



# NetWitness Log Decoder

Susam Pal

RSA NetWitness



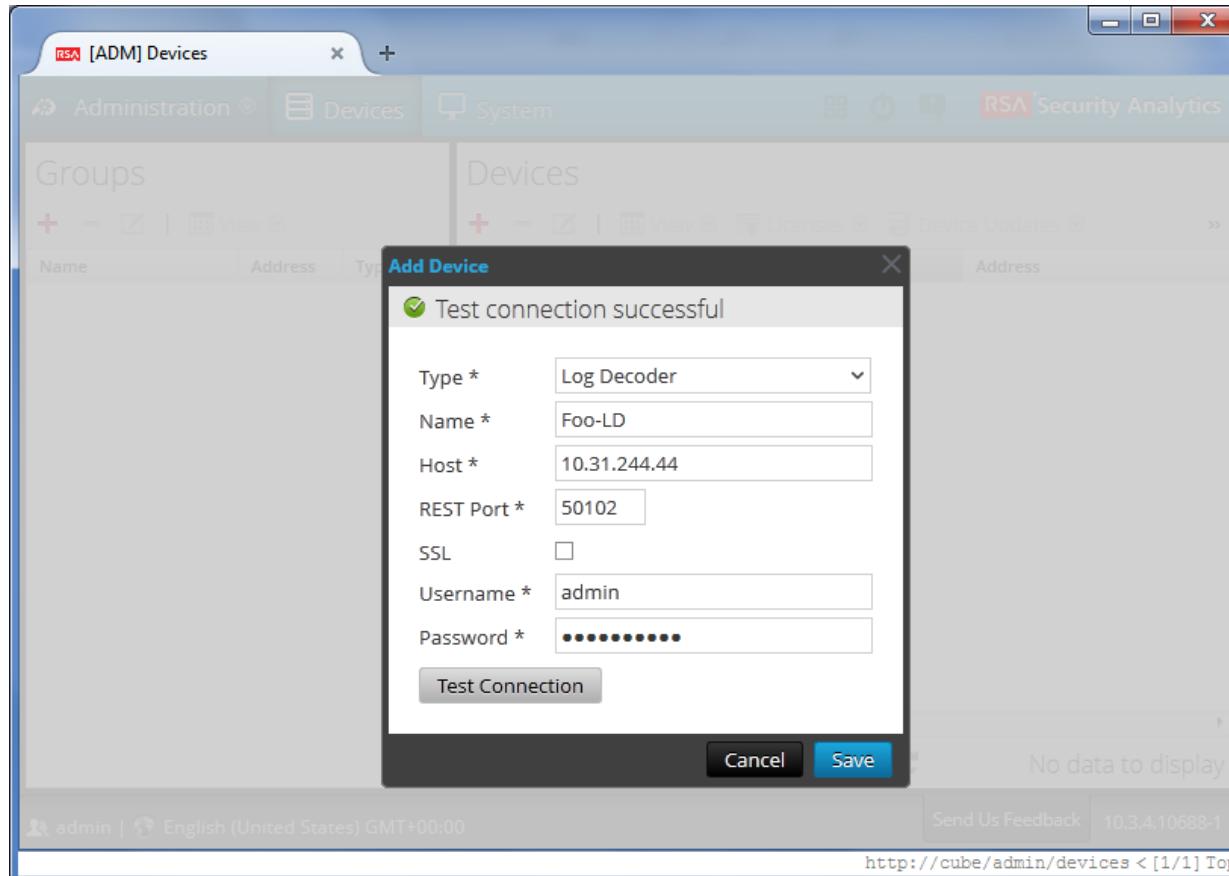


# Basic Configuration

## NetWitness Log Decoder



# Basic Configuration



Add Log Decoder (Administration → Devices)



# Basic Configuration

The screenshot shows the RSA Security Analytics configuration interface for device 'Foo-LD'. The top navigation bar includes 'Administration', 'Devices' (selected), and 'System'. The sub-navigation bar under 'Devices' shows 'Change Device' (selected), 'Foo-LD', and 'Config'. The main content area has three tabs: 'General' (selected), 'Files', 'App Rules', 'Correlation Rules', 'Feeds', 'Parsers', and 'Appliance Service'. The 'General' tab displays two sections: 'System Configuration' and 'Log Decoder Configuration'. 'System Configuration' contains a table with a single row: 'Compression' with value '0'. 'Log Decoder Configuration' contains sections for 'Adapter' (with 'Capture Interface Selected') and 'Cache' (with 'Cache Directory' set to '/var/netwitness/logdecoder/c...'). To the right, there are two tables: 'Parsers Configuration' and 'Device Parsers Config...'. 'Parsers Configuration' lists 'ALERTS' (checked), 'FeedParser' (checked), and 'GeoIP' (unchecked). 'Device Parsers Config...' lists 'accurev' (checked), 'actiancevantage' (checked), and 'actividentity' (checked). A large 'Apply' button is at the bottom right. The footer includes 'admin | English (United States) GMT+00:00', 'Send Us Feedback', '10.3.4.10688-1', and the URL 'http://cube/admin/devices/1/config < [1/1] All'.

Default Configuration (LD → Config → General)

# Basic Configuration

The screenshot shows the RSA Security Analytics configuration interface for a device named "Foo-LD". The main window title is "RSA [ADM] 'Foo-LD' config". The navigation bar includes tabs for Administration, Devices, System, and Security Analytics. The current view is under the "Devices" tab, specifically the "Config" sub-tab for "Change Device | Foo-LD". The configuration pages are organized into sections: "System Configuration", "Log Decoder Configuration", and "Parsers Configuration".

In the "Log Decoder Configuration" section, there is a table with columns "Name" and "Config Value". It contains two entries: "Capture Interface Selected" and "Cache Directory". A red circle highlights the "Capture Interface Selected" row, which has a value of "Selected".

On the right side of the interface, there are two configuration panels: "Parsers Configuration" and "Device Parsers Config...". Both panels show tables with "Name" and "Config Value" columns. In the "Parsers Configuration" panel, the "ALERTS" parser is selected. In the "Device Parsers Config..." panel, several parsers are listed, all with checked checkboxes.

At the bottom of the interface, there is an "Apply" button and a status bar with the user "admin", language "English (United States) GMT+00:00", a feedback link "Send Us Feedback", and the IP address "10.3.4.10688-1". The URL "http://cube/admin/devices/1/config < [1/1] All" is also visible.

Capture Interface must be set



# Basic Configuration

The screenshot shows the RSA Security Analytics configuration interface for a device named "Foo-LD". The main window has tabs for General, Files, App Rules, Correlation Rules, Feeds, Parsers, and Appliance Service. The "General" tab is selected. On the left, there's a "System Configuration" table with a single entry for "Compression" set to "0". Below it is a "Log Decoder Configuration" section with two expandable categories: "Adapter" and "Cache". Under "Adapter", the "Capture Interface Selected" dropdown is set to "log\_events,Log Events", with a hand cursor pointing at it. Under "Cache", the "Cache Directory" is set to "/var/netwitness/logdecoder/c...". On the right, there are two tables: "Parsers Configuration" and "Device Parsers Config...". The "Parsers Configuration" table lists "ALERTS", "FeedParser", and "GeoIP" with checkboxes. The "Device Parsers Config..." table lists "accurev", "actiancevantage", and "actividentity" with checkboxes. At the bottom right is a blue "Apply" button. The footer includes the user "admin", language "English (United States) GMT+00:00", a feedback link "Send Us Feedback", the IP "10.3.4.10688-1", and a URL "http://cube/admin/devices/1/config < [1/1] All".

Set Capture Interface



# Basic Configuration

The screenshot shows the RSA Security Analytics configuration interface for device 'Foo-LD'. The 'Log Decoder Configuration' section is highlighted with a green oval around the 'Capture Interface Selected' field, which contains the value 'log\_events,Log Events'. The interface includes tabs for General, Files, App Rules, Correlation Rules, Feeds, Parsers, and Appliance Service. It also displays System Configuration and Parsers Configuration sections.

Name	Config Value
Compression	0

Name	Config Value
Adapter	log_events,Log Events
Cache	/var/netwitness/logdecoder/c...

Name	Config Value
ALERTS	<input checked="" type="checkbox"/>
FeedParser	<input checked="" type="checkbox"/>
GeoIP	<input type="checkbox"/>

Name	Config Value
accurev	<input checked="" type="checkbox"/>
actiancevantage	<input checked="" type="checkbox"/>
actividentity	<input checked="" type="checkbox"/>

Apply

admin | English (United States) GMT+00:00 Send Us Feedback 10.3.4.10688-1  
http://cube/admin/devices/1/config < [1/1] All

Set Capture Interface



# Basic Configuration

The screenshot shows the RSA Security Analytics web interface. The top navigation bar includes tabs for Administration, Devices, and System, with the System tab selected. A sub-menu under System shows 'Change Device' and 'Foo-LD'. Below the navigation is a toolbar with buttons for Upload Log File, Start Capture (which has a mouse cursor hovering over it), Reset Log Stats, Appliance Tasks, Shutdown Service, and Shutdown Appliance.

**Log Decoder Service Information**

Name	cube (Log Decoder)
Version	10.3.4.2634 (Rev edbc98716500)
Memory Usage	2617 MB (8.12% of 32239 MB)
CPU	0%
Running Since	2014-Oct-03 12:52:37
Uptime	39 minutes 37 seconds
Current Time	2014-Oct-03 13:32:14

**Appliance Service Information**

Name	cube (Appliance)
Version	10.3.4.2634 (Rev edbc98716500)
Memory Usage	31204 KB (0.09% of 32239 MB)
CPU	0%
Running Since	2014-Oct-03 11:23:01
Uptime	2 hours 9 minutes 12 seconds
Current Time	2014-Oct-03 13:32:13

**Log Decoder User Information**

Name	admin
Groups	Administrators

**Appliance User Information**

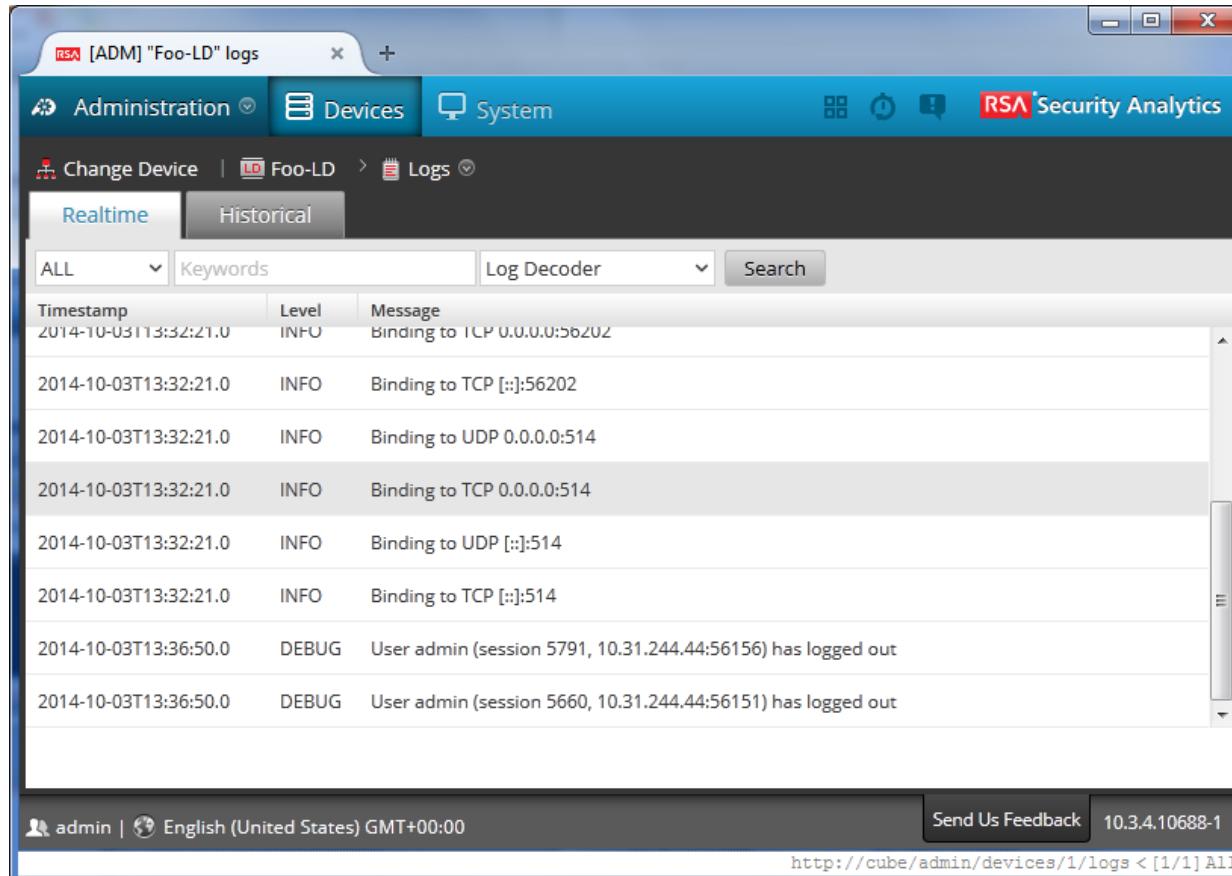
Name	admin
Groups	Administrators

Footer information includes: admin | English (United States) GMT+00:00, Send Us Feedback, 10.3.4.10688-1, and the URL http://cube/admin/devices/1/info < [1/1] All.

Start Capture (LD → System)



# Basic Configuration



The screenshot shows the RSA Security Analytics web interface. The title bar reads "RSA [ADM] 'Foo-LD' logs". The top navigation bar includes "Administration", "Devices" (selected), and "System" tabs, along with a "Security Analytics" logo. Below the navigation is a breadcrumb trail: "Change Device" > "Foo-LD" > "Logs". A tab bar at the top of the main content area has "Realtime" (selected) and "Historical" tabs. A search bar includes dropdowns for "Keywords" and "Log Decoder", and a "Search" button. The main pane displays a table of log entries:

Timestamp	Level	Message
2014-10-03T13:32:21.0	INFO	Binding to TCP 0.0.0.0:56202
2014-10-03T13:32:21.0	INFO	Binding to TCP [::]:56202
2014-10-03T13:32:21.0	INFO	Binding to UDP 0.0.0.0:514
2014-10-03T13:32:21.0	INFO	Binding to TCP 0.0.0.0:514
2014-10-03T13:32:21.0	INFO	Binding to UDP [::]:514
2014-10-03T13:32:21.0	INFO	Binding to TCP [::]:514
2014-10-03T13:36:50.0	DEBUG	User admin (session 5791, 10.31.244.44:56156) has logged out
2014-10-03T13:36:50.0	DEBUG	User admin (session 5660, 10.31.244.44:56151) has logged out

The bottom navigation bar includes "Send Us Feedback" and the IP address "10.3.4.10688-1". The URL "http://cube/admin/devices/1/logs < [1/1] All" is also visible.

