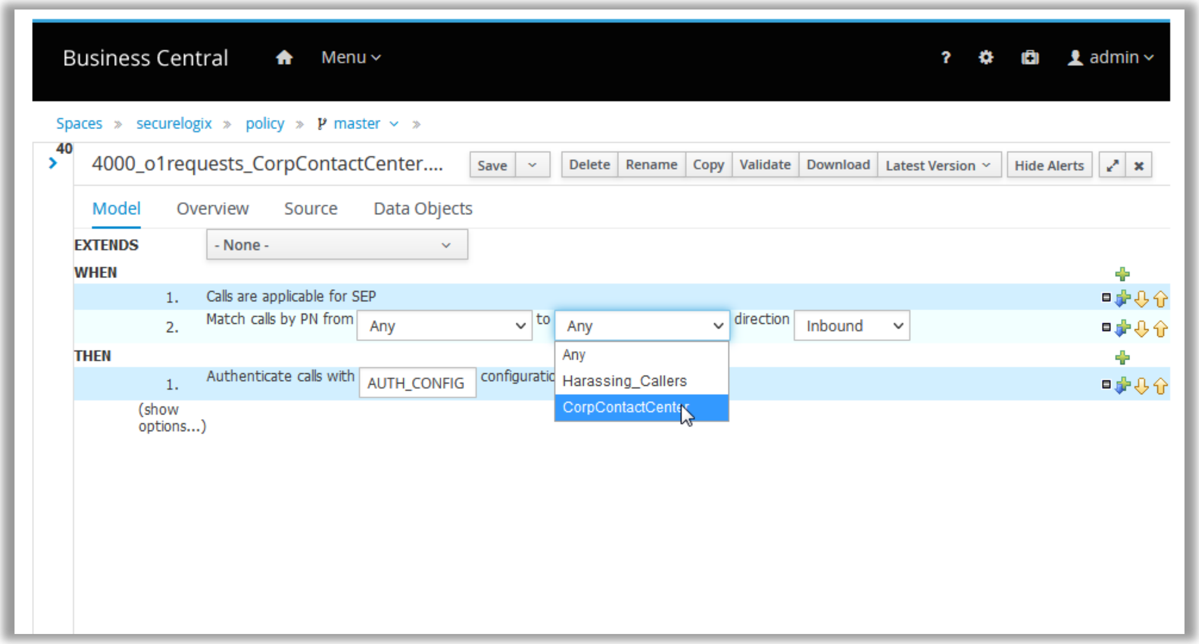
THEN

(show options...)

5. In the **direction** box, click the down arrow and click **Inbound**.

6. Next, you’ll specify the source and/or destination numbers to which this Rule applies. For example, suppose you want to specify that all inbound calls to your Corporate Contact Center that don’t match a previous Whitelist or Blacklist Terminate/Redirect Rule are to trigger an Orchestra One Verification Request. Leave the **from** box (source) set to **Any**. Click the down arrow in the **to** (destination) box.

7. Click the List that contains the destinations in your organization to which this Rule applies.



EXTENDS - None -

WHEN

1. Calls are applicable for SEP

2. Match calls by PN from Any to Any direction Inbound

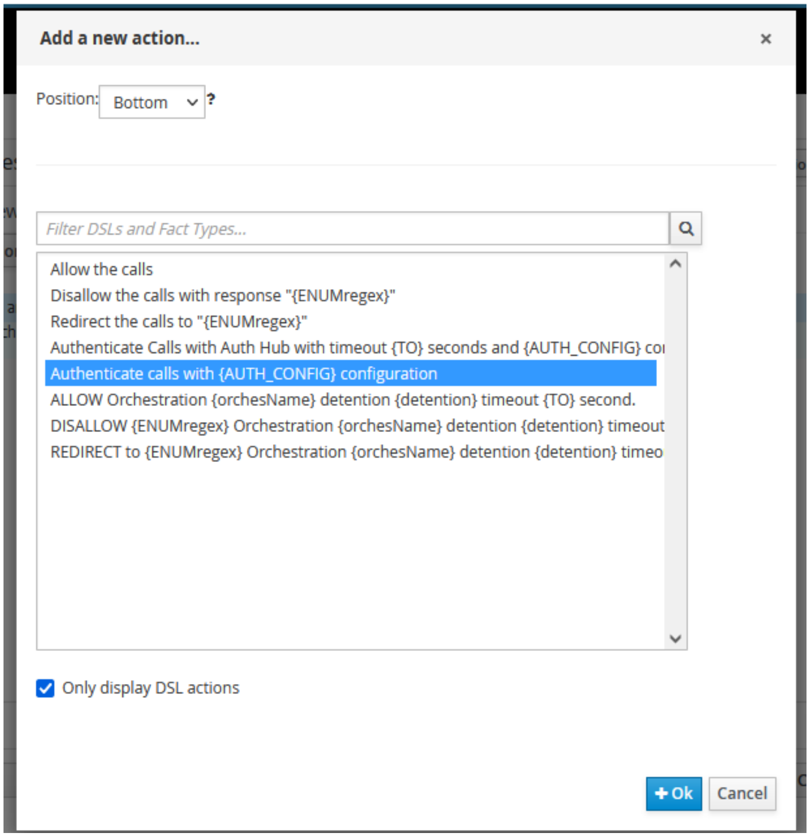
Any
Harassing_Callers
CorpContactCenter

THEN

1. Authenticate calls with AUTH_CONFIG configuratio

(show options...)

8. Next, you’ll define the **THEN** action that denotes what is to occur when the **WHEN** conditions match a call, in this case an Orchestra One Verification Request. Click the green PLUS SIGN to the right of **THEN**. The **Add new action** dialog box appears.

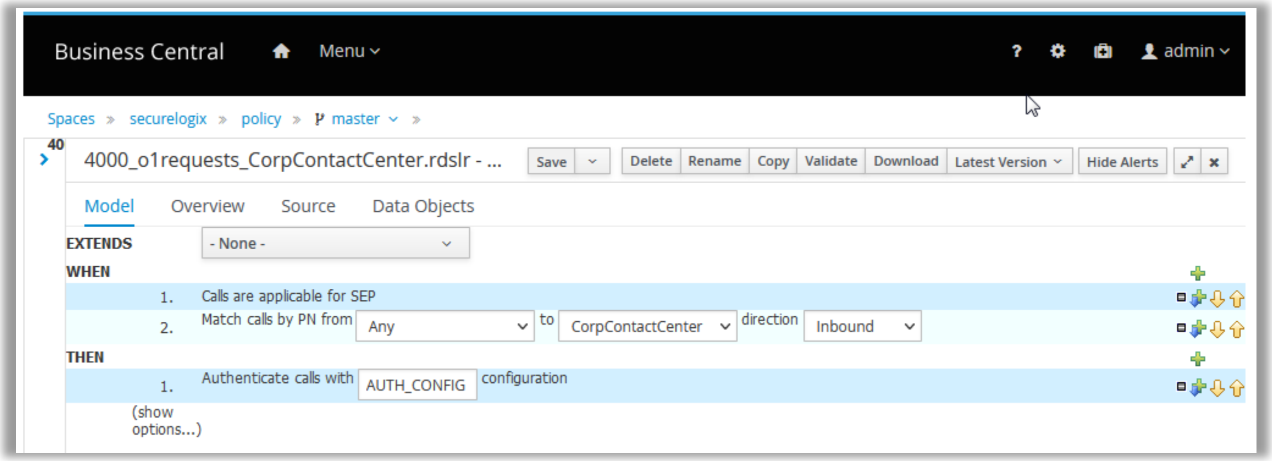


Position: Bottom ?

Allow the calls
Disallow the calls with response "{ENUMregex}"
Redirect the calls to "{ENUMregex}"
Authenticate Calls with Auth Hub with timeout {TO} seconds and {AUTH_CONFIG} co
Authenticate calls with {AUTH_CONFIG} configuration
ALLOW Orchestration {orchesName} detention {detention} timeout {TO} second.
DISALLOW {ENUMregex} Orchestration {orchesName} detention {detention} timeout
REDIRECT to {ENUMregex} Orchestration {orchesName} detention {detention} timeo

☑ Only display DSL actions

9. Click **Authenticate calls with (AUTH_CONFIG) configuration** and then click **OK**.



4000_o1requests_CorpContactCenter.rdslr - ...

EXTENDS - None -

WHEN

1. Calls are applicable for SEP

2. Match calls by PN from Any to CorpContactCenter direction Inbound

THEN

1. Authenticate calls with AUTH_CONFIG configuration

(show options...)

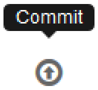
10. Replace **AUTH_CONFIG** with the name of a unique identifier you have added to a Data Processor in the **correlationXcvr.json** file, or type **default** to match requests with config names that do not match any other Data Processor. See “Defining Unique Identifiers for Rule-Based o1 Processing” on page 57 for more information.

11. Leave **show options** set to **mvel** for **Dialect**.

12. The Rule is complete. Click **Save** on the **Rule Editor** main menu. **Note**: Orchestra One Verification Request Rules uses the default **Allow** Regex for matching calls. You can view this on the **Source** tab.

13. Click **Validate**.

14. Click the **Commit** button on the right below the PolicyGuru GUI main menu to install the SEP Policy on the ENUM Servers in your deployment for the Rule to take effect.



Defining Unique Identifiers for Rule-Based o1 Processing

To use Rule-based processing for Orchestra One requests, you must edit the **correlationXcvr.json** file on the Mediation Server to add transceivers with unique config names for each Rule you create, and then you replace **AUTH_CONFIG** in each Rule with the applicable config name. See the *PolicyGuru® System Administration Guide* for instructions for editing this file.

Real-Time Analytics

Viewing Call and Policy Processing Data with On-Screen Analytics

Real-Time Analytics provide graphical, drill-down onscreen reporting views of call and Policy processing data. Drill-down ENUM Call Details (using ENUM Server data) and SIP Call Details views (using Metadata Probe data), and "drill-up" Phone Number Analytics views (using Metadata Probe data for forensic analysis of a specific phone number or set of phone numbers) are available.

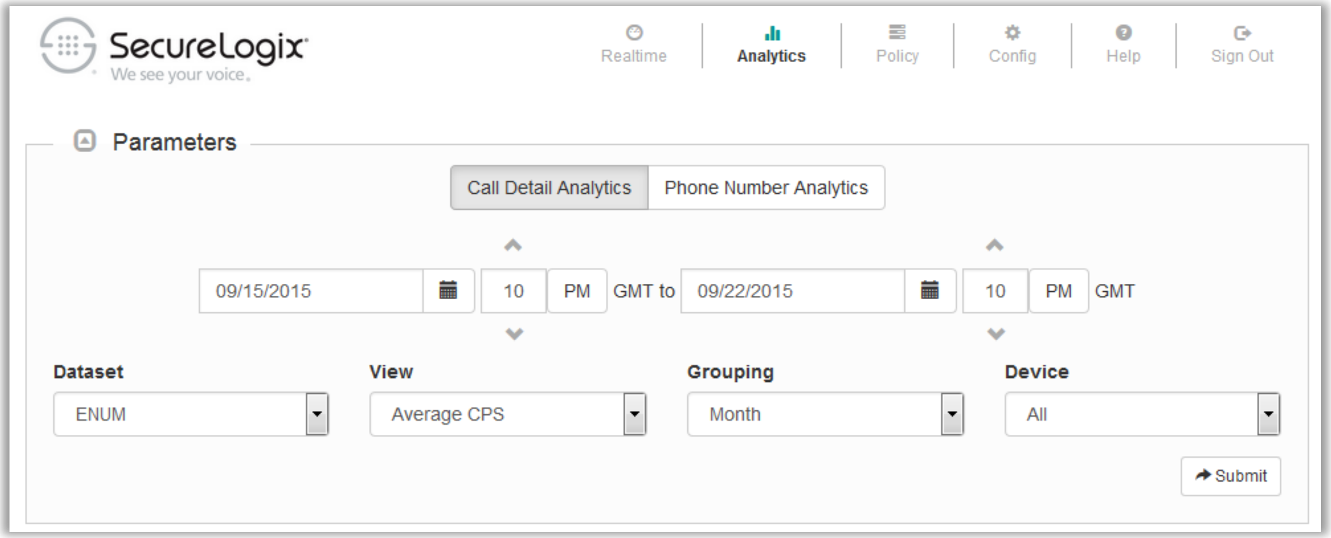
ENUM Call Detail Analytics

ENUM Call Details provide drill-down analytics views using ENUM Server call and SEP Policy processing data.

To view ENUM Call Detail Analytics

1. On the PolicyGuru main menu, click **Analytics**.

   The **Analytics** screen appears set to **Call Detail Analytics**, with **ENUM** selected as the dataset by default.



09/15/2015 10 PM GMT to 09/22/2015 10 PM GMT

Dataset: ENUM

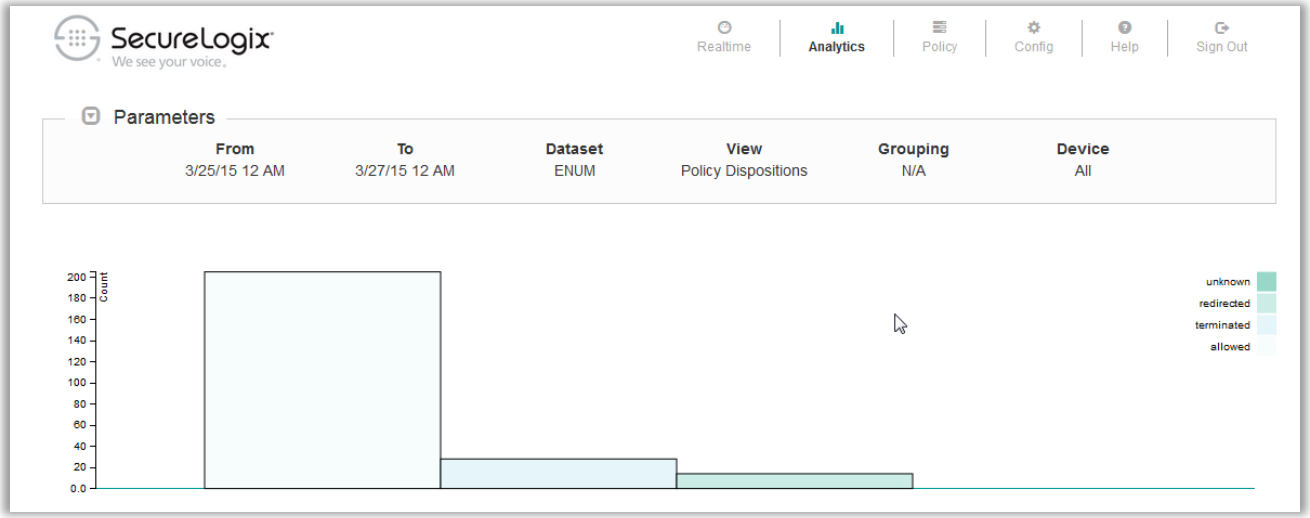
View: Average CPS

Grouping: Month

Device: All

2. In the **Calendar** fields, select the start and end days and times for which to retrieve data. Times are in GMT.
3. Leave the **Dataset** field set to ENUM.

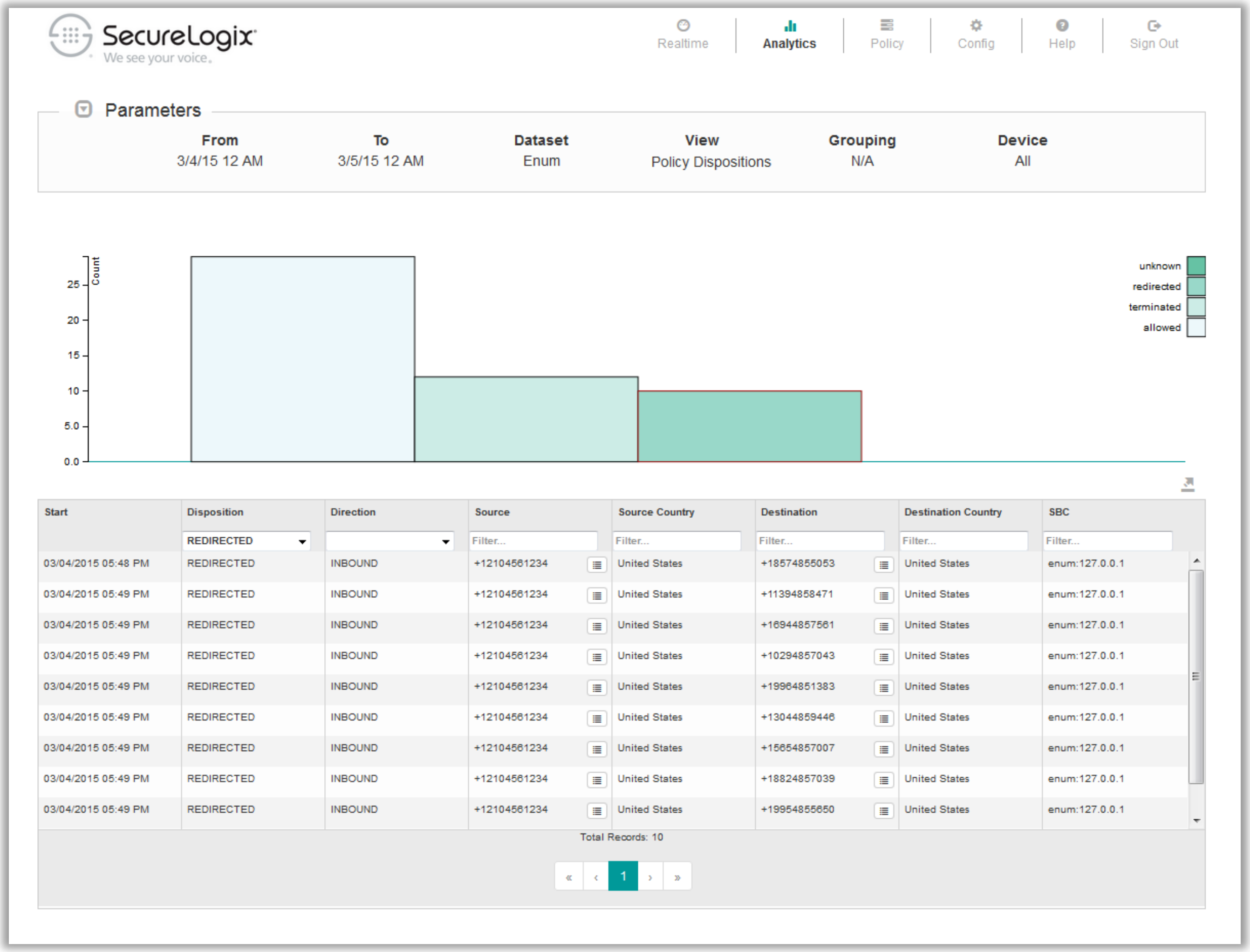
4. In the **View** field, click the down arrow and select the view for which you want to retrieve data:
   - **Average CPS**
   - **Total Calls**
   - **Policy Dispositions**
   - **Top 10 Source**
   - **Top 10 Destination**
   - **Counts by Source Country**
   - **Counts by Destination Country**
5. If you selected **Average CPS** or **Total Calls**, click the down arrow in the **Grouping** field and select the interval by which to group data: **Month**, **Day**, or **Hour**. All other views provide ungrouped data for the selected time range.
6. In the **Device** field, click the down arrow and select the SBC for which to view data, or leave the default of **All**.
7. Click **Submit**. The data is retrieved and displayed in a graph. The illustration below shows a **Policy Dispositions** view.



Parameters

From	To	Dataset	View	Grouping	Device
3/25/15 12 AM	3/27/15 12 AM	ENUM	Policy Dispositions	N/A	All

8. Clicking the graph drills down to the next-lower grouping (**Month | Day | Hour | Call Details**).. When you drill down to **Call Details**, a table appears below the graph providing individual call details, including the total number of records.



Parameters

From	To	Dataset	View	Grouping	Device
3/4/15 12 AM	3/5/15 12 AM	Enum	Policy Dispositions	N/A	All

Start	Disposition	Direction	Source	Source Country	Destination	Destination Country	SBC
	REDIRECTED		Filter...	Filter...	Filter...	Filter...	Filter...
03/04/2015 05:48 PM	REDIRECTED	INBOUND	+12104561234	United States	+18574855053	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+11394858471	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+16944857561	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+10294857043	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+19964851383	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+13044859446	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+15654857007	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+18824857039	United States	enum:127.0.0.1
03/04/2015 05:49 PM	REDIRECTED	INBOUND	+12104561234	United States	+19954855650	United States	enum:127.0.0.1

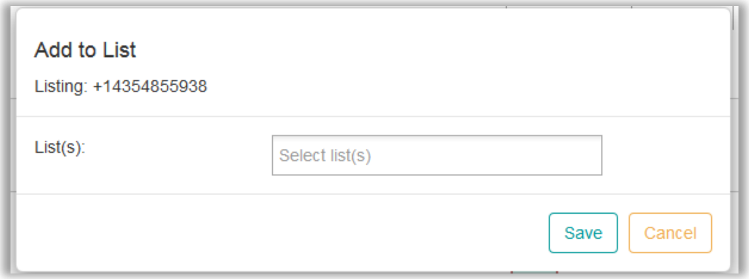
Total Records: 10

9. You can filter the display by **Disposition**, **Direction**, **Source**, **Source Country**, **Destination**, **Destination Country**, and **SBC**. The filters are applied automatically as you type or select.

10. You can sort the fields by clicking the heading of the column you want to sort by.

11. To add a phone number directly to a Blacklist or Whitelist from this view, click the **Add to List** icon to the right of the phone number. ≡

12. The **Add to List** dialog box appears.

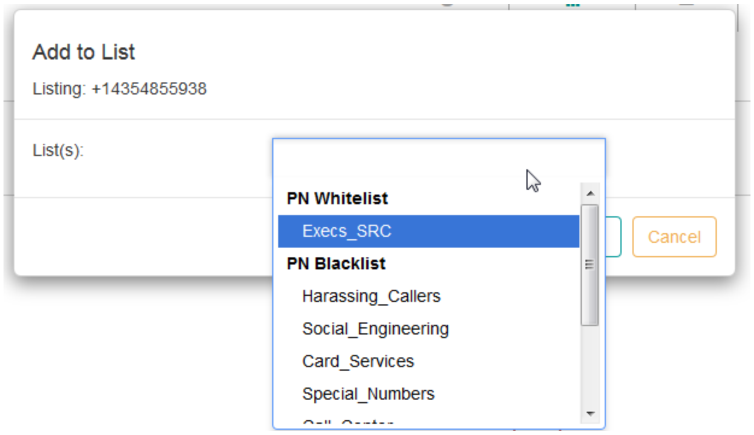


Add to List

Listing: +14354855938

List(s): Select list(s)

13. In the **List(s)** field, click the down arrow and click each List to which you want to add the selected number.



Add to List

Listing: +14354855938

List(s):

14. When you have added all applicable Lists, click **Save**. The phone number is added to the selected List(s).

    Note: To effect the change on the ENUM Server, you must reinstall the Policy that includes the List(s). The Policy is not automatically pushed.

SIP Call Detail Analytics

SIP Call Detail Analytics provides analytics views based on the SIP call data captured by the Metadata Probe.