Logs you see on successful capture start (LD → Logs → Realtime/Historical)



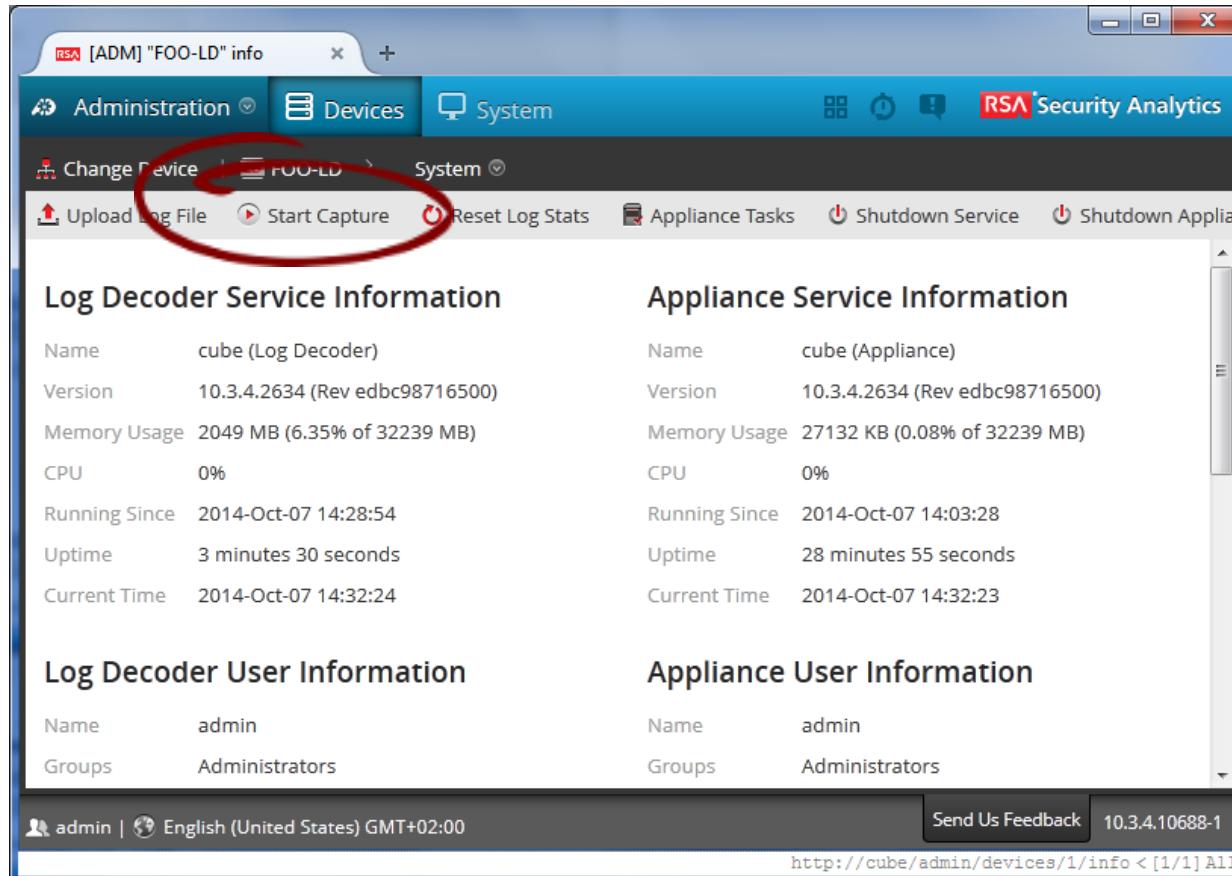


# Capture Failure

## NetWitness Log Decoder



# Capture Failure: Disk Space Issue



The screenshot shows the RSA Security Analytics web interface. At the top, there's a header with tabs for 'Administration', 'Devices' (which is selected), and 'System'. Below the header, a sub-navigation bar shows 'FOO-LD' is selected under 'Change Device'. The main content area is divided into two columns: 'Log Decoder Service Information' and 'Appliance Service Information'. Both sections provide detailed status information. At the bottom, there are sections for 'Log Decoder User Information' and 'Appliance User Information', both listing a single user named 'admin'. The footer includes a user info bar ('admin | English (United States) GMT+02:00'), a feedback link ('Send Us Feedback'), and a session ID ('10.3.4.10688-1'). A URL at the very bottom is 'http://cube/admin/devices/1/info < [1/1] All'.

Symptom: 'Start Capture' button never turns into 'Stop Capture' button



# Capture Failure: Disk Space Issue

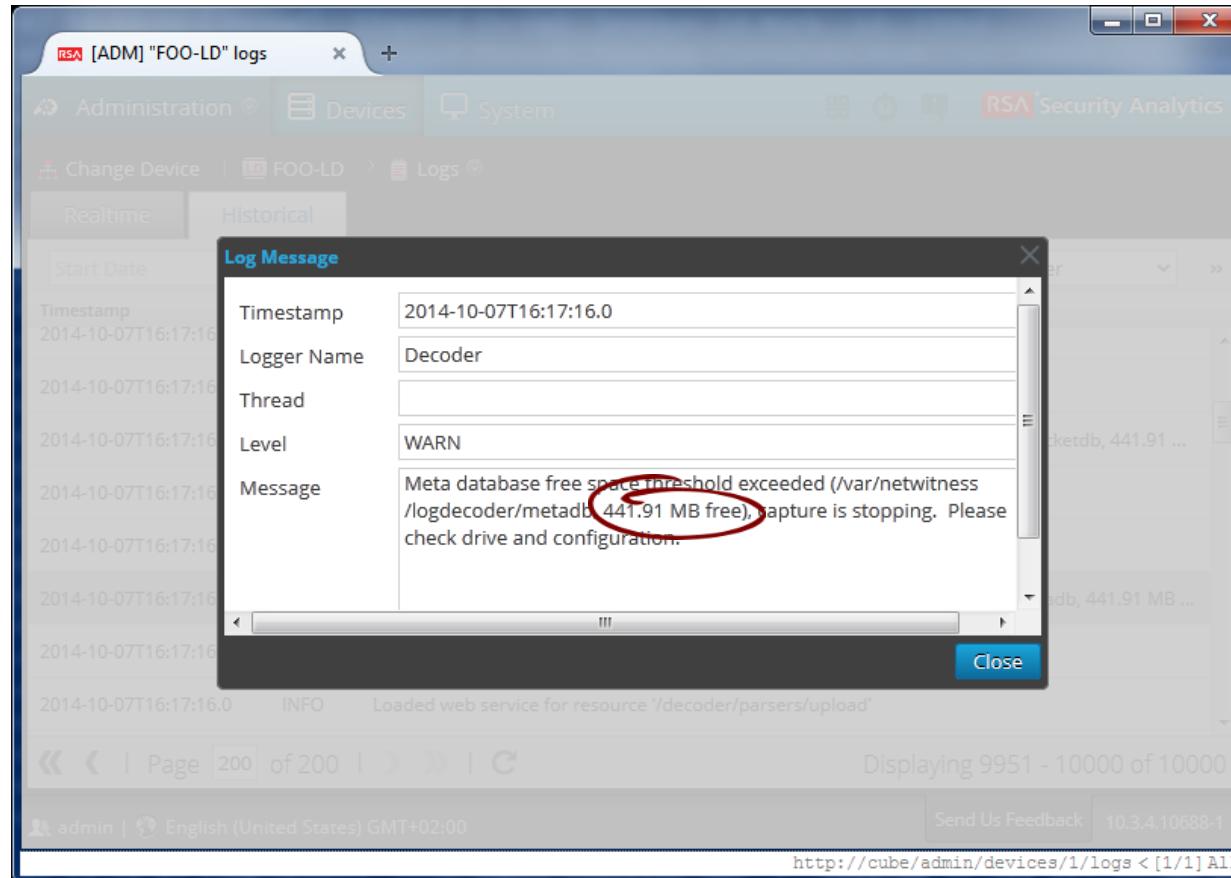
The screenshot shows the RSA Security Analytics interface for device FOO-LD. The 'Logs' tab is selected, and the 'Historical' view is active. A red circle highlights a warning log entry from October 7, 2014, at 16:17:16.0, which states: "Meta database free space threshold exceeded (/var/netwitness/logdecoder/metadb, 441.91 MB ...)".

Timestamp	Level	Message
2014-10-07T16:17:16.0	INFO	Binding to UDP 0.0.0.0:514
2014-10-07T16:17:16.0	INFO	Loaded web service for resource '/decoder/import'
2014-10-07T16:17:16.0	WARN	Packet database free space threshold exceeded (/var/netwitness/logdecoder/packetdb, 441.91 ...)
2014-10-07T16:17:16.0	INFO	Binding to TCP 0.0.0.0:514
2014-10-07T16:17:16.0	INFO	Loaded web service for resource '/decoder/agg'
2014-10-07T16:17:16.0	WARN	Meta database free space threshold exceeded (/var/netwitness/logdecoder/metadb, 441.91 MB ...)
2014-10-07T16:17:16.0	INFO	Binding to UDP [::]:514
2014-10-07T16:17:16.0	INFO	Loaded web service for resource '/decoder/parsers/upload'

Clues in the logs (LD → Logs → Historical)

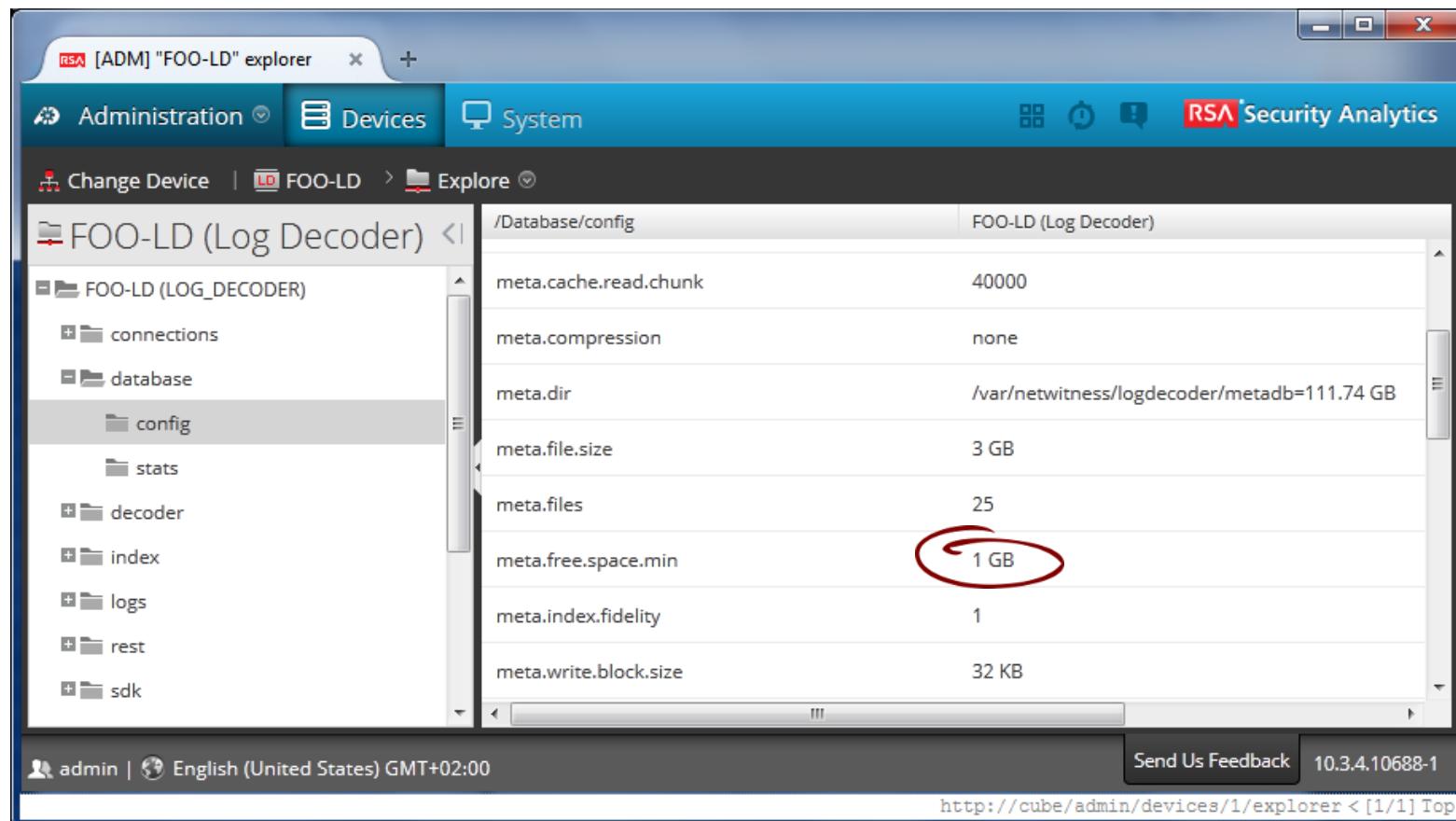


# Capture Failure: Disk Space Issue



Clues in the logs: metadb is almost full

# Capture Failure: Disk Space Issue



The screenshot shows the RSA Security Analytics interface for device 'FOO-LD'. The left sidebar shows a tree view of the device structure, with 'database/config' selected. The main pane displays configuration parameters:

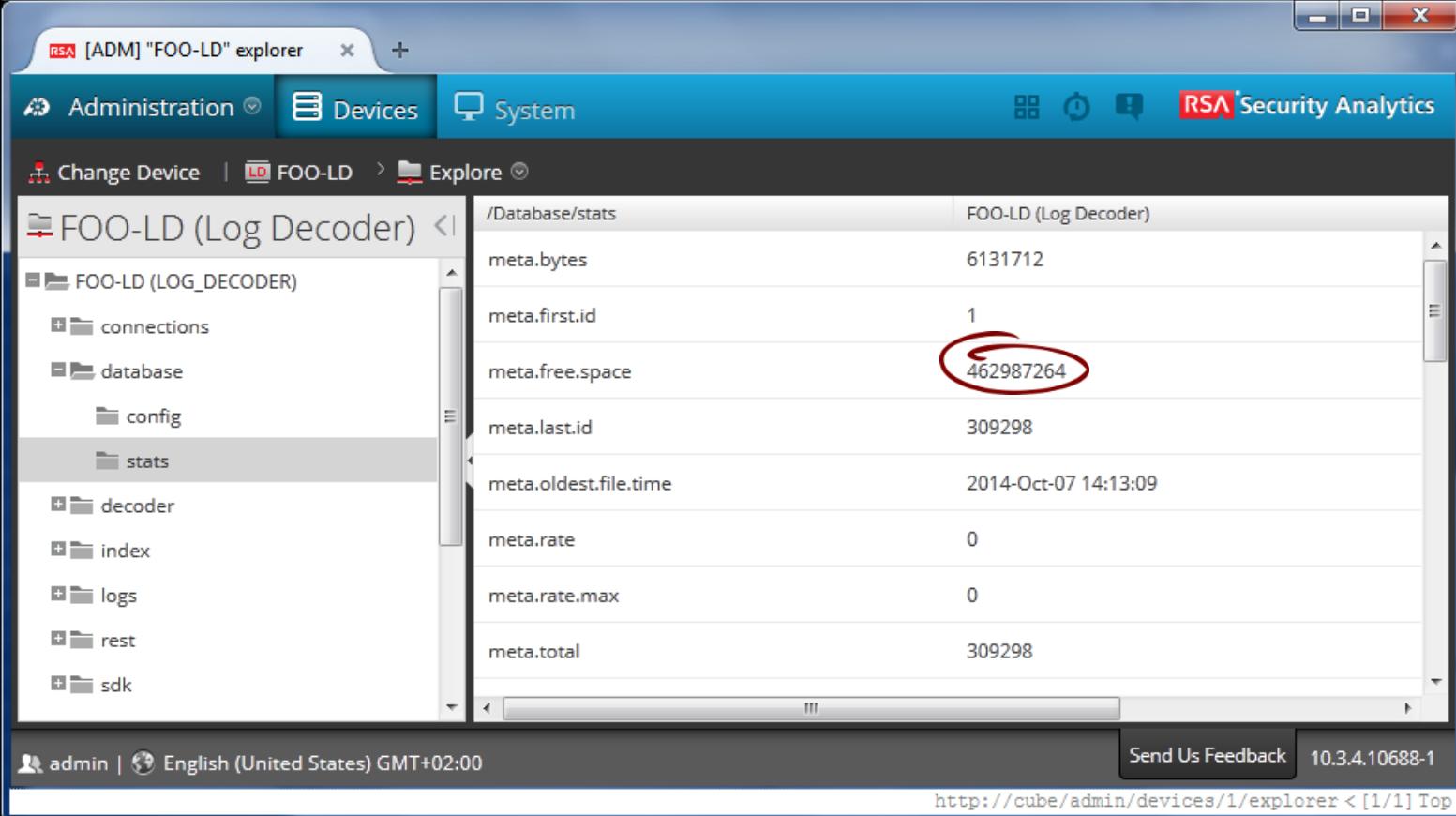
Parameter	Value
meta.cache.read.chunk	40000
meta.compression	none
meta.dir	/var/netwitness/logdecoder/metadb=111.74 GB
meta.file.size	3 GB
meta.files	25
meta.free.space.min	1 GB
meta.index.fidelity	1
meta.write.block.size	32 KB

A red circle highlights the value '1 GB' for 'meta.free.space.min'. The bottom status bar shows the user is 'admin' and the time is 'English (United States) GMT+02:00'. The URL in the status bar is <http://cube/admin/devices/1/explorer>.