To view SIP Call Detail Analytics

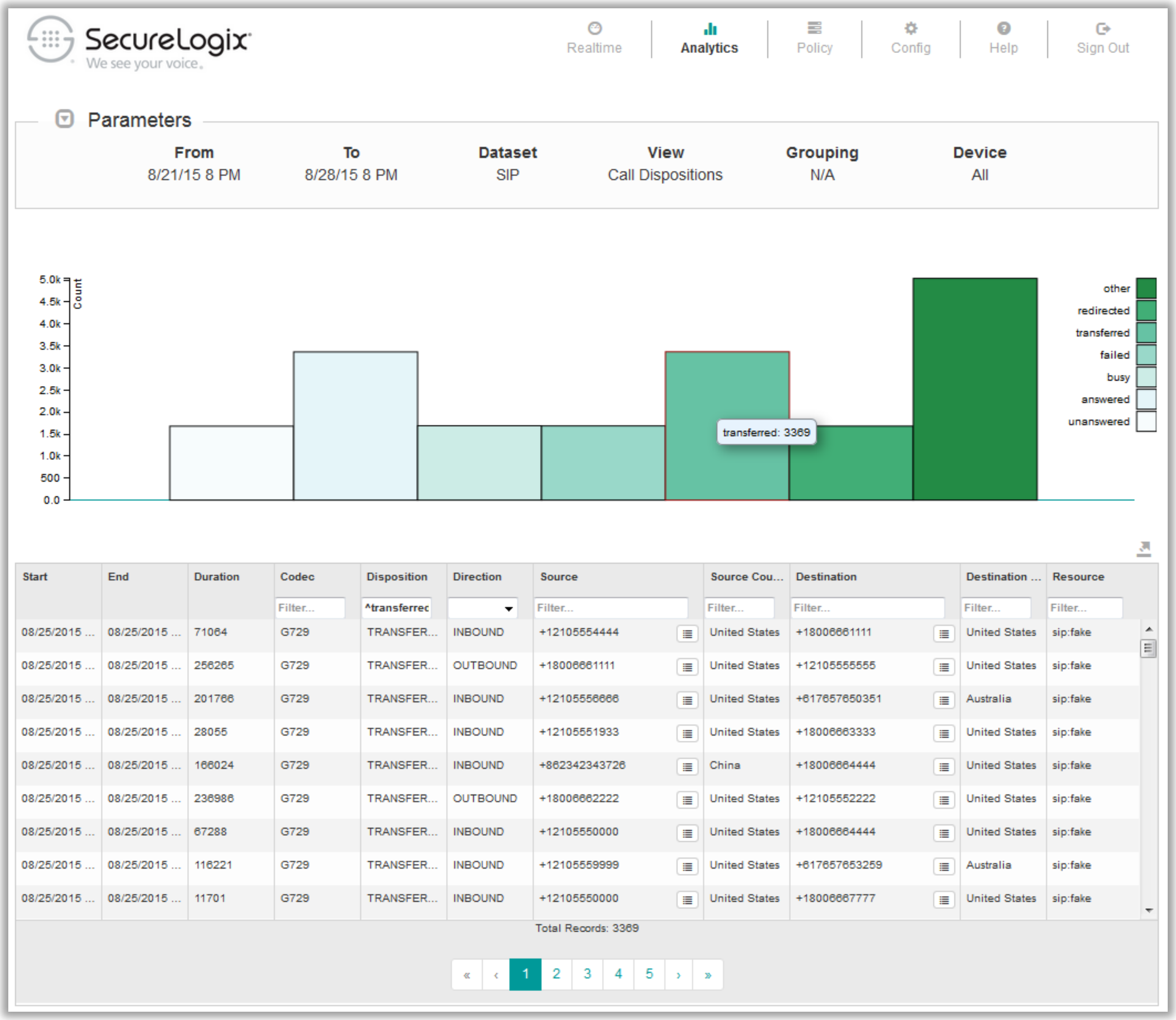
1. On the PolicyGuru main menu, click **Analytics**.

   The **Analytics** screen appears set to **Call Detail Analytics**, with **ENUM** selected as the dataset by default.

2. In the **Dataset** box, click the down arrow and click **SIP**.
3. In the **Calendar** fields, select the start and end days and times for which to retrieve data. Times are in GMT.
4. In the **View** field, click the down arrow and select the view for which you want to retrieve data:
   - **Average CPS**
   - **Total Calls**
   - **Call Dispositions**
   - **Top 10 Source**
   - **Top 10 Destination**
   - **Counts by Source Country**
   - **Counts by Destination Country**
   - **Concurrent Calls**
5. If you selected **Average CPS** or **Total Calls**, click the down arrow in the **Grouping** field and select the interval by which to group data: **Month**, **Day**, or **Hour**. All other views provide ungrouped data for the selected time range.

6. In the **Device** field, click the down arrow and select the Device for which to view data, or leave the default of **All**.

7. Click **Submit**. The data is retrieved and displayed in a graph. As with ENUM Call Detail Analytics, you can view a tooltip by hovering over the data in the graph, and clicking the display provides progressive drill-down until the call details appear in a table below the graph. The illustration below shows a **Call Dispositions** view.



Parameters

From	To	Dataset	View	Grouping	Device
8/21/15 8 PM	8/28/15 8 PM	SIP	Call Dispositions	N/A	All

Start	End	Duration	Codec	Disposition	Direction	Source	Source Cou...	Destination	Destination ...	Resource
			Filter...	^transferrec	▾	Filter...	Filter...	Filter...	Filter...	Filter...
08/25/2015 ...	08/25/2015 ...	71064	G729	TRANSFER...	INBOUND	+12105554444 ≣	United States	+18006661111 ≣	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	256265	G729	TRANSFER...	OUTBOUND	+18006661111 ≣	United States	+12105555555 ≣	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	201766	G729	TRANSFER...	INBOUND	+12105556666 ≣	United States	+617657650351 ≣	Australia	sip:fake
08/25/2015 ...	08/25/2015 ...	28055	G729	TRANSFER...	INBOUND	+12105551933 ≣	United States	+18006663333 ≣	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	166024	G729	TRANSFER...	INBOUND	+862342343726 ≣	China	+18006664444 ≣	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	236986	G729	TRANSFER...	OUTBOUND	+18006662222 ≣	United States	+12105552222 ≣	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	67288	G729	TRANSFER...	INBOUND	+12105550000 ≣	United States	+18006664444 ≣	United States	sip:fake
08/25/2015 ...	08/25/2015 ...	116221	G729	TRANSFER...	INBOUND	+12105559999 ≣	United States	+617657653259 ≣	Australia	sip:fake
08/25/2015 ...	08/25/2015 ...	11701	G729	TRANSFER...	INBOUND	+12105550000 ≣	United States	+18006667777 ≣	United States	sip:fake

Total Records: 3369

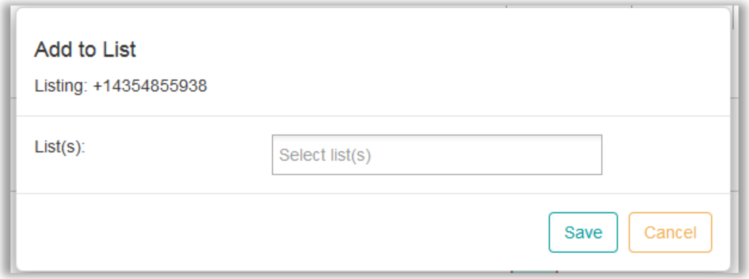
« ‹ 1 2 3 4 5 › »

8. You can filter the display by **Disposition**, **Direction**, **Source**, **Source Country**, **Destination**, **Destination Country**, and **Resource**. The filters are applied automatically as you type or select.

9. You can sort the fields by clicking the heading of the column you want to sort by.

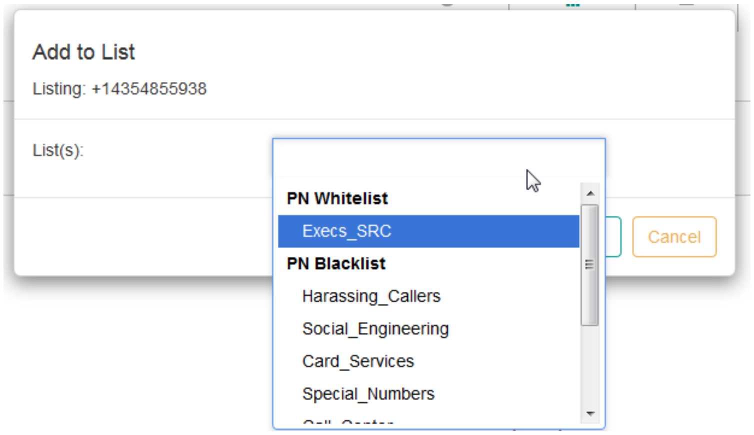
10. To add a phone number directly to a Blacklist or Whitelist from this view, click the **Add to List** icon to the right of the phone number. ≣

11. The **Add to List** dialog box appears.


Add to List

Listing: +14354855938

12. In the **List(s)** field, click the down arrow and click each List to which you want to add the selected number.


Add to List

Listing: +14354855938

13. When you have added all applicable Lists, click **Save**. The phone number is added to the selected List(s).

    Note: To effect the change on the ENUM Server, you must reinstall the Policy that includes the List(s). The Policy is not automatically pushed.

SIP Phone Number Analytics

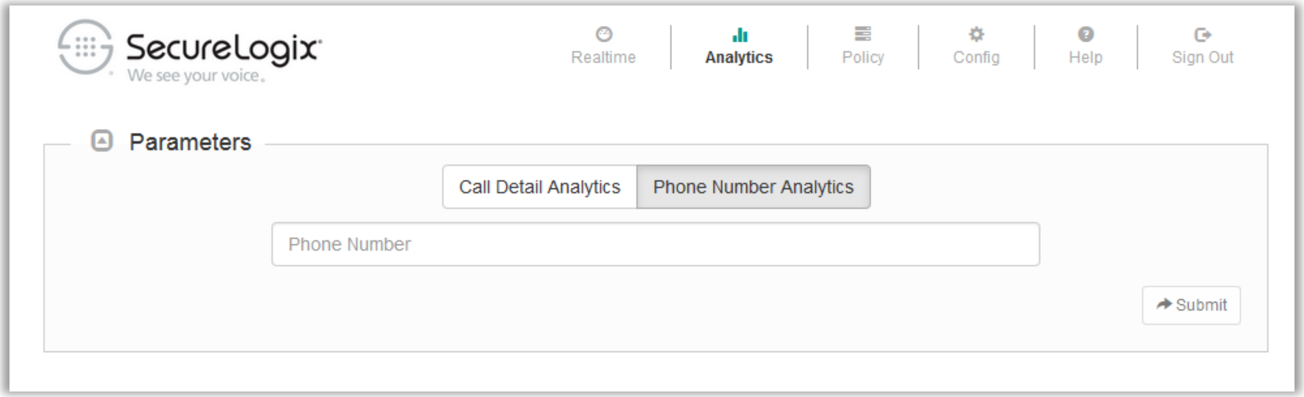
SIP Phone Number Analytics provides “drill up” analytics views for a specific phone number of interest, based on the SIP call data captured by the Metadata Probe.

To view SIP Phone Number Analytics

1. On the PolicyGuru main menu, click **Analytics**.

   The **Analytics** screen appears set to **Call Detail Analytics**, with **ENUM** selected as the dataset by default.

2. Click **Phone Number Analytics**. As mentioned, Phone Number Analytics always uses Metadata Probe data.

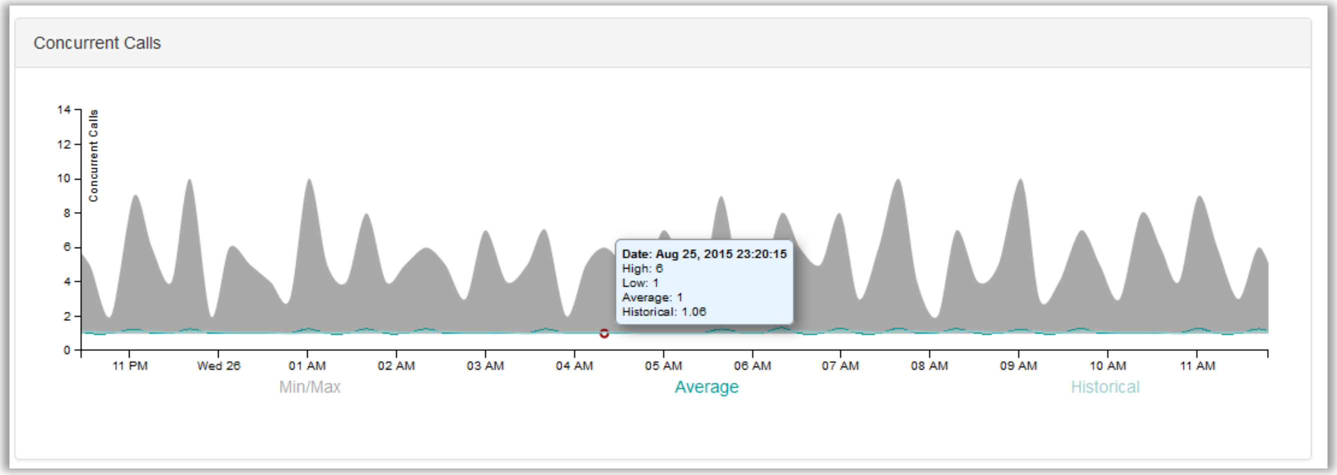


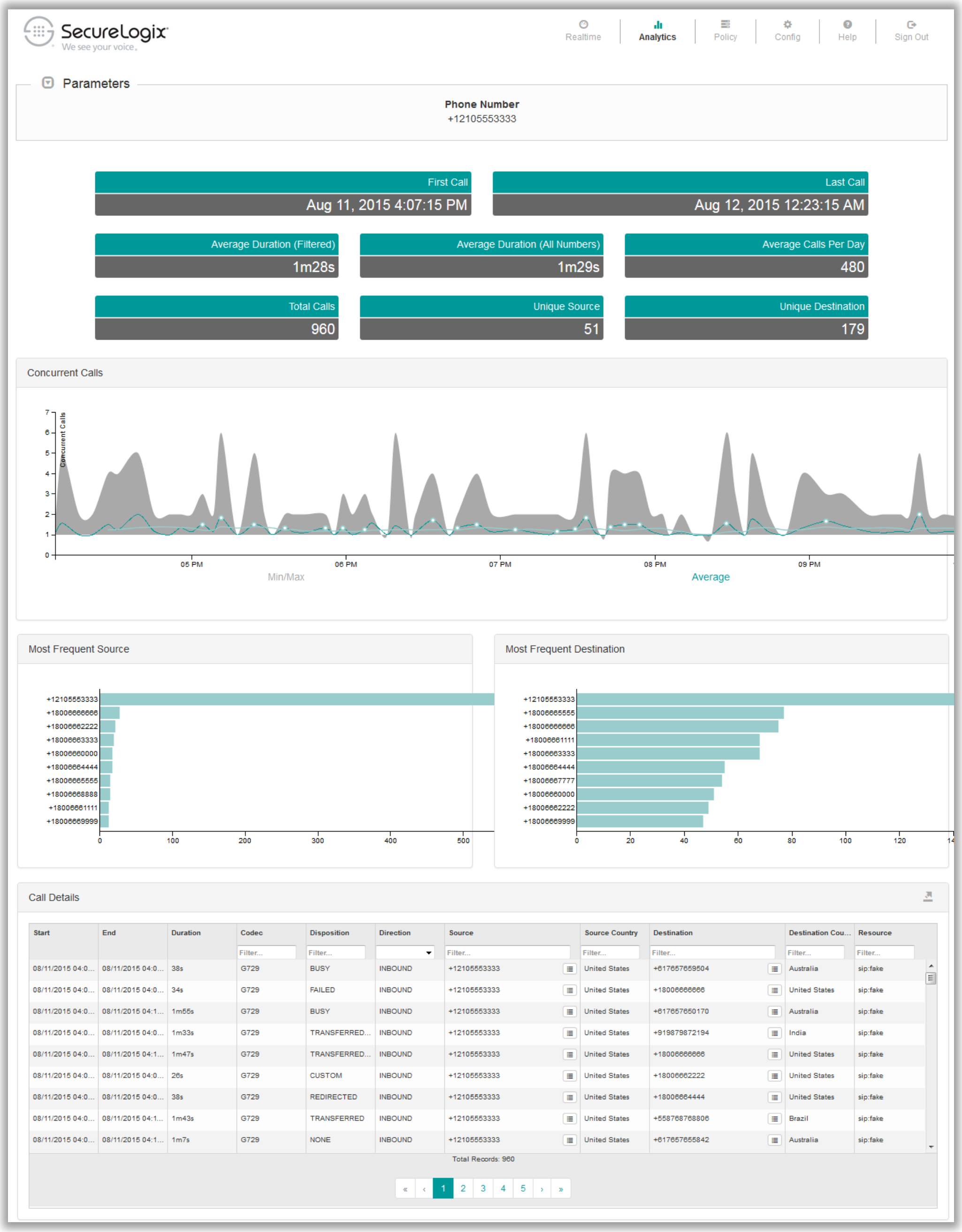
3. In the **Phone Number** box, type the phone number of interest. You can also type just a portion of a number, such as +1210, to retrieve all matching calls.
4. Click **Submit**. All data for the search string is retrieved and displayed in charts, graphs, and tables, as shown in the illustration on the next page.

   As with other views, you can sort and search the data table, export the view to a JSON file, and click the icon to add a number to a Whitelist or Blacklist.

   You can use the scroll wheel on a mouse to move and resize the **Concurrent Calls** graph.

   You can also hover your mouse over any point in the **Concurrent Calls** graph to get details about activity at that point, as shown below.





Phone Number
+12105553333

First Call	Last Call
Aug 11, 2015 4:07:15 PM	Aug 12, 2015 12:23:15 AM

Average Duration (Filtered)	Average Duration (All Numbers)	Average Calls Per Day
1m28s	1m29s	480

Total Calls	Unique Source	Unique Destination
960	51	179

Start	End	Duration	Codec	Disposition	Direction	Source	Source Country	Destination	Destination Cou...	Resource
			Filter...	Filter...		Filter...	Filter...	Filter...	Filter...	Filter...
08/11/2015 04:0...	08/11/2015 04:0...	38s	G729	BUSY	INBOUND	+12105553333	United States	+617657659504	Australia	sip:fake
08/11/2015 04:0...	08/11/2015 04:0...	34s	G729	FAILED	INBOUND	+12105553333	United States	+18006666666	United States	sip:fake
08/11/2015 04:0...	08/11/2015 04:1...	1m55s	G729	BUSY	INBOUND	+12105553333	United States	+617657650170	Australia	sip:fake
08/11/2015 04:0...	08/11/2015 04:0...	1m33s	G729	TRANSFERRED...	INBOUND	+12105553333	United States	+919879872194	India	sip:fake
08/11/2015 04:0...	08/11/2015 04:1...	1m47s	G729	TRANSFERRED...	INBOUND	+12105553333	United States	+18006666666	United States	sip:fake
08/11/2015 04:0...	08/11/2015 04:0...	26s	G729	CUSTOM	INBOUND	+12105553333	United States	+18006662222	United States	sip:fake
08/11/2015 04:0...	08/11/2015 04:0...	38s	G729	REDIRECTED	INBOUND	+12105553333	United States	+18006664444	United States	sip:fake
08/11/2015 04:0...	08/11/2015 04:1...	1m43s	G729	TRANSFERRED	INBOUND	+12105553333	United States	+558768768806	Brazil	sip:fake
08/11/2015 04:0...	08/11/2015 04:1...	1m7s	G729	NONE	INBOUND	+12105553333	United States	+617657655842	Australia	sip:fake

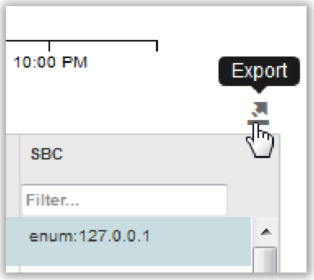
Total Records: 960

Exporting Analytics Data

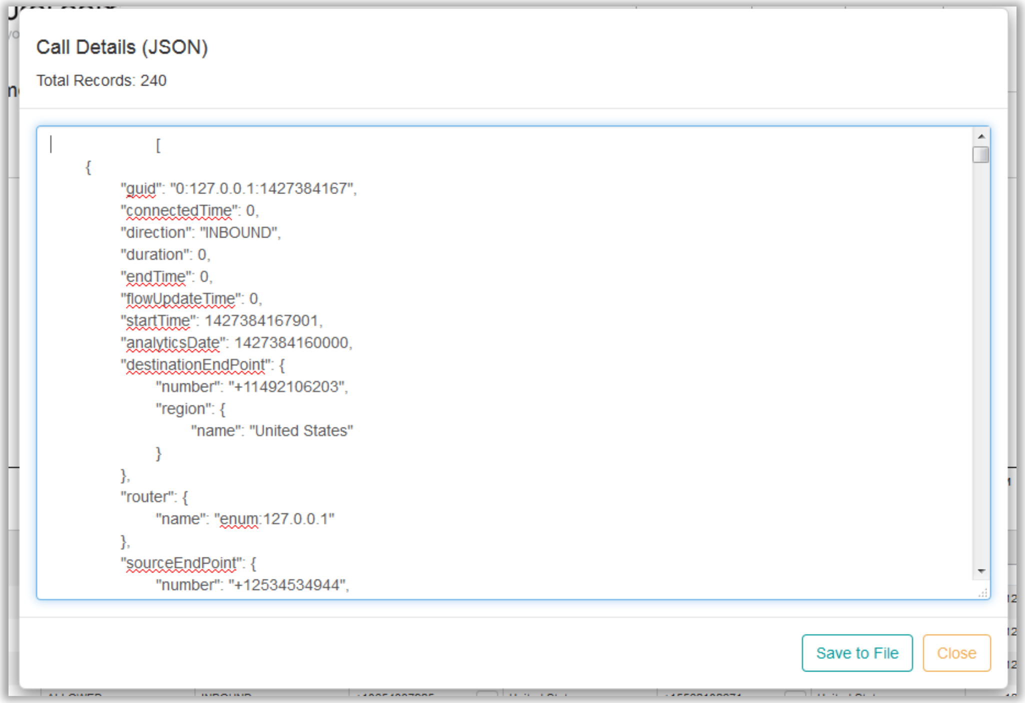
You can export Analytics data in JSON format for further offline analysis and reporting.

To export Analytics data

1. Above **Resource** on the **Call Detail** table, click the **Export** icon.



2. The **Call Details (JSON)** dialog box appears containing the data for the current page of records.


Call Details (JSON)

Total Records: 240

```
[
    {
        "guid": "0:127.0.0.1:1427384167",
        "connectedTime": 0,
        "direction": "INBOUND",
        "duration": 0,
        "endTime": 0,
        "flowUpdateTime": 0,
        "startTime": 1427384167901,
        "analyticsDate": 1427384160000,
        "destinationEndPoint": {
            "number": "+11492106203",
            "region": {
                "name": "United States"
            }
        },
        "router": {
            "name": "enum:127.0.0.1"
        },
        "sourceEndPoint": {
            "number": "+12534534944",
```

3. For a short number of results, you can simply copy the information out of the display, or for longer result sets, you can click **Save to File** to save it to a file.