Database configuration (LD → Explore → /decoder/database/config)



# Capture Failure: Disk Space Issue



The screenshot shows the RSA Security Analytics interface with the title bar "rsa [ADM] "FOO-LD" explorer". The navigation bar includes "Administration", "Devices" (selected), and "System". Below the navigation is a breadcrumb trail: "Change Device" → "FOO-LD" → "Explore". The left sidebar lists device components: "FOO-LD (LOG\_DECODER)" (connections, database, config, stats, decoder, index, logs, rest, sdk), "decoder", "index", "logs", "rest", and "sdk". The "stats" folder is currently selected. The main pane displays "Database/stats" for "FOO-LD (Log Decoder)". A red circle highlights the "meta.free.space" value, which is 462987264. Other statistics include: meta.bytes (6131712), meta.first.id (1), meta.last.id (309298), meta.oldest.file.time (2014-Oct-07 14:13:09), meta.rate (0), meta.rate.max (0), and meta.total (309298). At the bottom, the status bar shows "admin | English (United States) GMT+02:00", "Send Us Feedback", "10.3.4.10688-1", and the URL "http://cube/admin/devices/1/explorer < [1/1] Top".

/Database/stats	FOO-LD (Log Decoder)
meta.bytes	6131712
meta.first.id	1
meta.free.space	462987264
meta.last.id	309298
meta.oldest.file.time	2014-Oct-07 14:13:09
meta.rate	0
meta.rate.max	0
meta.total	309298

Database statistics (LD → Explore → /decoder/database/stats)





# Capture Failure: Disk Space Issue

These are the parameters in Log Decoder → Explore → /decoder/database/config that specify the DB locations and allocated maximum size.

meta.dir

packet.dir

session.dir

By default, they point to the following locations, respectively.

/var/netwitness/logdecoder/metadb

/var/netwitness/logdecoder/packetdb

/var/netwitness/logdecoder/sessiondb

By default, approximately 95% of the total disk space is allocated for each configuration. For example, if the metadb is on a partition of size 118 GB, then by default,

meta.dir = /var/netwitness/logdecoder/metadb=111.74 GB





# Capture Failure: Disk Space Issue

These are the parameters in Log Decoder → Explore → /decoder/database/config that specify the minimum free space required for capture to start.

meta.free.space.min

packet.free.space.min

session.free.space.min

By default, the value of each is set to  
(total disk space / 115) or 2 TB, whichever is less.

Example:

If metadb is on a partition of size 118 GB, then by default,  
meta.free.space.min = 1 GB



# Capture Failure: Disk Space Issue



The most common reason for metadb becoming full is creation of core dump files.

By default, Log Decoder's current working directory is </var/netwitness/logdecoder/metadb>. As a result, any Log Decoder core dumps are written to this location.

A single core dump may be as large as 30 GB in size. A couple of core dumps may fill up metadb, if metadb's partition is small.

Log Decoder's current working directory is defined with the [chdir](#) command in [/etc/init/nwlogdecoder.conf](#).

```
chdir /var/netwitness/logdecoder/metadb
```

If metadb is becoming full due to core dump files, consider changing this command to point to another location, say [/var/netwitness/logdecoder/packetdb](#) and [restart nwlogdecoder](#). Use the command `df -h` to see disk space usage.



# Capture Failure (continued ...)

## NetWitness Log Decoder

# Capture Failure: Parser Error

The screenshot shows the RSA Security Analytics interface for device "Foo-LD". The "System" tab is selected. In the top navigation bar, there is a red circle around the "Start Capture" button. Below it, under "Log Decoder Service Information", the "Name" field is listed as "cube (Log Decoder) Initialization Error", which is also circled in red. The "Appliance Service Information" section shows standard system details.

Name	cube (Log Decoder) Initialization Error
Version	10.3.4.2634 (Rev edbc98716500)
Memory Usage	98644 KB (0.30% of 32239 MB)
CPU	0%
Running Since	2014-Oct-02 09:09:55
Uptime	2 hours 5 minutes 35 seconds
Current Time	2014-Oct-02 11:15:30

Name	cube (Appliance)
Version	10.3.4.2634 (Rev edbc98716500)
Memory Usage	37124 KB (0.11% of 32239 MB)
CPU	0%
Running Since	2014-Sep-30 13:35:53
Uptime	1 day 21 hours 39 minutes 36 seconds
Current Time	2014-Oct-02 11:15:29

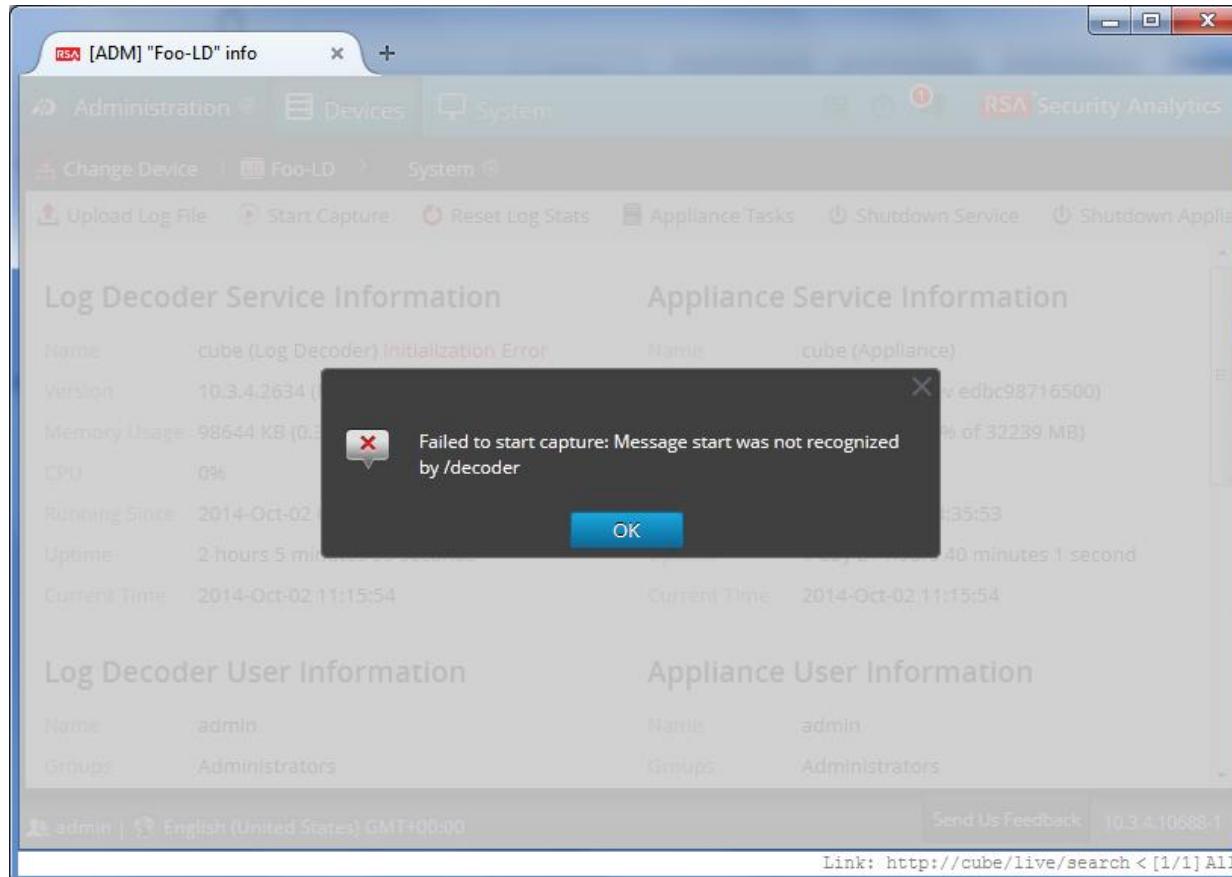
Name	admin
Groups	Administrators

Name	admin
Groups	Administrators

Bottom status bar: admin | English (United States) GMT+00:00 | Send Us Feedback | 10.3.4.10688-1 | Link: http://cube/live/search < [1/1] All

Symptom: 'Initialization Error' (LD → System)

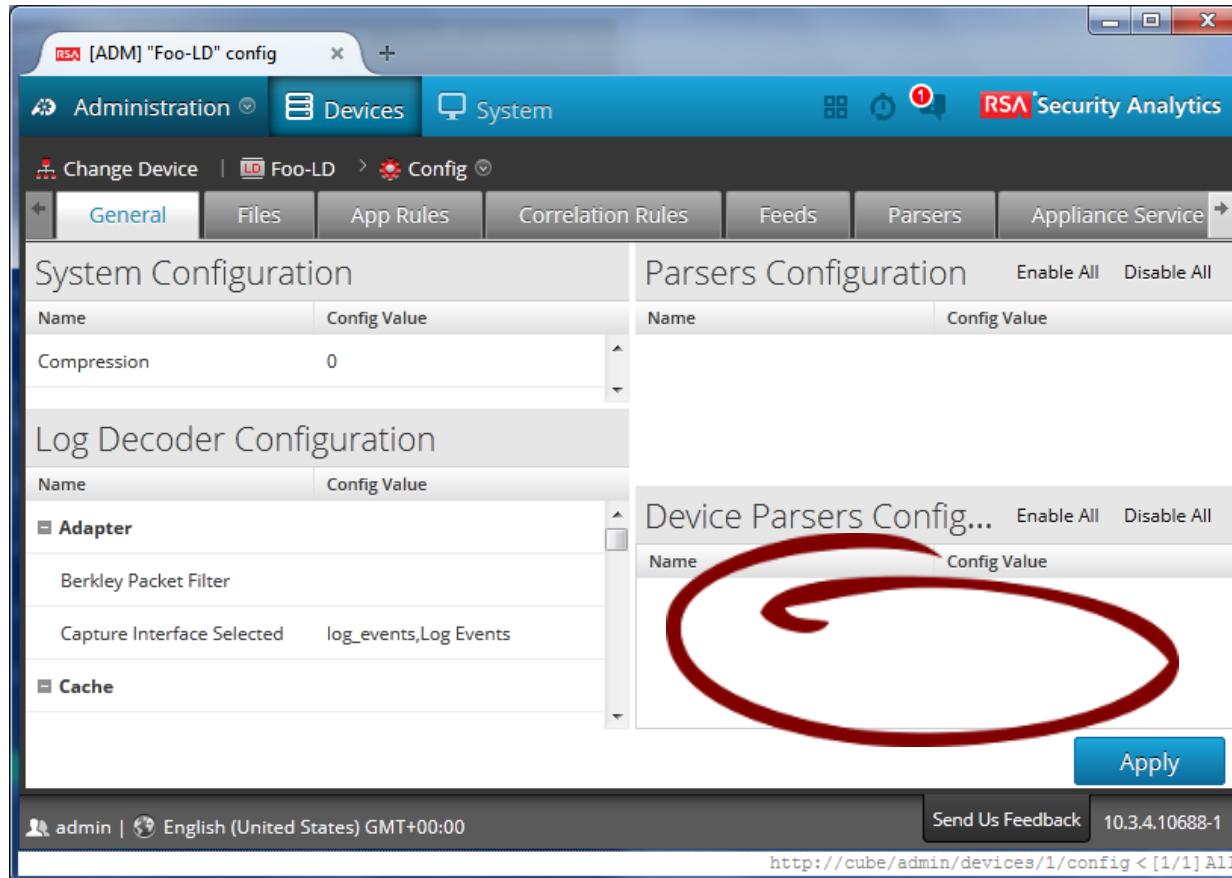
# Capture Failure: Parser Error



Symptom: 'Failed to start capture' message



# Capture Failure: Parser Error



Symptom: No parsers listed

# Capture Failure: Parser Error

The screenshot shows a log viewer interface for the RSA Security Analytics platform. The title bar reads "RSA [ADM] 'Foo-LD' logs". The top navigation bar includes "Administration", "Devices", "System", and the "RSA Security Analytics" logo. Below the navigation is a breadcrumb trail: "Change Device" > "Foo-LD" > "Logs". The log viewer has two tabs: "Realtime" (selected) and "Historical". It features filters for "Start Date", "End Date", "Level" (set to ALL), "Keywords", and "Log Decoder". The main area displays a table of log entries:

Timestamp	Level	Message
2014-10-02T09:09:56.0	INFO	File celarra content loaded
2014-10-02T09:09:57.0	INFO	File checkpointfw1 content loaded
2014-10-02T09:09:58.0	INFO	File ciscoasa content loaded
2014-10-02T09:09:58.0	INFO	File ciscomeraki content loaded
2014-10-02T09:09:58.0	INFO	File cisconcm content loaded
2014-10-02T09:09:58.0	WARN	Module logdecoder failed to load: ERROR: Invalid attribute name "%B %F %W %". Names cannot..
2014-10-02T09:09:58.0	WARN	Module logdecoder failed to load: Diagnostic information: Throw in function std::basic_string<ch..
2014-10-02T09:09:58.0	INFO	Found 2 files (548 KB) when loading /var/netwitness/logdecoder/statdb of max size 1 GB

A red circle highlights the second-to-last log entry (WARN level). A red bracket groups the last three log entries (WARN level) under the heading "Why did it fail?". The bottom of the screen shows pagination controls ("Page 200 of 200"), a timestamp ("Displaying 9951 - 10000 of 10000"), user information ("admin | English (United States) GMT+00:00"), and a feedback link ("Send Us Feedback"). The URL at the bottom is "http://cube/admin/devices/1/logs < [1/1] All".

**What failed?**  
The parser that was about to be loaded after 'cisconcm'.

**Why did it fail?**  
Something went wrong while reading "%B %F %W ...".

Clues in the logs (LD → Logs → Historical)

# Capture Failure: Parser Error

The `/etc/netwitness/ng/etc/devices` directory contains all the device parsers.

grep's `-A N` option allows us to display  $N$  number of lines after each match.

```
# ls /etc/netwitness/ng/envision/etc/devices | grep -A5 cisconcm
cisconcm ← We know this one could be loaded
cisconxos ← Did this one fail?
ciscopix
ciscorouter
ciscosecagent
ciscosecureacs
```

(The above command is run on Log Decoder.)

We know that the failure occurred after loading 'cisconcm'.

Did 'cisconxos' fail?





# Capture Failure: Parser Error

```
# ls /etc/netwitness/ng/envision/etc/devices | grep -A5 cisconcm  
cisconcm  
cisconxos  
ciscopix  
ciscorouter  
ciscosecagent  
ciscosecureacs
```

(The above command is run on Log Decoder.)

## Did 'cisconxos' fail?

We need to consider that disabled parsers are not loaded, so disabled parsers cannot fail.

## Is 'cisconxos' a disabled parser?

We need to check the REST resource </decoder/parsers/config/devices.disabled> to find out.