System Configuration

General System Configuration

Use the procedures below to configure alerting, mid-call digit storage, and users and passwords, and to view/configure connected ENUM Servers and Metadata Probes.

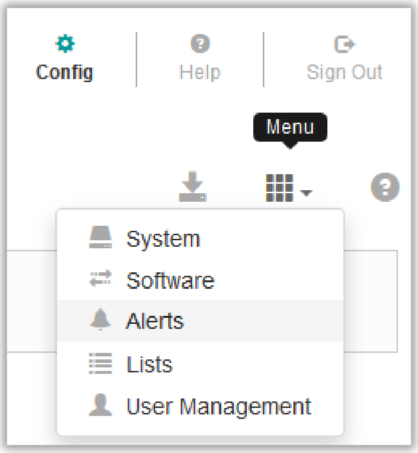
Alerting Configuration

The PolicyGuru System supports SNMP, Syslog, and Email alerting, in addition to the alert messages on the **Realtime** screen.

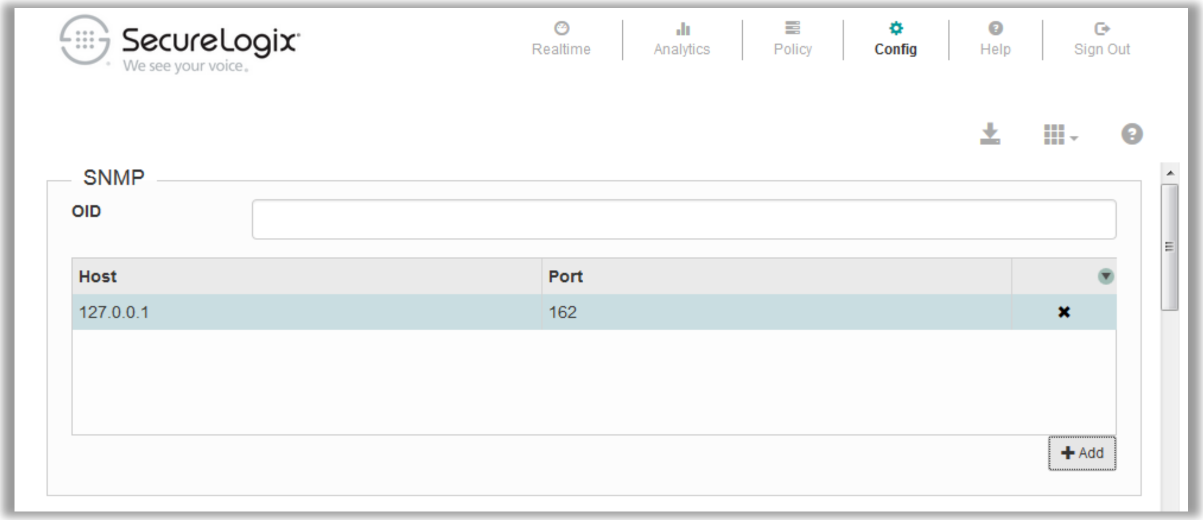
Configuring SNMP Alerts

To configure SNMP alerts

1. On the PolicyGuru main menu, click **Config**..
2. Click the **Show Configuration Menu** icon at the right.



3. In the drop-down list, click **Alerts**. The **Alerts** screen appears.



OID

Host	Port	
127.0.0.1	162	✖

4. In the **SNMP** area, click **Add**.
5. A new row appears with placeholder values for **Host** and **Port**, as shown in the illustration above.
6. In the **OID** field, type the OID.
7. Double-click in the **Host** field to enable editing, and then type the IP address of the SNMP Agent host.
8. Double-click in the **Port** field to enable editing, and then type the port for the SNMP Agent.
9. Click the **Save** icon. ⤓

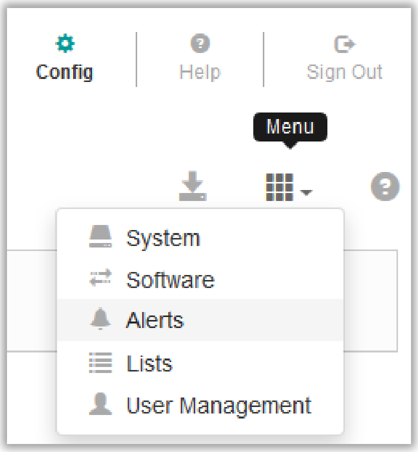
Configuring Email Notifications

Before email alerts can be generated, the **Reply-To** address and the list of email addresses to which alerts are to be sent must be specified. Additionally, the SMTP information for the email host must be specified. The procedure below explains adding the necessary email addresses. See “Specifying SMTP Information for Email Alerts” on page 69 for instructions.

IMPORTANT: Do not configure email alerting settings in the GUI if no SMTP information has been configured in the server file, to avoid filling up the logs with errors related to an unavailable email server.

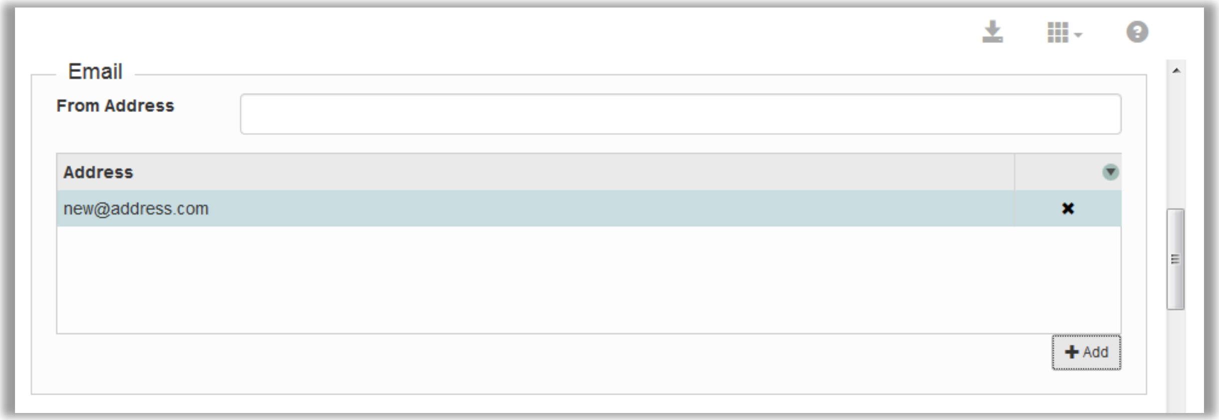
To configure Email alerts

1. On the PolicyGuru main menu, click **Config**.
2. Click the **Show Configuration Menu** icon at the right.



3. In the drop-down list, click **Alerts**. The **Alerts** screen appears.

4. In the **Email** area, click **Add**.



5. A new row is added with a placeholder email address. Double-click the **Address** field and replace the placeholder with an email address to which email alerts are to be sent.

6. Repeat steps 1 and 2 for each email address that is to receive email alerts.

7. In the **From Address** field, type the email address that is to appear in the **From** field in PolicyGuru email alerts.

8. Click the **Save** icon. ⤓

Specifying SMTP Information for Email Alerts

To specify SMTP information

Note: See the section above for information for supplying recipients and **Reply-to** address.

- On the Mediation Server, edit the following files:

  /opt/ngp/config/network/jboss-email-smtp-host

  Supply the IP address of the email server.

  /opt/ngp/config/network/jboss-email-smtp-port

Supply the port of the email server, if other than the default SMTP port 23.

/opt/ngp/config/network/jboss-email-smtp-user

Supply the username for the email server, if one is used.

/opt/ngp/config/network/jboss-email-smtp-password

Supply the password for the email server user, if one is used.

/opt/ngp/config/network/jboss-email-smtp-ssl

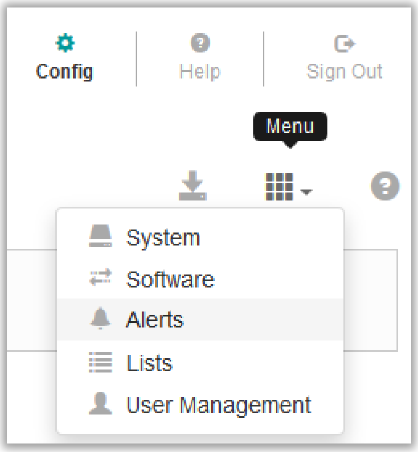
If SSL is not used, specify **false**. If it is used, specify **true**.

IMPORTANT: If the NGP service is running, you must restart it after specifying the SMTP information in the file.