# Capture Failure: Parser Error

The screenshot shows the RSA [ADM] "Foo-LD" explorer interface. The navigation bar includes Administration, Devices, System, and the RSA Security Analytics logo. The main pane displays the configuration for "Foo-LD (Log Decoder)" under the path Change Device > Foo-LD > Explore. The left sidebar lists various sections like connections, database, decoder, config, devices, parsers, etc. The right pane shows a table of configuration parameters:

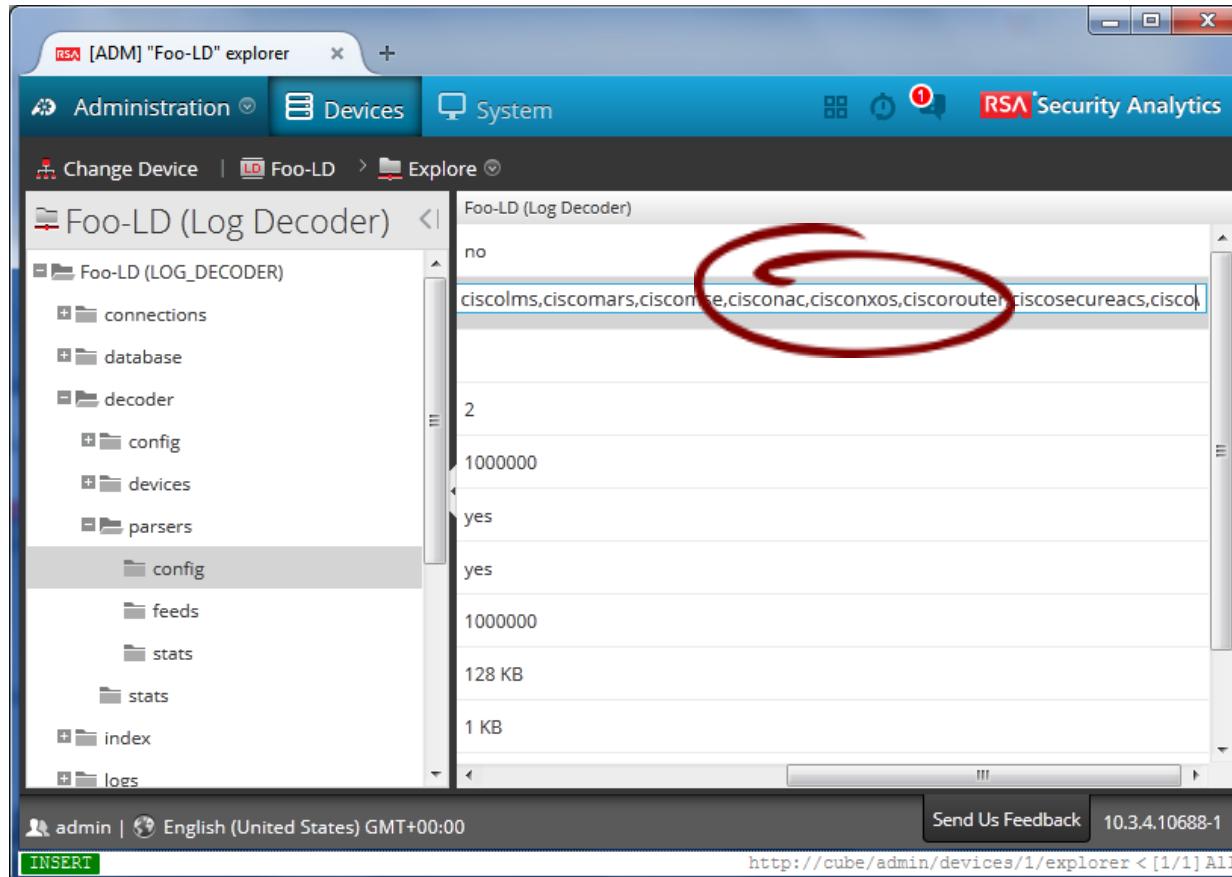
Path	Value
/Decoder/parsers/config	Foo-LD (Log Decoder)
detailed.stats	no
devices.disabled	airmagnet,apachetomcat,appsecd
feeds.disabled	
filename.meta	
flex.instruction.limit	1000000
legacy.content	yes
lua.enabled	yes
lua.instruction.limit	1000000
parse.bytes.max	128 KB
parse.bytes.min	1 KB

A tooltip for the "feeds.disabled" field states: "Disables the listed log devices. Change takes effect immediately." The bottom status bar shows the user is admin, the language is English (United States) GMT+00:00, and the IP address is 10.3.4.10688-1. The URL http://cube/admin/devices/1/explorer < [1/1] All is also visible.

Is 'cisconxos' a disabled parser?



# Capture Failure: Parser Error



'cisconxos' is disabled but 'ciscopix' isn't

# Capture Failure: Parser Error

A nifty shell command to check if a device parser is disabled.

These commands are run on Log Decoder.

```
# curl -su admin:netwitness http://localhost:50102/decoder/parsers/config/devices.disabled  
| tr , "\n" | grep cisconxos  
cisconxos  
#
```

(The above command is run on Log Decoder.)

‘cisconxos’ exists in `/decoder/parsers/config/devices.disabled`, so it is disabled.

```
# curl -su admin:netwitness http://localhost:50102/decoder/parsers/config/devices.disabled  
| tr , "\n" | grep ciscopix  
#
```

(The above command is run on Log Decoder.)

‘ciscopix’ does not exist in `/decoder/parsers/config/devices.disabled`, so it is enabled.

# Capture Failure: Parser Error

```
# ls /etc/netwitness/ng/envision/etc/devices | grep -A5 cisconcm
cisconcm ← This one could be loaded
cisconxos ← This one is disabled, so this cannot fail
ciscopix ← This one is enabled, so this must have failed
ciscorouter
ciscosecagent
ciscosecureacs
```

Did ‘**cisconxos**’ fail?

No, because ‘**cisconxos**’ is disabled.

Did ‘**ciscopix**’ fail?

Yes, because ‘**ciscopix**’ is enabled, and it is the first enabled parser that comes after ‘**cisconcm**’, in alphabetical order.

What went wrong in the ‘**ciscopix**’ parser?

We need to check the logs and the parser to find out.

# Capture Failure: Parser Error

The screenshot shows a log viewer interface for the RSA Security Analytics platform. The title bar reads "RSA [ADM] 'Foo-LD' logs". The navigation bar includes "Administration", "Devices", "System", and the "RSA Security Analytics" logo. Below the navigation is a breadcrumb trail: "Change Device" > "Foo-LD" > "Logs". The log viewer has two tabs: "Realtime" (selected) and "Historical". It features filters for "Start Date", "End Date", "Level" (set to ALL), "Keywords", and "Log Decoder". The main area displays a table of log entries:

Timestamp	Level	Message
2014-10-02T09:09:56.0	INFO	File celarra content loaded
2014-10-02T09:09:57.0	INFO	File checkpointfw1 content loaded
2014-10-02T09:09:58.0	INFO	File ciscoasa content loaded
2014-10-02T09:09:58.0	INFO	File ciscomeraki content loaded
2014-10-02T09:09:58.0	INFO	File cisconcm content loaded
2014-10-02T09:09:58.0	WARN	Module logdecoder failed to load: ERROR: Invalid attribute name "%B %F %W %". Names cannot..
2014-10-02T09:09:58.0	WARN	Module logdecoder failed to load: Diagnostic information: Throw in function std::basic_string<ch..
2014-10-02T09:09:58.0	INFO	Found 2 files (548 KB) when loading /var/netwitness/logdecoder/statdb of max size 1 GB

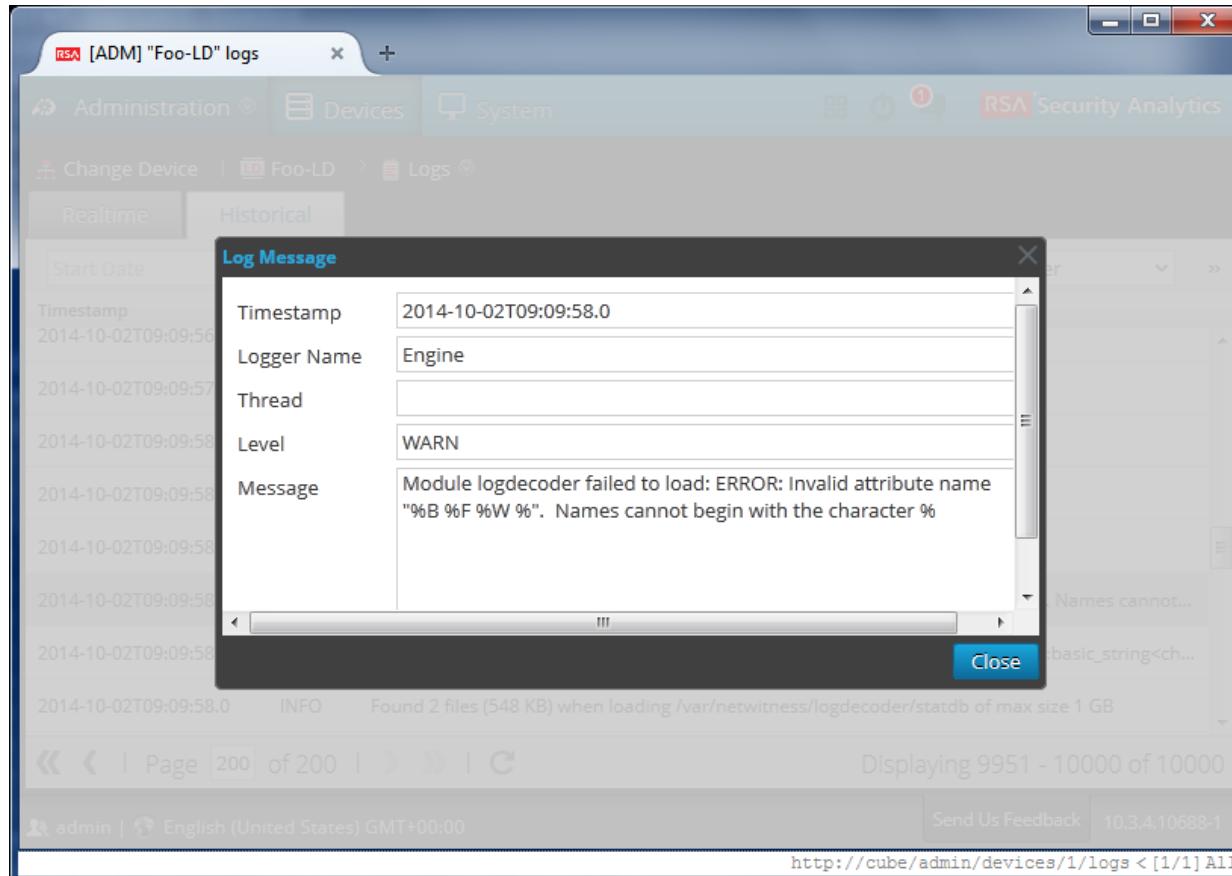
A red circle highlights the second-to-last log entry (WARN level). A red bracket groups the last two log entries (WARN level) with the message containing "%B %F %W %". The status bar at the bottom shows "Displaying 9951 - 10000 of 10000" and the URL "http://cube/admin/devices/1/logs < [1/1] All".

**What failed?**  
The parser that was about to be loaded after 'cisconcm'.

**Why did it fail?**  
Something went wrong while reading "%B %F %W ...".

Clues in the logs (LD → Logs → Historical)

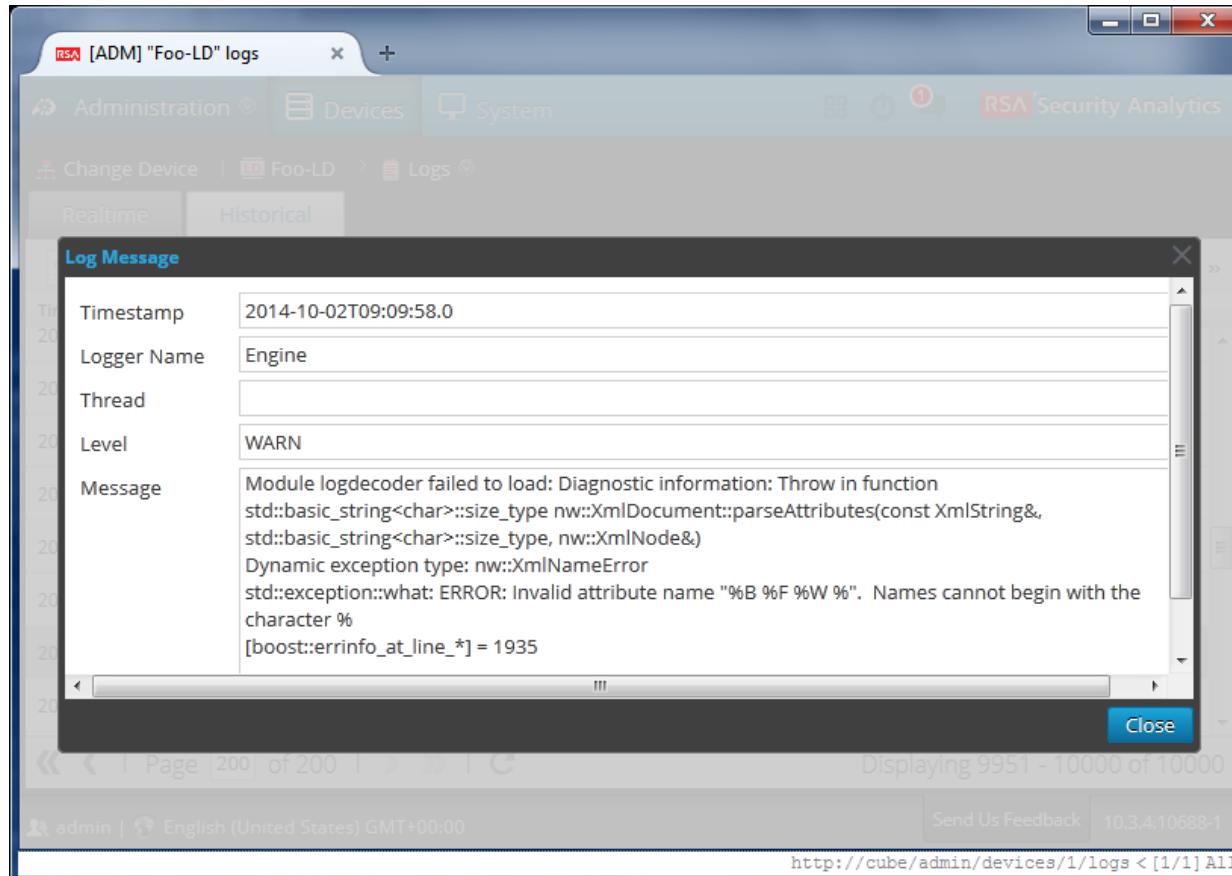
# Capture Failure: Parser Error



Clues in the logs: '%B %F %W' are time format specifiers used in the device parsers



# Capture Failure: Parser Error



Clues in the logs: XML related error. Device parsers are written in XML.



# Capture Failure: Parser Error

Syntax error in [/etc/netwitness/ng/envision/etc/devices/ciscopix/v20\\_ciscopixmsg.xml](#)

```
<MESSAGE
    level="1"
    parse="1"
    parsedefvalue="1"
    id1="101001"
    id2="101001"
    tableid="77"
    eventcategory="1604000000"
    content="&lt;@level:*HDR(level)&gt;&lt;@event_time:*EVNTTIME($HDR, "%B %F %W
%N:%U:%O",month,day,year,time)&gt;&lt;@msg:*PARMVAL($MSG)&gt;&lt;@fld61:*PARMVAL(context)&gt;(&
lt;context&gt;)&lt;event_description&gt;"'/>
```

Fix: Use single-quotes inside double-quoted attribute values

```
content="&lt;@level:*HDR(level)&gt;&lt;@event_time:*EVNTTIME($HDR, 'B F W
%N:%U:%O',month,day,year,time)&gt;&lt;@msg:*PARMVAL($MSG)&gt;&lt;@fld61:*PARMVAL(context)&gt;(&
lt;context&gt;)&lt;event_description&gt;"'/>
```



# Capture Failure: Parser Error

Syntax error in /etc/etw/etwmon.xml/device/etc/devices/import/v20/cisco-xml.xml

<MESSAGE

```
level="1"
par="1"
parsedefvalue="1"
id1="101001"
id2="101001"
tableid="77"
eventcategory="1604000000"
```

content="&lt;@level:\*HDR(level)&gt;&lt;@event\_time:\*EVNTTIME(\$HDR, "%B %F %W %N:%U:%O",month,day,year,time)&gt;&lt;@context:\*PARMVAL(\$G1000&gt;:old61:\*PARMVAL(context)&gt;(&lt;context&gt;)&lt;event\_desc:&gt;%&lt;/event\_desc&gt;)"&gt;

This was an example based on  
a true story about a custom  
device parser.

fix: Use single quotes for quoted attributes





# Log Ingestion

## NetWitness Log Decoder





# Log Ingestion

Feed some logs to Log Decoder.

In Bash, `/dev/<protocol>/<host>/<port>` is a special magic device that represents a local or remote host and port. Redirecting any output to it leads to establishing a connection with the `<host>:<port>` and sending the output as payload to it.

We'll use this feature to feed some logs to Log Decoder and demonstrate log ingestion concepts. We are not feeding logs from Log Collector to keep the demonstration simple and to help us type the logs by hand.

```
# echo "<1> %PIX-1-101001: (PRIORITY) Failover cable OK." > /dev/tcp/127.0.0.1/514
# echo "<1> %PIX-1-101001: (PRIORITY) Bad failover cable." > /dev/tcp/127.0.0.1/514
```



# Log Ingestion

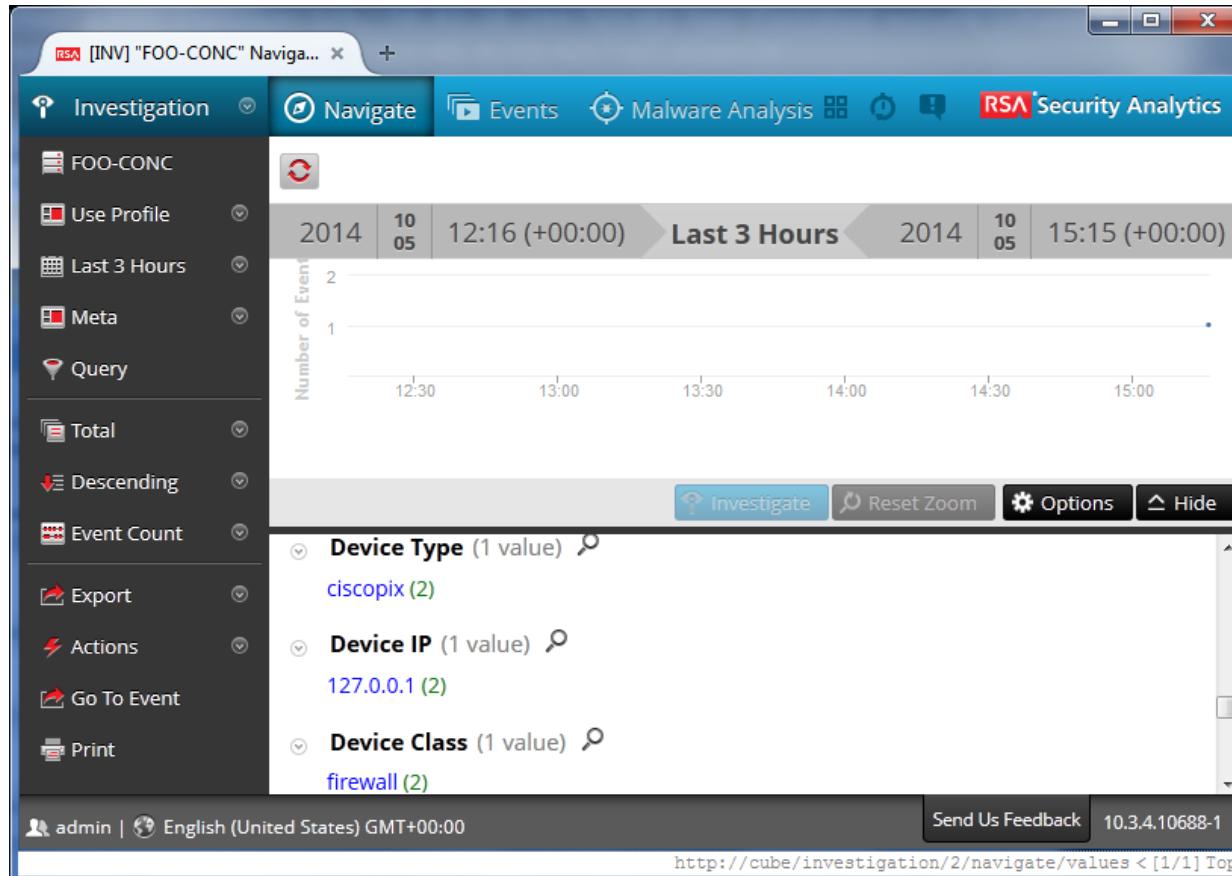
The screenshot shows the RSA Security Analytics interface with the title bar "RSA [INV] View Events". The top navigation bar includes "Investigation", "Events", "Malware Analysis", and "Security Analytics". Below the navigation is a search bar with the query "device.type = 'ciscopix'". The main content area displays two log entries:

Event Time	Event Type	Event Theme	Size	Details
2014-10-05T15:15:38	Log	System.Heartbeats	85 bytes	<ul style="list-style-type: none"><li>↳ sessionid : 1</li><li>↳ device.ip : 127.0.0.1</li><li>↳ medium : 32</li><li>↳ device.type : ciscopix</li><li>↳ device.class : Firewall</li><li>↳ header.id : 0001</li><li>↳ level : 1</li><li>↳ event.desc : Failover cable OK.</li><li>↳ msg.id : 101001</li></ul> <p><a href="#">+ Show Additional Meta</a> <a href="#">View Details</a></p> <ul style="list-style-type: none"><li>↳ sessionid : 2</li><li>↳ device.ip : 127.0.0.1</li><li>↳ medium : 32</li><li>↳ device.type : ciscopix</li></ul>

At the bottom, there are navigation controls (back, forward, search), a page size selector (25 items per page), and a status bar showing "2 events", "admin | English (United States) GMT+00:00", "Send Us Feedback", "10.3.4.10688-1", and the URL "http://cube/investigation/events < [1/1] All".

We can see the logs (Concentrator → Investigation → Events)

# Log Ingestion



Meta values and their counts (Concentrator → Investigation → Navigate)



# Log Ingestion

The screenshot shows the RSA [ADM] "FOO-LD" explorer interface. The left sidebar shows a tree view of device components under "FOO-LD (LOG\_DECODER)". The "stats" node is selected. The main pane displays a table of log decoder statistics:

/Decoder/stats	FOO-LD (Log Decoder)
capture.header.bytes	1811183756
capture.interface	Log Events
capture.kept	44191116
capture.packet.rate	21789
capture.packet.rate.max	36139
capture.payload.bytes	449809425
capture.rate	24
capture.rate.max	41
capture.received	44191116
capture.status	started

Annotations on the right side explain the data:

- Number of logs (points to capture.kept)
- Current EPS (points to capture.packet.rate)
- In megabits per second (points to capture.rate)

Page footer: LD → Explore → /decoder/stats

# Log Ingestion

This screenshot shows the RSA [INV] View Events application interface. The window title is "Sessions". The "Time Window" is set to "All Data". There are two log entries displayed:

View	Date	Size	Details
<a href="#">View</a>	Oct 05 2014 20:45:38	85 bytes	sessionid: 1 device.ip: 127.0.0.1 medium: Logs device.type: ciscopix device.class: Firewall header.id: 0001 level: 1 event.desc: Failover cable OK. msg.id: 101001 event.cat.name: System.Heartbeats
<a href="#">View</a>	Oct 05 2014 20:45:42	86 bytes	sessionid: 2 device.ip: 127.0.0.1 medium: Logs device.type: ciscopix device.class: Firewall header.id: 0001 level: 1 event.desc: Bad failover cable. msg.id: 101001 event.cat.name: System.Heartbeats

A green callout box points to the first log entry with the text: "This time is in local timezone of the system where you are running the browser."

A green callout box points to the URL bar at the bottom of the browser window with the text: "URL to this SDK App is <http://<LD>:50102/sdk/app/sessions>".

The URL in the browser's address bar is: [http://cube:50102/sdk/app/sessions \[2/2\] All](http://cube:50102/sdk/app/sessions [2/2] All)

The logs can also be seen at <http://<LD>:50102/sdk/app/sessions>

# Log Ingestion

The screenshot shows a software interface titled "RSA [INV] View Events" with a tab labeled "Reports". The "Time Window" is set to "2014-Oct-05 9:15:42"-u. The "Filters" section contains several items:

- Data Size (2 values) [close]
- (1) (1)  
device.class (1 value) [close]
- (2)
- device.ip (1 value) [close]
- (2)
- device.type (1 value) [close]
- (2)  
ht.cat.name (1 value) [close]  
(2)
- header.id (1 value) [close]
- (2)

Annotations on the interface:

- A callout box points to the time window field with the text: "This time range is in UTC."
- A callout box points to the "ht.cat.name" filter item with the text: "This is showing that there is 1 'device.type' value that occurs 2 times. Clicking it would show us what that value is."
- A callout box points to the bottom status bar with the URL: "URL to this SDK App is http://<LD>:50102/sdk/app/reports".
- A callout box points to the bottom status bar with the URL: "http://cube:50102/sdk/app/reports < [2/2] Top".

Meta value counts can also be seen at <http://<LD>:50102/sdk/app/reports>



# Log Ingestion

The screenshot shows a window titled "RSA Security Analytics Login" with a tab labeled "Sessions". The window displays the following information:

- Time Window: "2014-Oct-05 9:15:42"-u
- Filters: device.type="ciscopix"(X)
- Sessions 1 - 0 of 0 >>
- sessionIdsRequest: /sdk?msg=query&id1=...&id2=...&size=20&query=select%20sessionid%20where%20time%3D%222014-Oct-05%209%3A15%3A42%22-u%20%26%20...&filter=device.type%3D%22ciscopix%22

A green callout box points from the text "device.type = 'ciscopix'" in the filter section to the explanatory text below.

Here we can see that we have logs of device.type = 'ciscopix'. Unfortunately, we can't see what those logs are because Log Decoder does not index this meta.

<http://cube:50102/sdk/app/sessions?filter=time=%222014-Oct-05 9:15:42%22-u&filter=device.type%3D%22ciscopix%22> < [2/2] All

No results because Log Decoder does not index this meta



# Log Ingestion

If we see `/etc/netWitness/ng/index-logdecoder.xml`, we'll find that only 'time' is indexed, i.e. it is defined with `level="IndexValues"`. No other metas are indexed, i.e. they are defined with `level="IndexNone"`.

This can be overridden in `/etc/netWitness/ng/index-logdecoder-custom.xml` but it is not recommended because Log Decoder is not meant to perform indexing on other metas. Concentrator is meant to do that.



# Log Ingestion Issues

## NetWitness Log Decoder





# Log Ingestion Issues

While performing investigation on Concentrator, if desired logs are not available, first rule out parsing failure.

Log Decoder may have failed to parse the log, so the '[device.type](#)' of the logs may be '[unknown](#)'. Query for `device.type = 'unknown'` and see if the logs can be found.

If the desired logs are found with `device.type = 'unknown'` meta then it is not a log ingestion issue because the log is available. Instead, it is a log parsing issue.

# Log Ingestion Issues

If it is confirmed that the logs cannot be found in Concentrator at all (even with `device.type = 'unknown'` filter), then check the following.

1. Can the logs be found in Log Decoder's SDK sessions app (<http://<LD>:50102/sdk/app/sessions>). Note: <LD> must be replaced with the IP address of the Log Decoder.
2. If there are too many logs in Log Decoder's SDK sessions app, they can be filtered by time range. The time range must be specified in UTC. An example:  
[http://<LD>:50102/sdk/app/sessions?filter=time="2014-Oct-05 9:15:45"- "2014-Oct-05 17:15:46"](http://<LD>:50102/sdk/app/sessions?filter=time='2014-Oct-05%209%3A15%3A45'-'2014-Oct-05%2017%3A15%3A46)
3. If the logs are available in Log Decoder's SDK sessions app, troubleshoot Concentrator. Is Concentrator aggregating?
4. If the logs are not available in Log Decoder's SDK sessions app, check if Log Decoder is capturing. Is TCP port 514 open? Is UDP port 514 open?
5. If Log Decoder is capturing, then check if any logs are arriving on port 514?

# Log Ingestion Issues

Test connectivity to Log Decoder port 514.

In Bash, and any POSIX shell, `:` is a null command. It does nothing. Therefore, we can use the `:` command instead of the `echo` command if we just want to test connectivity to TCP port 514 of Log Decoder.

```
# : > /dev/tcp/127.0.0.1/514
-bash: connect: Connection refused
-bash: /dev/tcp/127.0.0.1/514: Connection refused
```

(The above command is run on Log Decoder.)

If connection could not be established, then error messages are displayed.

```
# : > /dev/tcp/127.0.0.1/514
#
```

(The above command is run on Log Decoder.)

If connection could be established, then there is no output.



# Log Ingestion Issues

If Log Decoder is capturing, and yet the desired logs do not appear in Log Decoder's SDK sessions app (<http://<LD>:50102/sdk/app/sessions>) or in Concentrator → Investigation → Events page, then we need to find out if any logs are coming to Log Decoder.

The easiest way is to run `tcpdump` to capture all traffic on port 514.

```
# tcpdump -i any -w 514.pcap "port 514"
```

(The above command is run on Log Decoder.)

Run the above command on Log Decoder. Press 'Ctrl + C' when you think the Log Decoder should have received the desired logs.

All traffic to port 514 (TCP or UDP) would now be captured in the file [514.pcap](#).

This file may be viewed on the terminal (on Log Decoder itself) with the `tcpdump` command, or it may be transferred to a system with GUI Desktop and viewed with Wireshark. Also, please attach the PCAP file while opening a new JIRA ticket.

# Log Ingestion Issues

Command to view a PCAP file on the terminal itself.

```
# tcpdump -nnXr 514.pcap | less
```

(The above command is run wherever the PCAP file is.)

The above command with the **-X** option displays the content of each packet as a hexadecimal and ASCII dump. On scrolling down, one can see the various packets received on port 514. Here is one for example.

```
15:15:38.875840 IP 127.0.0.1.36441 > 127.0.0.1.514: Flags [P.], seq 1:50,
ack 1, win 257, options [nop,nop,TS val 1470803681 ecr 1470803681], length
49
0x0000: 4500 0065 932e 4000 4006 a962 7f00 0001 E..e..@.@@..b....
0x0010: 7f00 0001 8e59 0202 b015 3dce c3ad b9e7 .....Y.....=.....
0x0020: 8018 0101 fe59 0000 0101 080a 57aa aee1 .....Y.....W...
0x0030: 57aa aee1 3c31 3e20 2550 4958 2d31 2d31 W...<1>.%PIX-1-1
0x0040: 3031 3030 313a 2028 5052 494f 5249 5459 01001:(PRIORITY
0x0050: 2920 4661 696c 6f76 6572 2063 6162 6c65 ).Failover.cable
0x0060: 204f 4b2e 0a .OK..
```

# Log Ingestion Issues

Another command to view a PCAP file on the terminal.

```
# tcpdump -nnAr 514.pcap | less
```

(The above command is run wherever the PCAP file is.)

The above command with the -A option displays the content of each packet without the link layer headers as an ASCII dump. The TCP layer header appears as junk. This output is convenient to search for patterns with the `/<pattern>` command in `less`.

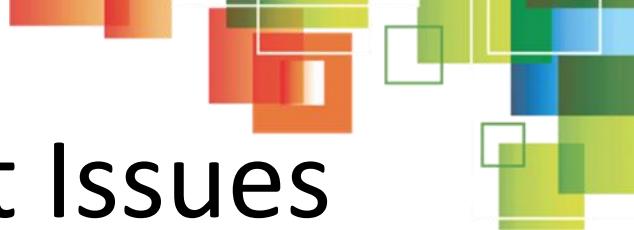
Example: One may search for the string “Failover cable OK” with the `/Failover cable OK` command and jump to a packet like the following.

```
15:15:38.875840 IP 127.0.0.1.36441 > 127.0.0.1.514: Flags [P.], seq  
1:50, ack 1, win 257, options [nop,nop,TS val 1470803681 ecr  
1470803681], length 49  
E..e..@.a..b.....Y....=.....Y.....  
W...W...<1> %PIX-1-101001: (PRIORITY) Failover cable OK.
```



# Unidentified Content Issues

## NetWitness Log Decoder



# Unidentified Content Issues

Valid log – A valid syslog payload

```
<1> %PIX-1-101001: (PRIORITY) Failover cable OK.
```

Typical log – Typically sent by Log Collector to Log Decoder

```
[][FOO-VLC1][10.0.0.1][1412278173101][] %PIX-1-101001: (PRIORITY) Failover cable OK.
```

Unknown log – A valid syslog payload but not supported by our device parsers

```
<1> hello world
```

Invalid log

```
(PRIORITY) Failover cable OK.
```

Another invalid log

```
foo bar
```



# Unidentified Content Issues

Let us feed all five logs we saw in the previous slide to Log Decoder and see what happens.

```
# echo "<1> %PIX-1-101001: (PRIORITY) Failover cable OK." > /dev/tcp/127.0.0.1/514
# echo "[][FOO-VLC1][10.0.0.1][1412278173101][] %PIX-1-101001: (PRIORITY) Failover
cable OK." > /dev/tcp/127.0.0.1/514
# echo "<1> hello world" > /dev/tcp/127.0.0.1/514
# echo "(PRIORITY) Failover cable OK." > /dev/tcp/127.0.0.1/514
# echo "foo bar" > /dev/tcp/127.0.0.1/514
```

(The above commands are run on Log Decoder.)

Valid log – Log Decoder accepts these logs. They have valid `device.type` meta.

Typical Log – Same as above

Unknown log – Log Decoder accepts these logs. But they have `device.type = 'unknown'`.

Invalid log – Log Decoder rejects these logs with ‘Unidentified content’ warnings.

# Unidentified Content Issues

The screenshot shows the RSA Security Analytics interface. The top navigation bar includes tabs for Investigation, Navigate, Events, Malware Analysis, and Security Analytics. Below the tabs, there are buttons for Last 3 Hours, Query, Use Profile, Detail View, and Actions. A search bar displays the query: device.type = 'ciscopix'.