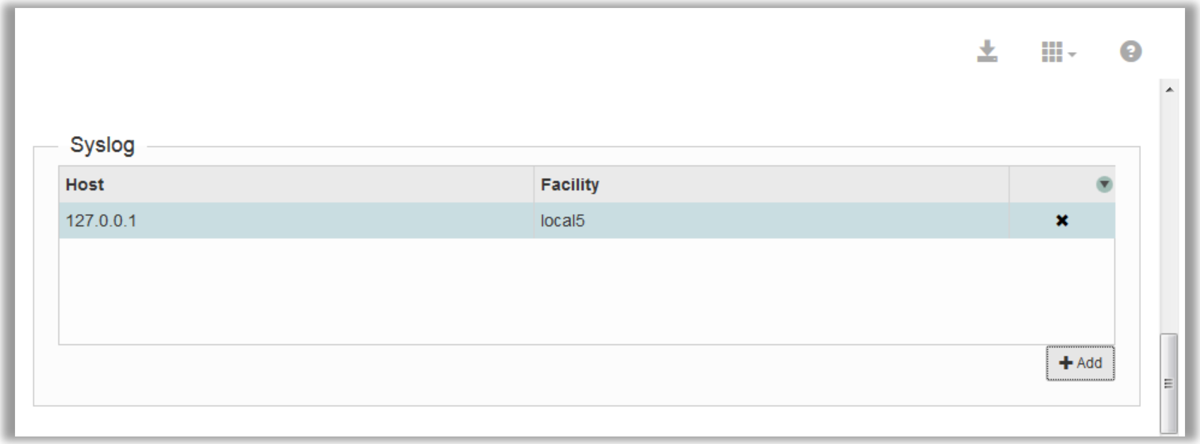
Configuring Syslog Alerts

To configure Syslog alerts

1. On the PolicyGuru main menu, click **Config**.
2. Click the **Show Configuration Menu** icon at the right.



3. In the drop-down list, click **Alerts**. The **Alerts** screen appears.


Syslog

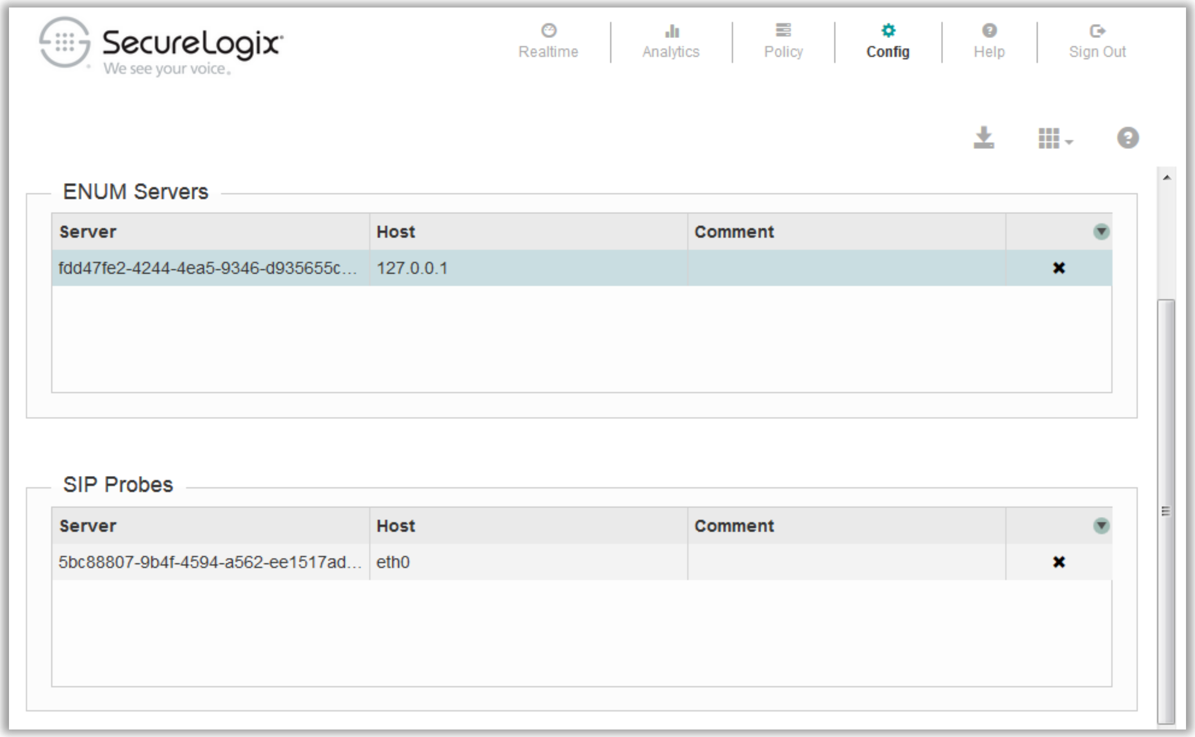
Host	Facility	
127.0.0.1	local5	✖

4. In the **Syslog** area, click **Add**. A new row is added with placeholder values, as shown in the illustration above.
5. Double-click the **Host** field and type the IP address of the Syslog host.
6. Double-click the **Facility** field, click the down arrow, and then select the facility keyword (that is, the facility code or value).
7. Click the **Save** icon.

Connected ENUM Server and Probe Configuration

To view the connected ENUM Servers and Metadata Probes

1. On the PolicyGuru main menu, click **Config**.
2. The **Config** tab appears showing the **System** screen. (Select **System** from the configuration menu if you were already on another screen of the **Config** tab.)



ENUM Servers

Server	Host	Comment	
fdd47fe2-4244-4ea5-9346-d935655c…	127.0.0.1		✖

SIP Probes

Server	Host	Comment	
5bc88807-9b4f-4594-a562-ee1517ad…	eth0		✖

Changing the Listener IP Addresses for ENUM Servers

To change the IP address on which an ENUM Server listens for ENUM requests

- On the **System** screen of the **Config** tab, double-click the **Host** field for the applicable device and type the new IP address. (It must already be configured on the device.)

Changing the Ethernet Port on Which Probes Monitor

To change the Ethernet port on which a Metadata Probe monitors for SIP messages

- On the **System** screen of the **Config** tab, double-click the **Host** field for the applicable device and type the Ethernet port. (It must already be configured on the device.)

Adding Identifying Comments to Connected ENUM Servers/Probes

To add a comment providing identifying information about the ENUM Servers/Probes

- On the **System** screen of the **Config** tab, double-click in the **Comment** field and type a comment.

User and Password Management

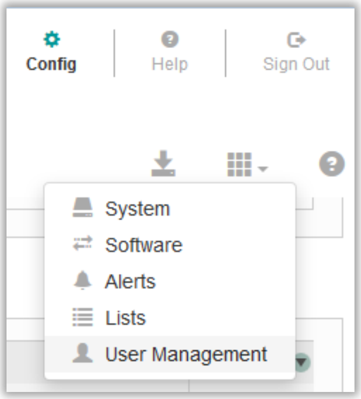
The sections below explain how to change the Management Interface **admin** password and provides information about using external LDAP authentication and group-based application permissions.

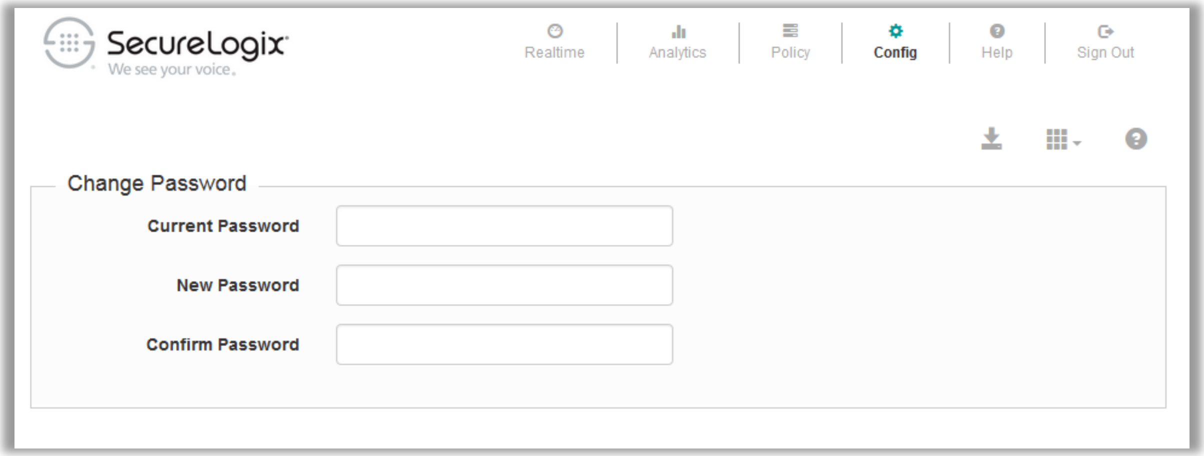
Changing the Management GUI Admin User Password

Note: The Management Interface provides a single user account, **admin**. You cannot add other users to the interface unless you use external LDAP for authentication. See “ External LDAP Authentication and Group Permissions” on page 73 for information.

To change the Management GUI default admin user password

1. On the PolicyGuru main menu, click **Config**. The **Config** screen appears.
2. Click the **Show Configuration Menu** icon at the right and click **User Management**.





Change Password

Current Password

New Password

Confirm Password

3. In the **Current Password** box, type the current **admin** user password.
4. In the **Update Password** and **Confirm Password** boxes, type the new password. Use a unique password that meets complexity and length security standards for your organization.

External LDAP Authentication and Group Permissions

You can configure the PolicyGuru Solution to use external LDAP for authentication. External OpenLDAP and Active Directory servers are supported. When you enable external LDAP authentication, the default user account is disabled and cannot be used to log in to the system.

When external LDAP is used, application-level user permissions govern access the PolicyGuru applications. Users only see the PolicyGuru main menu icons for the web applications for which they are a member of an authorized LDAP group. This applies to the **Realtime**, **Analytics**, **Policy**, and **Config** applications.

See the *PolicyGuru® System Administration Guide* for detailed information about configuring the PolicyGuru Solution to use external LDAP authentication and group-based application permissions.

Appendices

Appendix A: List Import Scripts

createlist_file_2.1.sh

See “Static List Import” on page 29 for instructions for use.

```
#!/bin/bash

if test $# -lt 4
then
    echo ""
    echo "Usage: ./createlist_file_2.1.sh <server> <list name> <list type>
<list file> [batch size] [username] [password]"
    echo ""
    echo "where:"
    echo "      <server> is the IP address of the PolicyGuru Mediation
Server"
    echo "      <list name> is the name of the List to create on the
PolicyGuru system"
    echo "      <list type> is 0 for PN Whitelist, 1 for PN Blacklist, 2
for Regex Whitelist, or 3 for Regex Blacklist"
    echo "      <list file> is the name of the text file containing
listings (one listing per line)"
    echo "      [batch size] is the optional number of listings to submit
at a time.  If omitted, the batch size will default to 100."
    echo "      [username] is the optional username for PolicyGuru
application login.  If omitted, the user will be prompted for username
input."
    echo "      [password] is the optional password for PolicyGuru
application login.  If omitted, the user will be prompted for password
input."
    echo ""
    exit 1
fi

server=$1
listname=$2
listtype=$3
listfile=$4
batchsize=$5
username=$6
password=$7

if [ ! -f "$listfile" ]
then
    echo "Invalid file $listfile"
    exit 1
fi

if [ "$batchsize" == "" ]
then
    batchsize=100
fi

if [ "$username" == "" ]
then
    echo -n "Enter PolicyGuru Username: "
    read username
fi

if [ "$password" == "" ]
then
    echo -n "Enter PolicyGuru Password: "
    read -s password
    echo ""
fi

##response=$(curl -k -f -w HTTPResponseCode%{http_code} -X PUT -H
"Content-Type: application/json" -d '{"userId":"admin",
"password":"SecureLogix1!"}'
https://$server:8443/mgmt/rest/security/login)
command="curl -k -f -w HTTPResponseCode%{http_code} -X PUT -H \"Content-
Type: application/json\" -d '{\"userId\":\"$username\",
\"password\":\"$password\"}'
https://$server:8443/mgmt/rest/security/login"
response=$(eval $command)
if [ $(echo $response |awk -F'HTTPResponseCode' '{print $2}') != "200" ]
then
    echo "login failed, verify server, username, and password settings"
    exit 1
fi

token=$(echo $response | awk -F'"token":"' '{print $2}' | awk -F'"'
'{print $1}')

verify=$(curl -k -f -w %{http_code} -X GET -H "Access-Token: $token"
https://$server:8443/mgmt/rest/security/login)
if [ $verify != "200" ]
then
    echo "token verification failed"
    exit 1
fi

command="curl -k -f -X POST -H \"Access-Token: $token\" -H \"Content-Type:
application/json\" -d '{\"name\":\"$listname\", \"type\":\"$listtype\",
\"listings\":[]}' https://$server:8443/mgmt/rest/lists"
##echo "list create: $command"
eval $command
if test $? -ne 0
then
    echo "create list command failed, result=$?"
    echo "the list may already exist"
    exit 2
```

```
fi

sleep 1
#sleep 30

count=0
overall=0
while read -r listing
do
    #Add a leading backslash to any backslashes that exists in the read in
listing to allow proper curl operation with the MS
    listing=$(echo $listing | sed 's/\\/\\\\/g')

    if test $count -eq 0
    then
        command=
        listingstring="time curl -k -f -X PUT -H \"Access-Token: $token\"
-H \"Content-Type: application/json\" -d '{\"name\":\"$listname\",
\"addListings\":[{\"value\":\"$listing\"}"
    else
        listingstring=", {\"value\":\"$listing\"}"
    fi
    command=${command}${listingstring}
    count=$(($count + 1))
    overall=$(($overall + 1))
    if test $count -ge $batchsize
    then
        endcommand="]}' https://$server:8443/mgmt/rest/lists/$listname"
        command=${command}${endcommand}
        #echo "final command: $command"
        count=0

        for ((j=0;j<3;j++))
        do
            eval $command
            commandresult=$?
            if test $commandresult -eq 0
            then
                break
            elif test $commandresult -eq 35
            then
                if test $j -eq 2
                then
                    echo "command=$command"
                    echo "commandresult=$commandresult"
                    exit 3
                fi
            else
                echo "command=$command"
                echo "commandresult=$commandresult"
                echo "verify there are no repeated values (including
repeated blank lines)"
                exit 4
            fi
            sleep 0.02
            #sleep 30
        done
        echo "$overall Listings Submitted"
    fi

    sleep 0.02
    #sleep 30

done < "$listfile"

if test $count -gt 0
then
    endcommand="]}' https://$server:8443/mgmt/rest/lists/$listname"
    command=${command}${endcommand}
    #echo "final command: $command"

    for ((j=0;j<3;j++))
    do
        eval $command
        commandresult=$?
        if test $commandresult -eq 0
        then
            break
        elif test $commandresult -eq 35
        then
            if test $j -eq 2
            then
                echo "command=$command"
                echo "commandresult=$commandresult"
                exit 3
            fi
        else
            echo "command=$command"
            echo "commandresult=$commandresult"
            echo "verify there are no repeated values (including repeated
blank lines)"
            exit 4
        fi
        sleep 0.02
        #sleep 30
    done
    echo "$overall Listings Submitted"
fi
```