The main content area shows a single event entry:

Event Time	Event Type	Event Theme	Size	Details
2014-10-05T20:48:44	Log	System.Heartbeats	85 bytes	<ul style="list-style-type: none"><li>↳ sessionid : 1</li><li>↳ device.ip : 127.0.0.1</li><li>↳ medium : 32</li><li>↳ device.type : ciscopix</li><li>↳ device.class : Firewall</li><li>↳ header.id : 0001</li><li>↳ level : 1</li><li>↳ event.desc : Failover cable OK.</li><li>↳ msg.id : 101001</li><li>↳ event.cat.name : System.Heartbeats</li><li>↳ did : cube</li><li>↳ rid : 1</li></ul> <p><a href="#">Hide Additional Meta</a> <a href="#">View Details</a></p> <p>↳ sessionid : 2</p>

At the bottom of the event details, there are navigation controls for pages, items per page (set to 25), and a link to the event details. The footer contains user information (admin, English, GMT+02:00), feedback links, and a session ID (10.3.4.10688-1).

Valid log – Log Decoder accepts these logs. They have valid ‘device.type’ meta.



# Unidentified Content Issues

The screenshot shows the RSA Security Analytics interface. The top navigation bar includes tabs for Investigation, Navigate, Events, Malware Analysis, and Security Analytics. Below the navigation is a toolbar with buttons for FOO-CONC, Last 3 Hours, Query, Use Profile, Detail View, and Actions. A search bar displays the query: device.type = 'ciscopix'.

The main content area shows a table with columns: Event Time, Event Type, Event Theme, Size, and Details. One row is selected, showing the following details:

- Event Time: 2014-10-05T20:48:46
- Event Type: Log
- Event Theme: System.Heartbeats
- Size: 108 bytes
- Details:
  - sessionid : 2
  - lc.cid : FOO-VLC1
  - forward.ip : 127.0.0.1
  - device.ip : 10.0.0.1 (highlighted with a green oval)
  - medium : 32
  - device.type : ciscopix (highlighted with a green oval)
  - device.class : Firewall
  - header.id : 0001
  - level : 1
  - event.desc : Failover cable OK.
  - msg.id : 101001
  - event.cat.name : System.Heartbeats
  - did : cube
  - rid : 2

At the bottom of the interface, there are navigation controls (back, forward, page numbers), a dropdown for items per page (set to 25), and a status bar showing admin, English (United States) GMT+02:00, Send Us Feedback, 10.3.4.10688-1, and the URL http://cube/investigation/events# < [1/1] Top.

Typical log – Log Decoder accepts these logs. They have valid ‘device.type’ meta.



# Unidentified Content Issues

The screenshot shows the RSA Security Analytics interface. The top navigation bar includes tabs for Investigation, Navigate, Events, Malware Analysis, and Security Analytics. Below the navigation is a search bar with the query "device.type = 'unknown'". The main content area displays a single event log entry from "2014-10-05T20:48:48" with a size of "52 bytes". To the right of the log, a "Details" pane is open, showing several fields: sessionid : 3, device.ip : 127.0.0.1, medium : 32, device.type : unknown (this field is circled in red), did : cube, and rid : 3. A "View Details" link is also present. At the bottom of the interface, there are pagination controls (Page 1 of 1), a items per page dropdown set to 25, and a status bar indicating 1 events. The bottom navigation bar includes links for admin, English (United States) GMT+02:00, Send Us Feedback, and the IP address 10.3.4.10688-1. The URL http://cube/investigation/events# < [1/1] Top is also visible.

Unknown log – Log Decoder accepts these logs. They have device.type = 'unknown'.



# Unidentified Content Issues

The screenshot shows a log viewer interface for the RSA Security Analytics platform. The top navigation bar includes tabs for Administration, Devices, and System, along with a search bar and user information. The main content area displays a log table with columns for Timestamp, Level, and Message. The log table shows several entries, with two specific entries circled in red. The first circled entry is at timestamp 2014-10-05T21:48:50.0, level WARN, message "Unidentified content from 127.0.0.1 received on receiver: '(PRIORITY) Failover cable OK.'". The second circled entry is at timestamp 2014-10-05T21:48:52.0, level WARN, message "Unidentified content from 127.0.0.1 received on receiver: 'foo bar'". The bottom of the screen shows pagination controls and a status bar with user info, feedback links, and a URL.

Timestamp	Level	Message
2014-10-05T21:48:24.0	DEBUG	User admin (session 552, [::ffff:10.31.244.44]:53301) has logged in
2014-10-05T21:48:24.0	DEBUG	User admin (session 552, [::ffff:10.31.244.44]:53301) has requested the SDK summary i...
2014-10-05T21:48:50.0	WARN	Unidentified content from 127.0.0.1 received on receiver: '(PRIORITY) Failover cable OK.'
2014-10-05T21:48:52.0	WARN	Unidentified content from 127.0.0.1 received on receiver: 'foo bar'
2014-10-05T21:48:58.0	DEBUG	User admin (session 576, 10.42.45.175:61049) has logged in

Invalid log – Log Decoder rejects these logs with ‘Unidentified content’ warnings.





# Unidentified Content Issues

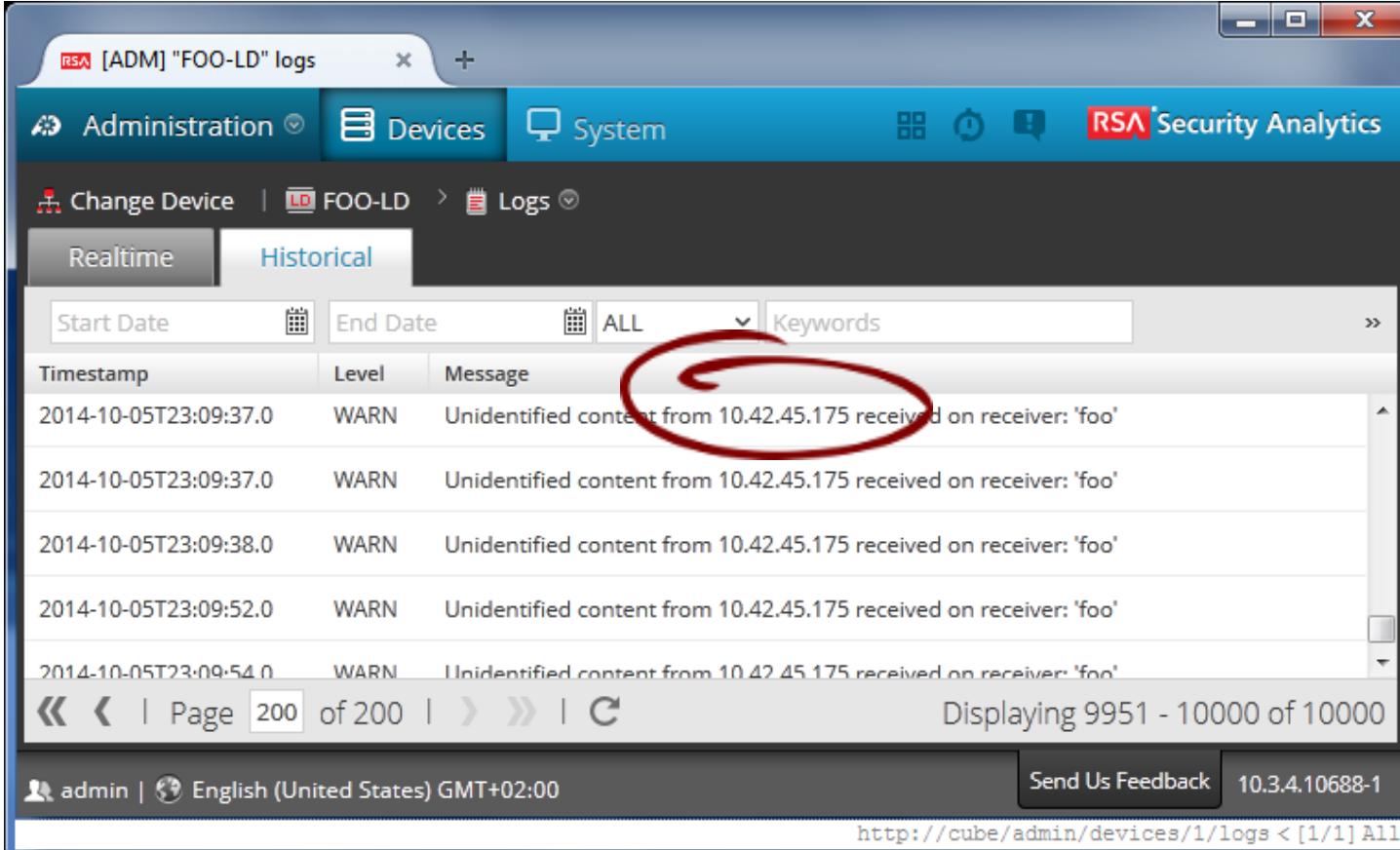
Ideally, invalid logs that cause ‘Unidentified content’ warnings should not reach Log Decoder.

But, in practice, sometimes invalid logs do reach Log Decoder.

**Where are the invalid logs coming from?**

We have to check the logs to figure out.

# Unidentified Content Issues



The screenshot shows a log viewer interface for the device 'FOO-LD'. The logs are filtered to show 'WARN' level messages related to 'Unidentified content' from the IP address '10.42.45.175'. A red circle highlights the first log entry. The log table has columns for Timestamp, Level, and Message. The message column contains repeated entries of 'Unidentified content from 10.42.45.175 received on receiver: 'foo''. The bottom of the screen shows pagination information, indicating pages 200 through 10000, and a URL for the log viewer.

Timestamp	Level	Message
2014-10-05T23:09:37.0	WARN	Unidentified content from 10.42.45.175 received on receiver: 'foo'
2014-10-05T23:09:37.0	WARN	Unidentified content from 10.42.45.175 received on receiver: 'foo'
2014-10-05T23:09:38.0	WARN	Unidentified content from 10.42.45.175 received on receiver: 'foo'
2014-10-05T23:09:52.0	WARN	Unidentified content from 10.42.45.175 received on receiver: 'foo'
2014-10-05T23:09:54.0	WARN	Unidentified content from 10.42.45.175 received on receiver: 'foo'

Displaying 9951 - 10000 of 10000

admin | English (United States) GMT+02:00 Send Us Feedback 10.3.4.10688-1  
[http://cube/admin/devices/1/logs < \[1/1\] All](http://cube/admin/devices/1/logs < [1/1] All)

Check the IP address in the 'Unidentified content' warnings



# Unidentified Content Issues

If the invalid logs are coming from another system, we need to figure out why that system is sending invalid logs to our Log Decoder.

If the invalid logs are coming from '127.0.0.1' then we need to figure which process is sending the invalid logs to Log Decoder.

# Unidentified Content Issues

Command to figure the list of all processes connecting to Log Decoder.

```
# netstat -nopa | grep ":514\>"
```

(The above command is run on Log Decoder. The "`\>`" in the regular expression ensures that it matches "`:514`" but not "`:5140`" etc.)

One may see some other process in the output.

tcp	0	0	0.0.0.0:514	0.0.0.0:*	LISTEN	7362/NwLogDecoder	off	(0.00/0/0)
tcp	0	0	10.101.202.84:514	10.101.202.82:50706	ESTABLISHED	7362/NwLogDecoder	off	(0.00/0/0)
tcp	0	0	10.101.202.84:514	10.100.200.23:47962	ESTABLISHED	7362/NwLogDecoder	off	(0.00/0/0)
tcp	0	0	10.101.202.84:514	10.101.202.84:14529	ESTABLISHED	7362/NwLogDecoder	off	(0.00/0/0)
tcp	0	0	10.101.202.84:514	10.101.202.85:38824	ESTABLISHED	7362/NwLogDecoder	off	(0.00/0/0)
tcp	0	0	10.101.202.84:14529	10.101.202.84:514	ESTABLISHED	2743/rsyslogd	off	(0.00/0/0)
tcp	0	0	10.101.202.84:514	10.101.202.86:52873	ESTABLISHED	7362/NwLogDecoder	off	(0.00/0/0)
tcp	0	0	127.0.0.1:514	127.0.0.1:55773	ESTABLISHED	7362/NwLogDecoder	off	(0.00/0/0)
tcp	0	0	127.0.0.1:55773	127.0.0.1:514	ESTABLISHED	2578/NwLogCollector	off	(0.00/0/0)
tcp	0	0	:::514	:::*	LISTEN	7362/NwLogDecoder	off	(0.00/0/0)
udp	0	0	0.0.0.0:514	0.0.0.0:*		7362/NwLogDecoder	off	(0.00/0/0)
udp	0	0	:::514	:::*		7362/NwLogDecoder	off	(0.00/0/0)

Since in this output '`rsyslogd`' seems to be connecting to Log Decoder, one needs to investigate if '`rsyslogd`' is the culprit.



# Unidentified Content Issues

Command to figure the list of all processes connecting to Log Decoder.

```
# netstat -nopa | grep ":514\>"
```

(The above command is run on Log Decoder. The "\>" in the regular expression ensures that it matches ":514" but not ":5140" etc.)

One may see some other process in the output.

**Based on a true story.**

tcp	0	0 0.0.0.0:*	LISTEN	7362/NwLogDecoder	off (0.00/0/0)
tcp	0	0 10.101.202.84:514	ESTABLISHED	7362/NwLogDecoder	off (0.00/0/0)
tcp	0	0 10.101.202.84:514	ESTABLISHED	7362/NwLogDecoder	off (0.00/0/0)
tcp	0	0 10.101.202.84:514	ESTABLISHED	7362/NwLogDecoder	off (0.00/0/0)
tcp	0	0 10.101.202.84:514	ESTABLISHED	7362/NwLogDecoder	off (0.00/0/0)
tcp	0	0 10.101.202.84:514	ESTABLISHED	7362/NwLogDecoder	off (0.00/0/0)
tcp	0	0 10.101.202.84:14529	ESTABLISHED	2743/rsyslogd	off (0.00/0/0)
tcp	0	0 10.101.202.84:14529	ESTABLISHED	7362/NwLogDecoder	off (0.00/0/0)
tcp	0	0 10.101.202.84:514	ESTABLISHED	7362/NwLogDecoder	off (0.00/0/0)
tcp	0	0 127.0.0.1:514	ESTABLISHED	7362/NwLogDecoder	off (0.00/0/0)
tcp	0	0 127.0.0.1:55773	ESTABLISHED	2578/NwLogCollector	off (0.00/0/0)
tcp	0	0 :::514	LISTEN	7362/NwLogDecoder	off (0.00/0/0)
udp	0	0 0.0.0.0:514		7362/NwLogDecoder	off (0.00/0/0)
udp	0	0 :::514		7362/NwLogDecoder	off (0.00/0/0)

Since in this output 'rsyslogd' seems to be connecting to Log Decoder, one needs to investigate if 'rsyslogd' is the culprit.



# Unidentified Content Issues

Command to figure the list of all processes connecting to Log Decoder.

```
# netstat -nopa | grep ":514\>"
```

(The above command is run on Log Decoder. The "`\>`" in the regular expression ensures that it matches "`:514`" but not "`:5140`" etc.)

Sometimes there may not be any other process to blame. So Log Collector becomes suspect.

tcp	0	0 0.0.0.0:6514	0.0.0.0:*	LISTEN	3345/NwLogDecoder	off	(0.00/0/0)
tcp	0	0 0.0.0.0:514	0.0.0.0:*	LISTEN	3345/NwLogDecoder	off	(0.00/0/0)
tcp	0	0 127.0.0.1:32809	127.0.0.1:514	ESTABLISHED	2109/NwLogCollector	off	(0.00/0/0)
tcp	0	0 127.0.0.1:514	127.0.0.1:32809	ESTABLISHED	3345/NwLogDecoder	off	(0.00/0/0)
tcp	0	0 :::6514	:::*	LISTEN	3345/NwLogDecoder	off	(0.00/0/0)
tcp	0	0 :::514	:::*	LISTEN	3345/NwLogDecoder	off	(0.00/0/0)
udp	0	0 0.0.0.0:514	0.0.0.0:*		3345/NwLogDecoder	off	(0.00/0/0)
udp	0	0 :::514	:::*		3345/NwLogDecoder	off	(0.00/0/0)

In this case, we need to capture traffic arriving on port 514 with '[tcpdump](#)' to understand what the invalid logs are and try to deduce where Log Collector is getting such logs from.

# Unidentified Content Issues

Command to figure the list of all processes connecting to Log Decoder.

## Based on a true story.

```
# netstat -nopa | grep ":514\>"
```

(The above command is run on Log Decoder. The "\>" in the regular expression ensures that it matches ":514" but not ":5140" etc.)

Sometimes there may not be any other process to blame. So Log Collector becomes suspect.

```
tcp      0      0 0.0.0.0:6514          0.0.0.0:*          LISTEN    3345/NwLogDecoder off (0.00/0/0)
tcp      0      0 0.0.0.0:514           0.0.0.0:*          LISTEN    3345/NwLogDecoder off (0.00/0/0)
tcp      0      0 127.0.0.1:22809     127.0.0.1:514      LISTEN    3345/NwLogDecoder off (0.00/0/0)
tcp      0      0 127.0.0.1:514        0.0.0.0:514       LISTEN    3345/NwLogDecoder off (0.00/0/0)
tcp      0      0 0.0.0.0:514         ::*:*                LISTEN    3345/NwLogDecoder off (0.00/0/0)
tcp      0      0 0.0.0.0:514         0.0.0.0:514       LISTEN    3345/NwLogDecoder off (0.00/0/0)
tcp      0      0 0.0.0.0:514         ::*:*                LISTEN    3345/NwLogDecoder off (0.00/0/0)
tcp      0      0 0.0.0.0:514         0.0.0.0:514       LISTEN    3345/NwLogDecoder off (0.00/0/0)
tcp      0      0 0.0.0.0:514         ::*:*                LISTEN    3345/NwLogDecoder off (0.00/0/0)
```

## Log Collector was found to be sending logs with newlines. A fix was developed for Log Collector.





# Unidentified Content Issues

We need to capture packets while the ‘[Unidentified content](#)’ warnings are being logged by Log Decoder, not after they are logged, so that we capture the packets that caused the issue.

This can be tricky if the ‘[Unidentified content](#)’ warnings comes in short bursts after long intervals. In such a case, we don’t know when to run the `tcpdump` command and how long to wait.

So we wrote a tool called ‘[autocap](#)’ to automatically monitor `/var/log/messages` and start ‘`tcpdump`’ and capture the output of ‘`netstat`’ automatically when ‘[Unidentified content](#)’ warnings are found, and stop the capture when sufficient number of such warnings have been found.

# Unidentified Content Issues

- Download ‘autocap’ from RSA and copy it to Log Decoder.
- In the shell, change your current directory to the directory where the script has been copied.
- Then execute the following command.

```
# nohup sh autocap -s "Unidentified content" > autocap.txt &
```

- The script would now log its progress to ‘autocap.txt’. You may use the following command to monitor its progress.

```
# tail -f autocap.txt
```

- In the end, the script will tell the user to send a tarball to RSA. Here is an example.

```
Send _autocap_2014-10-06_03-39-33.tar.gz to RSA.
```

- Pick up the .tar.gz file the script mentioned in its last message and use it for analysis. It’ll contain a PCAP with traffic for port 514 and second-by-second output from netstat for the duration when the issue was occurring.
- Attach this .tar.gz file to any JIRA ticket you create for ‘Unidentified content’ issue.



# Log Parsing

## NetWitness Log Decoder



# Log Parsing

This is an example of a valid and supported log.

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

When we feed this to Log Decoder, it parses fine and generates metas including device.type = 'apache'.

<input type="checkbox"/> 2014-10-06T14:53:13 Log	Content.Web Traffic	122 bytes	<ul style="list-style-type: none"><li>↔ sessionid : 1</li><li>⊕ device.ip : 127.0.0.1</li><li>⊕ column : 32</li><li>⊕ device.type : apache</li><li>⊕ device.class : Web Logs</li><li>⊕ header.id : 0003</li><li>⊕ level : 6</li><li>⊕ result.code : 200</li><li>⊕ bytes.src : 83</li><li>⊕ event.time : 2014-Oct-01 08:09:10.000</li><li>⊕ msg.id : 00100:01</li><li>⊕ event.cat.name : Content.Web Traffic</li><li>⊕ did : cube</li><li>⊕ rid : 1</li></ul>
--	---------------------	-----------	--



# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

The device log parsers are written in XML.

Log Decoder reads these XMLs and creates parsers to parse logs as defined in the XML.

There is a parser directory corresponding to each supported device in </etc/netWitness/ng/envision/etc/devices>.

For example, for Apache logs, the parser directory is </etc/netWitness/ng/envision/etc/devices/apache>. This directory contains a file called [v20\\_apachemsg.xml](#) that defines the supported Apache logs.





# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

[ ]

Header

Message

The header in the log should match a header definition in the device parser XML.

The message in the log should match a message definition in the device parser XML.





# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

[ ]

Header

Message

This log matches the following header definition in  
[/etc/netWitness/ng/envision/etc/devices/apache/v20\\_apachemsg.xml](/etc/netWitness/ng/envision/etc/devices/apache/v20_apachemsg.xml).

```
<HEADER
  id1="0003"
  id2="0003"
  content="%APACHE-&lt;level&gt;-&lt;messageid&gt;:&lt;!payload&gt;" />
```

After a matching message definition is found for the message, a meta called '[header.id](#)' is created with the [id1](#) of the matching header definition as its value. For example, for this log, [header.id = '0003'](#) is created.



# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

Header

Message

				↔ sessionid : 1 ⊕ device.ip : 127.0.0.1 ⊖ medium : 32 ⊕ device.type : apache ⊕ device.class : Web Logs ⊕ header.id : 0003 ⊖ level : 6 ⊕ result.code : 200 ⊕ bytes.src : 83 ⊕ event.time : 2014-Oct-01 08:09:10.000 ⊕ msg.id : 00100:01 ⊕ event.cat.name : Content.Web Traffic ⊖ did : cube ⊖ rid : 1
<input type="checkbox"/>	2014-10-06T14:53:13 Log	Content.Web Traffic	122 bytes	

# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

Header

Message

This log matches this header definition in  
[/etc/netWitness/ng/envision/etc/devices/apache/v20\\_apachemsg.xml](/etc/netWitness/ng/envision/etc/devices/apache/v20_apachemsg.xml).

```
<HEADER
  id1="0003"
  id2="0003"
  content="%APACHE-&lt;level&gt;-&lt;messageid&gt;: &lt;!payload&gt;" />
```

During the header match, the ‘messageid’ field in the header definition matches ‘GET’ in the log, so now a message definition with id2 = “GET” is looked up.



# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

[ ]

Header

Message

Message definition with `id2="GET"` is found.

```
<MESSAGE
  level="6"
  tableid="82"
  parse="1"
  parsedefvalue="1"
  eventcategory="1204000000"
  sitetrack="1"
  id1="00100·01"
  id2="GET"
  summary="NIC_B_ADDRESS_ACCOUNTING;"
  content="<saddr> <fld5> <username> [<fld7> <timezone>]
<web_method> <webpage> <network_service> <resultcode> <sbytes>
<@:SYSVAL($MSGID,$ID1)>&lt;@event_time:>*EVNTTIME($MSG, '%D/%B/%W:%N:%U:%O', fld7)>" />
```





# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

[ ] [ ]

Header

Message

```
<MESSAGE
  level="6"
  tableid="82"
  parse="1"
  parsedefvalue="1"
  eventcategory="1204000000"
  sitetrack="1"
  id1="00100:01"
  id2="GET" ...
```

After a matching message definition is found for the message, a meta called ‘msg.id’ is created with the ‘id1’ attribute’s value. For example, for this log, msg.id = '00100:01' is created.



# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

Header

Message

□ 2014-10-06T14:53:13 Log	Content.Web Traffic	122 bytes	↔ sessionid : 1 ⊕ device.ip : 127.0.0.1 ⊖ medium : 32 ⊕ device.type : apache ⊕ device.class : Web Logs ⊕ header.id : 0003 ⊕ level : 6 ⊕ result.code : 200 ⊕ bytestrc : 83 ⊕ event.time : 2014-Oct-01 08:09:10.000 ⊕ msg.id : 00100:01 ⊕ event.cat.name : Content.Web Traffic ⊖ did : cube ⊕ rid : 1
---------------------------	---------------------	-----------	--



# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

[ ] [ ]

Header

Message

```
<MESSAGE
  level="6"
  tableid="82"
  parse="1"
  parsedefvalue="1"
  eventcategory="1204000000"
  sitetrack="1"
  id1="00100:01" ...
```

The '[eventcategory](#)' number is looked up in [/etc/netWitness/ng/envision/etc/ecat.ini](#). It contains the following entry. The string is used as the value for '[event.cat.name](#)' meta.

```
1204000000,Content.Web Traffic
```



# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

Header

Message

				↔ sessionid : 1 ⊕ device.ip : 127.0.0.1 ⊖ medium : 32 ⊕ device.type : apache ⊕ device.class : Web Logs ⊕ header.id : 0003 ⊕ level : 6 ⊕ result.code : 200 ⊕ bytes.src : 83 ⊕ event.time : 2014-Oct-01 08:09:10.000 ⊕ msg.id : 00100:01 ⊕ event.cat.name : Content.Web Traffic ⊖ did : cat ⊕ rid : 1
<input type="checkbox"/>	2014-10-06T14:53:13 Log	Content.Web Traffic	122 bytes	

# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

Header

Message

```
...
content=&lt;saddr&gt; &lt;fld5&gt; &lt;username&gt; [&lt;fld7&gt; &lt;timezone&gt;]
&lt;web_method&gt; &lt;webpage&gt; &lt;network_service&gt; &lt;resultcode&gt; &lt;sbytes&gt;
&lt;@:/*SYSVAL($MSGID,$ID1)&gt;&lt;@event_time:*EVNTTIME($MSG,'%D/%B/%W:%N:%U:%O',fld7)&gt; " />
```



Apart from the fields extracted directly from the log, an additional 'event\_time' field is created by parsing the time extracted in the 'fld7' field.

# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

[ ] [ ]

Header

Message

```
...
content=&lt;saddr&gt; &lt;fld5&gt; &lt;username&gt; [&lt;fld7&gt; &lt;timezone&gt;]
&lt;web_method&gt; &lt;webpage&gt; &lt;network_service&gt; &lt;resultcode&gt; &lt;sbytes&gt;
&lt;@:/*SYSVAL($MSGID,$ID1)&gt;&lt;@event_time:*EVNTTIME($MSG,'%D/%B/%W:%N:%U:%O',fld7)&gt; " />
```

The parser fields are mapped to meta keys in [/etc/netWitness/ng/envision/etc/table-map.xml](#).





# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

[ ]

Header

Message

```
...
content=&lt;saddr&gt; &lt;fld5&gt; &lt;username&gt; [&lt;fld7&gt; &lt;timezone&gt;]
&lt;web_method&gt; &lt;webpage&gt; &lt;network_service&gt; &lt;resultcode&gt; &lt;sbytes&gt;
&lt;@:/*SYSVAL($MSGID,$ID1)&gt;&lt;@event_time:*EVNTTIME($MSG,'%D/%B/%W:%N:%U:%O',fld7)&gt; " />
```

An example mapping in [/etc/netWitness/ng/envision/etc/table-map.xml](#).

```
<mapping envisionName="event_time" nwName="event.time" flags="None" format="TimeT"
envisionDisplayName="EventDate/Time|EventTime|LastScanned|Created"/>
```

This maps the ‘`event_time`’ field in the device parser to ‘`event.time`’ meta. Since this meta has `flags="None"`, this meta is written to the disk.





# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

[ ] [ ]

Header

Message

				↔ sessionid : 1 ⊕ device.ip : 127.0.0.1 ⊖ medium : 32 ⊕ device.type : apache ⊕ device.class : Web Logs ⊕ header.id : 0003 ⊕ level : 6 ⊕ result.code : 200 ⊕ bytes : 83 ⊕ event.time : 2014-Oct-01 08:09:10.000 ⊕ msg.id : 00100:01 ⊕ event.cat.name : Content.Web Traffic ⊖ did : cube ⊕ rid : 1
<input type="checkbox"/>	2014-10-06T14:53:13	Log	Content.Web Traffic	122 bytes



EMC<sup>2</sup>

# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

[ ]

Header

Message

```
...
content=&lt;saddr&gt; &lt;fld5&gt; &lt;username&gt; [&lt;fld7&gt; &lt;timezone&gt;]
&lt;web_method&gt; &lt;webpage&gt; &lt;network_service&gt; &lt;resultcode&gt; &lt;sbytes&gt;
&lt;@:/*SYSVAL($MSGID,$ID1)&gt;&lt;@event_time:*EVNTTIME($MSG,'%D/%B/%W:%N:%U:%O',fld7)&gt; " />
```

An example mapping in [/etc/netWitness/ng/envision/etc/table-map.xml](#).

```
<mapping envisionName="username" nwName="user.dst" flags="None"
envisionDisplayName="UserName|UserID|User|UserName|Username" nullTokens="none| - "/>
```

The ‘username’ field matches ‘-’ (hyphen) in the log which is defined as a null token in this mapping, so this is not written to the disk.



# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

[ ] [ ]

Header

Message

```
...
content=&lt;saddr&gt; &lt;fld5&gt; &lt;username&gt; [&lt;fld7&gt; &lt;timezone&gt;]
&lt;web_method&gt; &lt;webpage&gt; &lt;network_service&gt; &lt;resultcode&gt; &lt;sbytes&gt;
&lt;@:/*SYSVAL($MSGID,$ID1)&gt;&lt;@event_time:*EVNTTIME($MSG,'%D/%B/%W:%N:%U:%O',fld7)&gt; " />
```

An example mapping in [/etc/netWitness/ng/envision/etc/table-map.xml](#).

```
<mapping envisionName="webpage" nwName="web.page" flags="Transient"
envisionDisplayName="WebPage"/>
```

This maps the ‘[webpage](#)’ field in the device parser to ‘[web.page](#)’ meta. Since this meta has [flags="Transient"](#), this meta is not written to the disk.



# Log Parsing

<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83

U U

## Message

```
...  
content=<saddr> <fld5> <username> [<fld7> <timezone>]  
<web_method> <webpage> <network_service> <resultcode> <sbytes>  
<@*:SYSVAL($MSGID,$ID1)> <@event_time:>*EVNTTIME($MSG, '%D/%B/%W:%N:%U:%O', fld7)> " />
```

An example mapping in `/etc/netwitness/ng/envision/etc/table-map.xml`.

```
<mapping envisionName="sbytes" nwName="bytes.src" flags="None" format="UInt64"  
nullTokens="(null)|-"/>
```

This maps the ‘`sbytes`’ field in the device parser to ‘`bytes.src`’ meta. Since this meta has `flags="None"`, this meta is written to the disk.



# Log Parsing

```
<6> %APACHE-6-GET: 10.0.0.1 - - [01/Oct/2014:08:09:10 +0100] GET /foo HTTP/1.1 200 83
```

[ ] [ ]

Header

Message

□ 2014-10-06T14:53:13 Log	Content.Web Traffic	122 bytes	<ul style="list-style-type: none"><li>↔ sessionid : 1</li><li>⊕ device.ip : 127.0.0.1</li><li>⊖ medium : 32</li><li>⊕ device.type : apache</li><li>⊕ device.class : Web Logs</li><li>⊕ header.id : 0003</li><li>⊕ level : 6</li><li>⊕ result.code : 200</li><li>⊕ bytes.src : 83</li><li>⊖ event.time : 2014-Oct-01 08:09:10.000</li><li>⊕ msg.id : 00100:01</li><li>⊕ event.cat.name : Content.Web Traffic</li><li>⊖ did : cube</li><li>⊕ rid : 1</li></ul>
---------------------------	---------------------	-----------	--



# Missing Metas

## NetWitness Log Decoder



# Missing Metas

The screenshot shows the RSA Security Analytics interface with the title bar "RSA [INV] View Events". The navigation bar includes "Investigation", "Events", "Malware Analysis", and "Security Analytics". Below the navigation bar, there are filters: "Foo-CONC", "Last 3 Hours", "Query", "Use Profile", "Detail View", and "Actions". A search bar contains the query "device.type = 'apache'".