createlist_range_2.1.sh

See “Range List Import” on page 31 for instructions for use.

```
#!/bin/bash

if test $# -lt 6
then
    echo ""
    echo "Usage: ./createlist_range_2.1.sh <server> <list name> <list 
type> <prefix> <value length> <count> [batch size] [username] 
[password]"
    echo ""
    echo "where:"
    echo "      <server> is the IP address of the PolicyGuru Mediation 
Server"
    echo "      <list name> is the name of the List to create on the 
PolicyGuru system"
    echo "      <list type> is 0 for PN Whitelist, 1 for PN Blacklist, 
2 for Regex Whitelist, or 3 for Regex Blacklist"
    echo "      <prefix> is a static string that starts every listing 
(such as +1210)"
    echo "      <value length> is the size of the number of additional 
characters that are appended to the prefix (such as 6 to append 6 
characters to the prefix)"
    echo "      <count> is the total number of listings to create (such 
as 5000 to create a list with 5000 listings)"
    echo "      [batch size] is the optional number of listings to 
submit at a time.  If omitted, the batch size will default to 100."
    echo "      [username] is the optional username for PolicyGuru 
application login.  If omitted, the user will be prompted for username 
input."
    echo "      [password] is the optional password for PolicyGuru 
application login.  If omitted, the user will be prompted for password 
input."
    echo ""
    exit 1
fi

server=$1
listname=$2
listtype=$3
prefix=$4
valuelength=$5
count=$6
batchsize=$7
username=$8
password=$9

if [ "$batchsize" == "" ]
then
    batchsize=100
fi
```

```
if [ "$username" == "" ]
then
    echo -n "Enter PolicyGuru Username: "
    read username
fi

if [ "$password" == "" ]
then
    echo -n "Enter PolicyGuru Password: "
    read -s password
    echo ""
fi

command="curl -k -f -w HTTPResponseCode%{http_code} -X PUT -H
\"Content-Type: application/json\" -d '{\"userId\":\"$username\",
\"password\":\"$password\"}'
https://$server:8443/mgmt/rest/security/login"
response=$(eval $command)
if [ $(echo $response |awk -F'HTTPResponseCode' '{print $2}') != "200"
]
then
    echo "login failed, verify server, username, and password settings"
    exit 1
fi

token=$(echo $response | awk -F'"token":"' '{print $2}' | awk -F'"'
'{print $1}')

verify=$(curl -k -f -w %{http_code} -X GET -H "Access-Token: $token"
https://$server:8443/mgmt/rest/security/login)
if [ $verify != "200" ]
then
    echo "token verification failed"
    exit 1
fi

command="curl -k -f -X POST -H \"Access-Token: $token\" -H \"Content-
Type: application/json\" -d '{\"name\":\"$listname\",
\"type\":\"$listtype\", \"listings\":[]}'
https://$server:8443/mgmt/rest/lists"
##echo "list create: $command"
eval $command
if test $? -ne 0
then
    echo "create list command failed, result=$?"
    exit 2
fi

sleep 1

for ((i=0;i<count;))
do
```

```
    command="curl -k -f -X PUT -H \"Access-Token: $token\" -H
\"Content-Type: application/json\" -d '{\"name\":\"$listname\",
\"addListings\":["
    ##echo "command start: $command"
    for ((j=0;j<$batchsize;j++))
    do
        if test $j -eq 0
        then
            listingstring="{\"value\":\"$prefix$(printf
%0${valuelength}d $i)\"}"
        else
            listingstring=", {\"value\":\"$prefix$(printf
%0${valuelength}d $i)\"}"
        fi
        command=${command}${listingstring}
        ##echo "current command: $command"
        i=$(($i + 1))
        if test $i -ge $count
        then
            break
        fi
    done
    endcommand="]}' https://$server:8443/mgmt/rest/lists/$listname"
    command=${command}${endcommand}
    ##echo "final command: $command"

    for ((k=0;k<3;k++))
    do
        eval $command
        commandresult=$?
        if test $commandresult -eq 0
        then
            break
        elif test $commandresult -eq 35
        then
            if test $k -eq 2
            then
                echo "command=$command"
                echo "commandresult=$commandresult"
                exit 3
            fi
        else
            echo "command=$command"
            echo "commandresult=$commandresult"
            exit 4
        fi
        sleep 0.02
    done
    sleep 0.02

done
echo "$count Listings Submitted"
```

addtolist_file_2.1.sh

See “Importing Listings Into An Existing List” on page 32 for instructions for use.

```
#!/bin/bash

if test $# -lt 3
then
    echo ""
    echo "Usage: ./addtolist_file_2.1.sh <server> <list name> <list file>
[batch size] [username] [password]"
    echo ""
    echo "where:"
    echo "      <server> is the IP address of the PolicyGuru Mediation Server"
    echo "      <list name> is the name of the List to add listings to"
    echo "      <list file> is the name of the text file containing listings
(one listing per line)"
    echo "      [batch size] is the number of listings to submit at a time.  If
omitted, the batch size will default to 100."
    echo "      [username] is the optional username for PolicyGuru application
login.  If omitted, the user will be prompted for username input."
    echo "      [password] is the optional password for PolicyGuru application
login.  If omitted, the user will be prompted for password input."
    echo ""
    exit 1
fi

server=$1
listname=$2
listfile=$3
batchsize=$4
username=$5
password=$6


if [ ! -f "$listfile" ]
then
    echo "Invalid file $listfile"
    exit 1
fi

if [ "$batchsize" == "" ]
then
    batchsize=100
fi

if [ "$username" == "" ]
then
    echo -n "Enter PolicyGuru Username: "
    read username
fi
```

```
if [ "$password" == "" ]
then
    echo -n "Enter PolicyGuru Password: "
    read -s password
    echo ""
fi

##response=$(curl -k -f -w HTTPResponseCode%{http_code} -X PUT -H "Content-
Type: application/json" -d '{"userId":"admin", "password":"SecureLogix1!"}'
https://$server:8443/mgmt/rest/security/login)
command="curl -k -f -w HTTPResponseCode%{http_code} -X PUT -H \"Content-Type:
application/json\" -d '{\"userId\":\"$username\", \"password\":\"$password\"}'
https://$server:8443/mgmt/rest/security/login"
response=$(eval $command)
if [ $(echo $response |awk -F'HTTPResponseCode' '{print $2}') != "200" ]
then
    echo "login failed, verify server, username, and password settings"
    exit 1
fi

token=$(echo $response | awk -F'"token":"' '{print $2}' | awk -F'"' '{print
$1}')

verify=$(curl -k -f -w %{http_code} -X GET -H "Access-Token: $token"
https://$server:8443/mgmt/rest/security/login)
if [ $verify != "200" ]
then
    echo "token verification failed"
    exit 1
fi

sleep 1

count=0
overall=0
while read -r listing
do
    #Add a leading backslash to any backslashes that exists in the read in
listing to allow proper curl operation with the MS
    listing=$(echo $listing | sed 's/\\/\\\\/g')

    if test $count -eq 0
    then
        command=
        listingstring="curl -k -f -X PUT -H \"Access-Token: $token\" -H
\"Content-Type: application/json\" -d '{\"name\":\"$listname\",
\"addListings\":[{\"value\":\"$listing\"}"
    else
        listingstring=", {\"value\":\"$listing\"}"
    fi
    command=${command}${listingstring}
    count=$(($count + 1))
    overall=$(($overall + 1))
    if test $count -ge $batchsize
    then
        endcommand="]}' https://$server:8443/mgmt/rest/lists/$listname"
        command=${command}${endcommand}
        #echo "final command: $command"
        count=0

        for ((j=0;j<3;j++))
        do
            eval $command
            commandresult=$?
            if test $commandresult -eq 0
            then
                break
            elif test $commandresult -eq 35
            then
                if test $j -eq 2
                then
                    echo "command=$command"
                    echo "commandresult=$commandresult"
                    exit 3
                fi
            else
                echo "command=$command"
                echo "commandresult=$commandresult"
                echo "verify that the list already exists and that there are no
repeated values in the file (including repeated blank lines)"
                exit 4
            fi
            sleep 1
        done
        echo "$overall Listings Submitted"

    fi
    sleep 1

done < "$listfile"

if test $count -gt 0
then
    endcommand="]}' https://$server:8443/mgmt/rest/lists/$listname"
    command=${command}${endcommand}
    #echo "final command: $command"

    for ((j=0;j<3;j++))
    do
        eval $command
        commandresult=$?
        if test $commandresult -eq 0
        then
            break
        elif test $commandresult -eq 35
        then
            if test $j -eq 2
            then
                echo "command=$command"
                echo "commandresult=$commandresult"
                exit 3
            fi
        else
            echo "command=$command"
            echo "commandresult=$commandresult"
            echo "verify that the list already exists and that there are no
repeated values in the file(including repeated blank lines)"
            exit 4
        fi
        sleep 1
    done
    echo "$overall Listings Submitted"
fi
```

deletelist_2.1.sh

See “Delete List Script” on page 33 for instructions for use.

```
#!/bin/bash

if test $# -lt 2
then
    echo ""
    echo "Usage: ./deletelist_2.1.sh <server> <list name> [batch size]
[username] [password]"
    echo ""
    echo "where:"
    echo "      <server> is the IP address of the PolicyGuru Mediation Server"
    echo "      <list name> is the name of the List to delete from the system"
    echo "      [batch size] is the optional number of listings to delete at a
time.  If omitted, the batch size will default to 100."
    echo "      [username] is the optional username for PolicyGuru application
login.  If omitted, the user will be prompted for username input."
    echo "      [password] is the optional password for PolicyGuru application
login.  If omitted, the user will be prompted for password input."
    echo ""
    exit 1
fi

server=$1
listname=$2
batchsize=$3
username=$4
password=$5
pagesize=500

if [ "$batchsize" == "" ]
then
    batchsize=100
fi

if [ "$username" == "" ]
then
    echo -n "Enter PolicyGuru Username: "
    read username
fi

if [ "$password" == "" ]
then
    echo -n "Enter PolicyGuru Password: "
    read -s password
    echo ""
fi
```

```
##response=$(curl -k -f -w HTTPResponseCode%{http_code} -X PUT -H "Content-
Type: application/json" -d '{"userId":"admin", "password":"SecureLogix1!"}'
https://$server:8443/mgmt/rest/security/login)
command="curl -k -f -w HTTPResponseCode%{http_code} -X PUT -H \"Content-Type:
application/json\" -d '{\"userId\":\"$username\", \"password\":\"$password\"}'
https://$server:8443/mgmt/rest/security/login"
response=$(eval $command)
if [ $(echo $response |awk -F'HTTPResponseCode' '{print $2}') != "200" ]
then
    echo "login failed, verify server, username, and password settings"
    exit 1
fi

token=$(echo $response | awk -F'"token":"' '{print $2}' | awk -F'"' '{print
$1}')

verify=$(curl -k -f -w %{http_code} -X GET -H "Access-Token: $token"
https://$server:8443/mgmt/rest/security/login)
if [ $verify != "200" ]
then
    echo "token verification failed"
    exit 1
fi

sleep 1

pagesleft=true
pagenumber=1
while [ "$pagesleft" == "true" ]
do
    result=$(curl -k -f -w HTTPResponseCode%{http_code} -X GET -H "Access-
Token: $token"
https://$server:8443/mgmt/rest/lists/$listname?pageNumber=$pagenumber\&pageSize
=$pagesize > .$listname.$pagenumber)
    if [ $(echo $response |awk -F'HTTPResponseCode' '{print $2}') != "200" ]
    then
        echo "failed to get list:$listname, page:$pagenumber, size:$pagesize"
        exit 2
    fi
    if [ "$(cat .$listname.$pagenumber | grep "listings\":\[\]" | awk -
F'HTTPResponseCode' '{print $2}')" == "200" ]
    then
        pagesleft=false
    else
        ((pagenumber++))
    fi
done

touch .$listname.id
touch .$listname.number
for ((i=1;i<pagenumber;i++))
do
```

```
    cat .$listname.$i | awk -F']}HTTPResponseCode200' '{print $1}' | awk -F'['
'{print $2}' |sed 's/},{/}\n{/g' > .$listname.id.$i
    cat .$listname.id .$listname.id.$i > .$listname.id.temp
    mv .$listname.id.temp .$listname.id
    while read id
    do
        echo "$id" | awk -F'"value":"' '{print $2}' | awk -F'"}' '{print $1}'
>> .$listname.number.$i
    done < .$listname.id.$i
    cat .$listname.number .$listname.number.$i > .$listname.number.temp
    mv .$listname.number.temp .$listname.number
done

count=0
overall=0
while read id
do
    #Add a leading backslash to any backslashes that exists in the read in id
to allow proper curl operation with the MS
    id=$(echo $id | sed 's/\\/\\\\/g')

    if test $count -eq 0
    then
        command=
        idstring="curl -k -f -X PUT -H \"Access-Token: $token\" -H \"Content-
Type: application/json\" -d '{\"name\":\"$listname\", \"removeListings\":[$id"
    else
        idstring=", $id"
    fi
    command=${command}${idstring}
    count=$(($count + 1))
    overall=$(($overall + 1))
    if test $count -ge $batchsize
    then
        endcommand="]}' https://$server:8443/mgmt/rest/lists/$listname"
        command=${command}${endcommand}
        #echo "final command: $command"
        count=0

        for ((j=0;j<3;j++))
        do
            eval $command
            commandresult=$?
            if test $commandresult -eq 0
            then
                break
            elif test $commandresult -eq 35
            then
                if test $j -eq 2
                then
                    echo "command=$command"
                    echo "commandresult=$commandresult"
                    exit 3
                fi
            else
                echo "command=$command"
                echo "commandresult=$commandresult"
                exit 4
            fi
            sleep 1
        done
        echo "$overall Listings Deleted"
    fi

    sleep 1

done < ".$listname.id"

if test $count -gt 0
then
    endcommand="]}' https://$server:8443/mgmt/rest/lists/$listname"
    command=${command}${endcommand}
    #echo "final command: $command"

    for ((j=0;j<3;j++))
    do
        eval $command
        commandresult=$?
        if test $commandresult -eq 0
        then
            break
        elif test $commandresult -eq 35
        then
            if test $j -eq 2
            then
                echo "command=$command"
                echo "commandresult=$commandresult"
                exit 3
            fi
        else
            echo "command=$command"
            echo "commandresult=$commandresult"
            exit 4
        fi
        sleep 1
    done
fi
echo "$overall Listings Deleted"
#delete the group itself
response=$(curl -k -f -w HTTPResponseCode%{http_code} -X DELETE -H "Access-
Token: $token" https://$server:8443/mgmt/rest/lists/$listname)
responsecode=$(echo $response |awk -F'HTTPResponseCode' '{print $2}')
if [ "$responsecode" != "200" ]
then
```

```
    echo "failed to delete list:$listname, response code:$responsecode"
    if [ "$responsecode" == "405" ]
    then
        echo "The list must be removed from SEP Policy before it can be
deleted.  Note that the Listings are already deleted."
    fi
    exit 5
fi
echo "List $listname Deleted"
rm -f .$listname.*
```

Appendix B: Querying System Events via REST API

System events can be queried via the REST API from a command line on the Mediation Server. You can query by count or by date and time range.

To query system events using REST

1. Log in to the Mediation Server via a command-line interface.
2. Log in via the webserver IP to get a token:

```
curl -k -f -w HTTPResponseCode%{http_code} -X PUT -H "Content-Type:
application/json" -d '{"userId":"<username>", "password":"<password>"}'
https://<webserver_IP>:8443/mgmt/rest/security/login
```

3. Execute one of the following queries, using the token you received in response the above command.

- **By count:**

```
curl -k -f -X GET -H "Access-Token: <token_from_login>"
https://172.20.25.84:8443/mgmt/rest/sysevents?count=2
```

Example:

```
curl -k -f -X GET -H "Access-Token:
eyJhbGciOiJIUzI1NiJ9.eyJleHAiOjE0MzAzNDI0MzIsInBlcm1zIjp7InJlYWx0aW1lIjp0c
nVlLCJhbmFseXRpY3MiOnRydWUsImNvbmZpZyI6dHJ1ZSwicG9saWN5Ijp0cnVlfSwiaXNzIjo
iUG9saWN5R1VSVSIsIm9yaWciOjE0NjQxNTgxMzcsImlhdCI6MTQzMDI1NjAzMiwidXNlciI6I
mFkbWluIn0.mzHsafiPvy2kJoYxWb8sRIsEf6Td2ffBF611T1GhU8A"}
https://172.20.25.84:8443/mgmt/rest/sysevents?count=30
```

- **By date/time:**

```
curl -k -f -X GET -H "Access-Token: <token_from_login>"
https://172.20.25.84/mgmt/rest/sysevents?fromDate=1430246820000\&toDate=14
30246880000
```

The date/time range values are expressed in epoch time. A web-based epoch time converter is available at the following link: http://www.epochconverter.com/

Example:

```
curl -k -f -X GET -H "Access-Token:
eyJhbGciOiJIUzI1NiJ9.eyJleHAiOjE0MzAzNDI0MzIsInBlcm1zIjp7InJlYWx0aW1lIjp0c
nVlLCJhbmFseXRpY3MiOnRydWUsImNvbmZpZyI6dHJ1ZSwicG9saWN5Ijp0cnVlfSwiaXNzIjo
iUG9saWN5R1VSVSIsIm9yaWciOjE0NjQxNTgxMzcsImlhdCI6MTQzMDI1NjAzMiwidXNlciI6I
mFkbWluIn0.mzHsafiPvy2kJoYxWb8sRIsEf6Td2ffBF611T1GhU8A"
https://172.20.25.84/mgmt/rest/sysevents?fromDate=1430246820000\&toDate=14
30246880000
```

Appendix C: Understanding Regular Expressions in Rules

Overview

When the SBC receives a new SIP INVITE request (i.e., call) it extracts the source (i.e., caller) and destination (i.e., callee) information and formats an ENUM Request, which it transmits to its configured PolicyGuru ENUM Server. The PolicyGuru ENUM Server decides whether to ALLOW, TERMINATE, or REDIRECT the call represented by the ENUM Request. The decision is in the form of a POSIX-style regular expression which is returned to the SBC in the ENUM Reply. See “Substitution Expression Grammar” on page 92 for syntax details. The SBC applies the regular expression to the SIP INVITE’s Request-URI. The output of the application of the regular expression to the Request-URI is the call’s new Request-URI. The regular expression supplied for an ALLOW decision should result in an unaltered Request-URI (i.e., route the call to its original destination). The application of the regular expression for the TERMINATE or REDIRECT decision should produce a different destination.

The ENUM Server decides how to direct the call by applying the installed SEP Policy Rules to the data supplied in the ENUM Request. If the call does not match any SEP Policy Rules, the ENUM Server defaults to an ALLOW call action. In this default case, the regular expression is a matter of configuration. The same configured default regular expression is returned for all calls that default to ALLOW. Otherwise, the regular expression is supplied in the SEP Policy Rule that best matches the call—which could specify an ALLOW, REDIRECT, or TERMINATE decision.

An ENUM Reply includes an **Answer** section when the ENUM Request has been processed successfully. An **Answer** section is not included in the ENUM Reply that reports a Format Error or a Server Error. For a successful ENUM Request, the **Answer** section of the ENUM Reply contains exactly one **ANSWER-RR**. The content of the **ANSWER-RR** includes the POSIX-style regular expression.

The regular expression is a delimited string in two parts: The first part is the regular expression itself, referred to as the “ere” part. The second part is the replacement string. The purpose of the first part is to perform pattern matching on the input string. When a match occurs, one or more characters in the replacement string are substituted for characters in the input string to produce an output string. Together, the two parts are collectively referred to as the "regular expression" field.

The "regular expression" field begins and ends with an exclamation point character "!" and the two parts of the "regular expression" are also delimited by an exclamation point.

Two "regular expression" examples follow:

```
!a^!guaranteed no replacement!
```

This is a "regular expression" field that could be returned to an SBC when SEP Policy dictates that the call be permitted to flow unaltered to its original destination (an ALLOW decision). The regular expression **a^** guarantees no matches when applied to a Request-URI. No matches means that nothing in the replacement string (i.e. 'guaranteed no replacement') is substituted for characters in the Request-URI. When a regular expression matches no characters in the input string, the output string is identical to the input string. A replacement string is still required, however, so the text “guaranteed no replacement” is simply a commentary or place holder in this instance.

```
!^.*$!sip:8888888@10.1.50.18!
```

In this "regular expression", the ere part **^.*$** means to replace the input string entirely with the provided replacement string (i.e., sip:8888888@10.1.50.18). This is an example of a regex the PolicyGuru ENUM Server might provide for a TERMINATE or REDIRECT decision.

IMPORTANT: See “Important Information About Regex in the **Guided Rule Editor**” on page 47 for information on required special character replacement when using Regex in SEP Rules.

IMPORTANT: The **$** character must be escaped (i.e., preceded by a \ character) when entering the regular expression into the ENUM Server's default config script (**createDefaultConfig.sh**); otherwise, the ***$** characters get dropped from the regular expression as the script creates the sqlite3 configuration database where it stores local configuration data.

Note: The "regular expression" field in an **ANSWER-RR** is followed by a replacement field. The "regular expression" field and the replacement field are mutually exclusive. If the "regular expression" field is populated, the replacement field is empty. The PolicyGuru Solution uses the regex field and leaves the replacement field empty.

Substitution Expression Grammar

The content of the regex field is a substitution expression. While various standards for regular expression syntax exist, the regular expression in the ENUM reply must be a "POSIX Extended Regular Expression" (see RFC 2915, Section 3). True sed(1) and Perl style substitution expressions are not appropriate for use in this application for a variety of reasons stemming from internationalization requirements and backref limitations; therefore; the contents of the regex field MUST follow the grammar below:

```
subst_expr   = delim-char  ere  delim-char  repl  delim-char  *flags

delim-char   = "/" / "!" / ... <Any non-digit or non-flag character

               other than backslash '\'. All occurances of a delim_char

               in a subst_expr must be the same character.>

ere          = POSIX Extended Regular Expression

repl         = 1 * ( OCTET /  backref )

backref      = "\" 1POS_DIGIT

flags        = "i"

POS_DIGIT    = %x31-39                 ; 0 is not an allowed backref
```

The definition of a POSIX Extended Regular Expression can be found in [8], section 2.8.4 of the following standard::

[8] IEEE, "IEEE Standard for Information Technology - Portable Operating System Interface (POSIX) - Part 2: Shell and Utilities (Vol. 1)", IEEE Std 1003.2-1992, January 1993.

The link below is a website with a regular expression tester that permits a checkbox to be set to assure the ere part complies with POSIX ERE (egrep) syntax and leftmost-longest match semantics. If you test a regular expression that has an ere part that does not comply with POSIX, an error is flagged.

http://www.regexplanet.com/advanced/golang/index.html