The main content area displays two log entries:

- Log 1:** Contains the meta "msg.id : 00100" (highlighted with a green arrow). It also lists other metadata: event.cat.name: Content.Web Traffic, did: cube, rid: 9. There is a link to "Hide Additional Meta" and another to "View Details".
- Log 2:** Contains the meta "sessionid : 11" (highlighted with a red arrow). It also lists other metadata: lc.cid: FOO-VLC, forward.ip: 127.0.0.1, device.ip: 10.2.0.2, medium: 32, device.type: apache, did: cube, rid: 11. There is a link to "View Details".

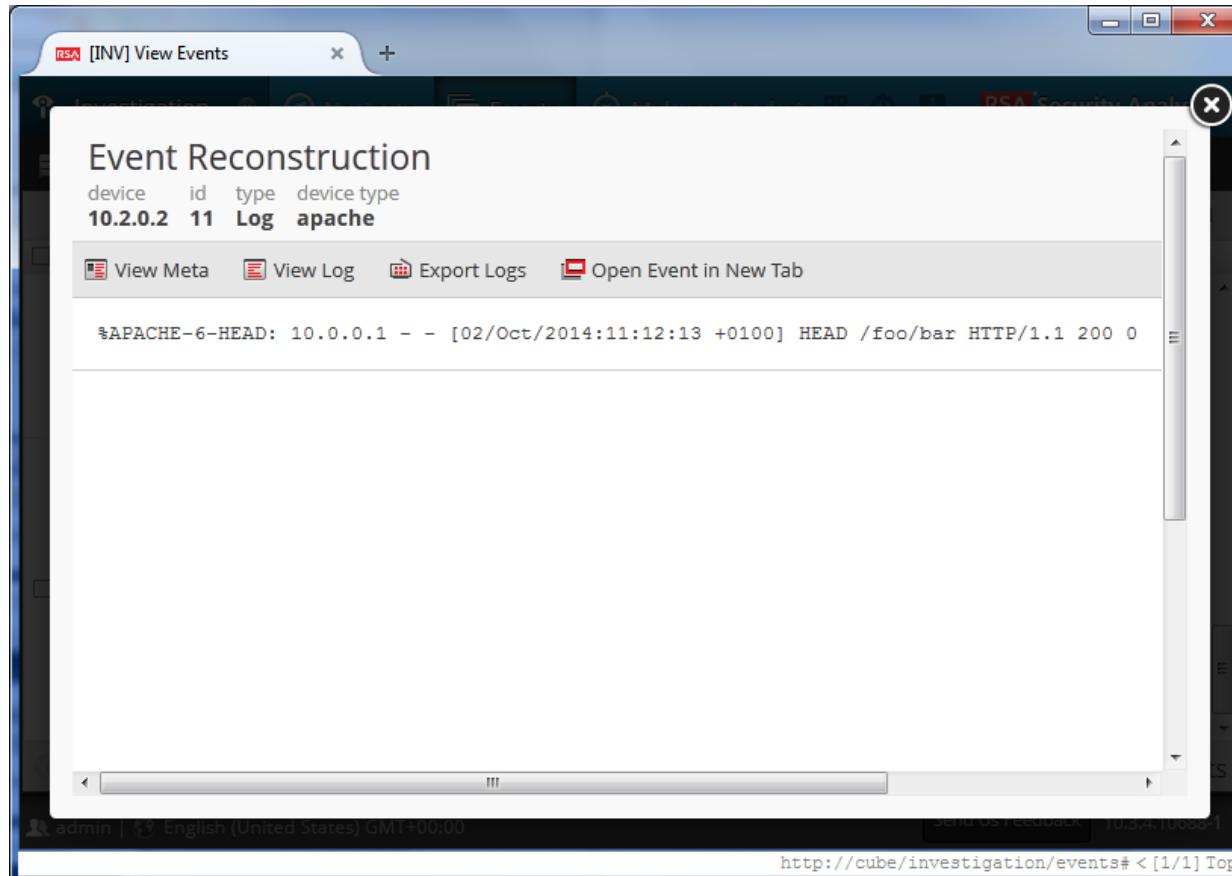
At the bottom, there are pagination controls ("Page 1", "25 items per page", "5 events"), user information ("admin | English (United States) GMT+00:00"), a feedback button ("Send Us Feedback"), and a URL "http://cube/investigation/events# < [1/1] Top".

**All logs should have msg.id meta**

**There is no msg.id meta for this log. This is an issue!**

'device.type' is present but 'msg.id' is missing

# Missing Metas



Is this a supported log?





# Missing Metas

## Log

```
%APACHE-6-HEAD: 10.0.0.1 - - [02/Oct/2014:11:12:13 +0100] HEAD /foo/bar HTTP/1.1 200 0
```

When we see ‘`device.type`’ meta for a log, but we don’t see any ‘`msg.id`’ meta for it, it implies that the log matched a header definition of a device parser, but it did not match a message definition.

In fact, we can find a matching header definition in  
`/etc/netwitness/ng/envision/etc/devices/apache/v20_apachemsg.xml`

```
<HEADER
    id1="0003"
    id2="0003"
    content="%APACHE-&lt;level&gt;-&lt;messageid&gt;: &lt;!payload&gt;" />
```



# Missing Metas

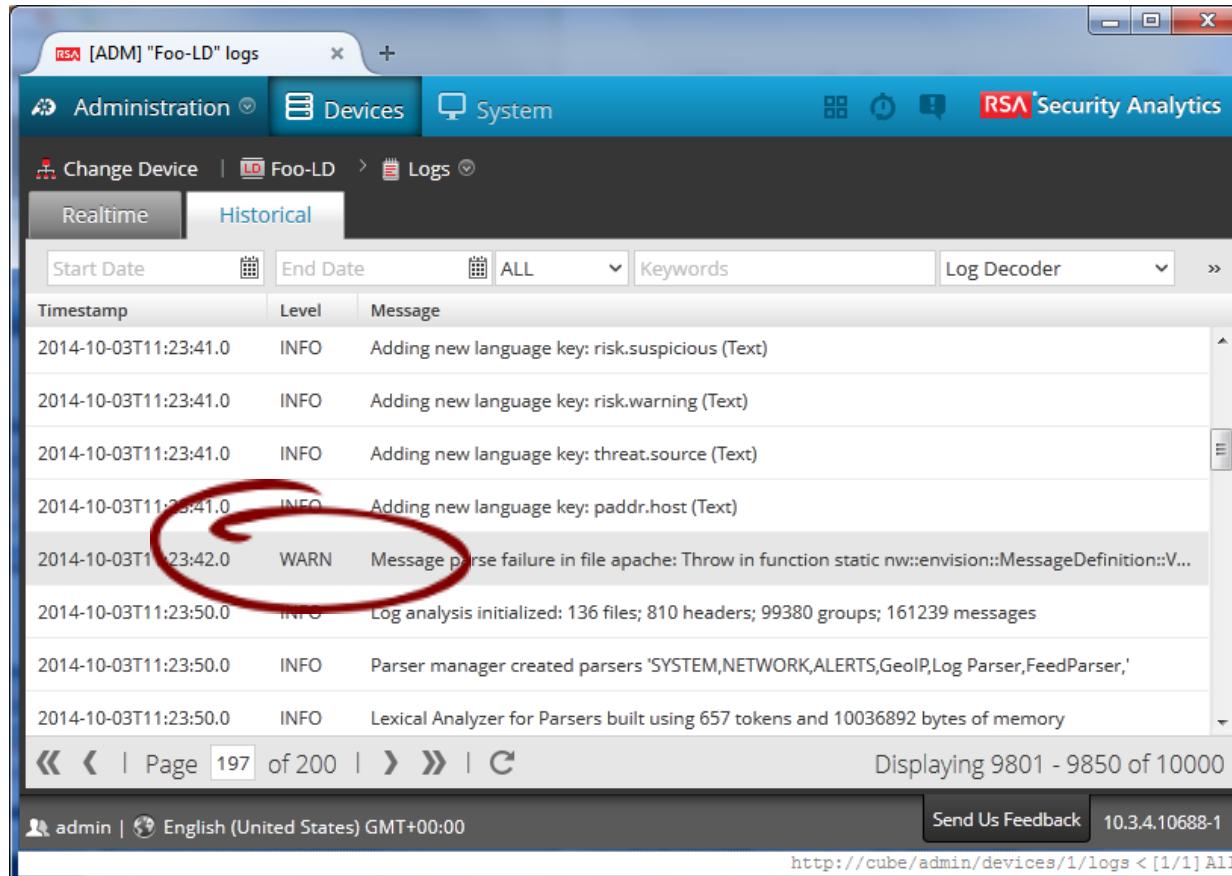
## Log

```
%APACHE-6-HEAD: 10.0.0.1 - - [02/Oct/2014:11:12:13 +0100] HEAD /foo/bar HTTP/1.1 200 0
```

There is a matching Message ID (`id2="HEAD"`) as well in  
`/etc/netwitness/ng/envision/etc/devices/apache/v20_apachemsg.xml`

```
<MESSAGE
    level="6"
    tableid="82"
    parse="1"
    parsedefvalue="1"
    eventcategory="1204000000"
    sitetrack="1"
    id1="00110:01"
    id2="HEAD"
    summary="NIC_B_ADDRESS_ACCOUNTING;"
    content="&lt;saddr&gt; &lt;fld5&gt; &lt;username&gt; [&lt;fld7&gt; &lt;timezone&gt;]
&lt;web_method&gt; &lt;webpage&gt; &lt;network_service&gt; &lt;resultcode&gt; &lt;sbytes&gt;
&lt;@:SYSVAL($MSGID,$ID1)&gt;&lt;@event_time:&*EVNTTIME($MSG,'%D/%B/%W:%N:%U:%O',fld6)&gt; " />
```

# Missing Metas



The screenshot shows a log viewer interface from RSA Security Analytics. The title bar reads "RSA [ADM] 'Foo-LD' logs". The top navigation bar includes "Administration", "Devices" (selected), and "System". The sub-navigation shows "Change Device" and "Logs" (selected). Below this is a search/filter bar with "Start Date", "End Date", "Level" (set to ALL), "Keywords", and "Log Decoder". The main area displays a table of log entries:

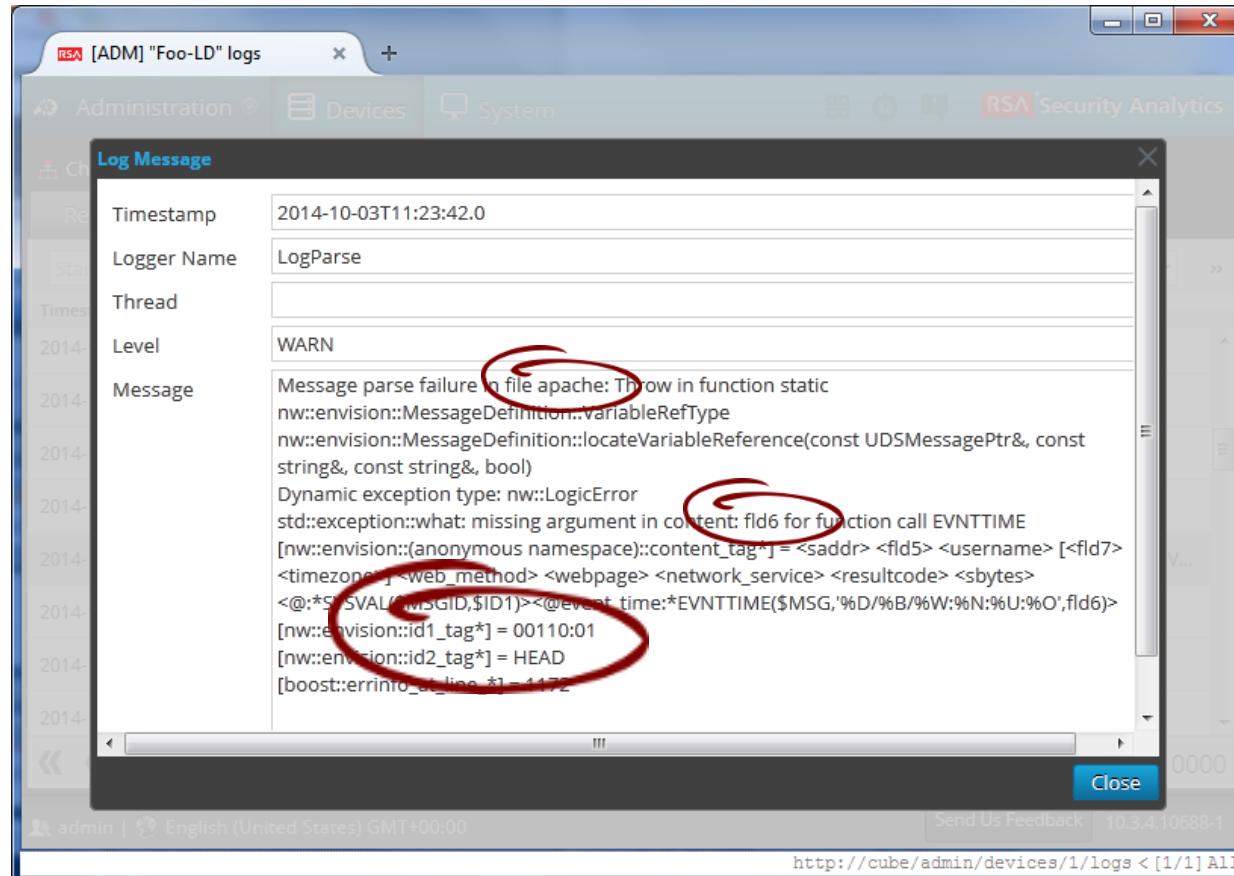
Timestamp	Level	Message
2014-10-03T11:23:41.0	INFO	Adding new language key: risk.suspicious (Text)
2014-10-03T11:23:41.0	INFO	Adding new language key: risk.warning (Text)
2014-10-03T11:23:41.0	INFO	Adding new language key: threat.source (Text)
2014-10-03T11:23:41.0	INFO	Adding new language key: paddr.host (Text)
2014-10-03T11:23:42.0	WARN	Message parse failure in file apache: Throw in function static nw::envision::MessageDefinition::V...
2014-10-03T11:23:50.0	INFO	Log analysis initialized: 136 files; 810 headers; 99380 groups; 161239 messages
2014-10-03T11:23:50.0	INFO	Parser manager created parsers 'SYSTEM,NETWORK,ALERTS,GeolP,Log Parser,FeedParser,'
2014-10-03T11:23:50.0	INFO	Lexical Analyzer for Parsers built using 657 tokens and 10036892 bytes of memory

A red circle highlights the second-to-last log entry, which contains the message "Message parse failure in file apache: Throw in function static nw::envision::MessageDefinition::V...". The bottom of the screen shows pagination controls ("Page 197 of 200"), a status message ("Displaying 9801 - 9850 of 10000"), and user information ("admin | English (United States) GMT+00:00").

'Message parse failure' warning indicates a semantic error in device parser



# Missing Metas



Error in 'apache' parser in the message tag with id1="00110:01" and id2="HEAD"



# Missing Metas

## Log

```
%APACHE-6-HEAD: 10.0.0.1 - - [02/Oct/2014:11:12:13 +0100] HEAD /foo/bar HTTP/1.1 200 0
```

EVNTTIME(\$MSG, ...) function must use variables defined in this message definition.  
fld6 is used in EVNTTIME(\$MSG, ...) but it is not defined anywhere here.

```
<MESSAGE
    level="6"
    tableid="82"
    parse="1"
    parsedefvalue="1"
    eventcategory="1204000000"
    sitetrack="1"
    id1="00110:01"
    id2="HEAD"
    summary="NIC_B_ADDRESS_ACCOUNTING;"
    content="&lt;saddr&gt; &lt;fld5&gt; &lt;username&gt; [&lt;fld7&gt; &lt;timezone&gt;]
&lt;web_method&gt; &lt;webpage&gt; &lt;network_service&gt; &lt;resultcode&gt; &lt;sbytes&gt;
&lt;@:SYSVAL($MSGID,$ID1)&gt;&lt;@event_time:&gt;*EVNTTIME($MSG, '%D/%B/%W:%N:%U:%O', fld6)&gt; " />
```

# Missing Metas

## Log

```
%APACHE-6-HEAD: 10.0.0.1 - - [02/Oct/2014:11:12:13 +0100] HEAD /foo/bar HTTP/1.1 200 0
```

`fld7` in the definition below matches the event time in the log, so `fld7` is the right argument to use with the `EVNTTIME($MSG, ...)` function.

```
<MESSAGE
    level="6"
    tableid="82"
    parse="1"
    parsedefvalue="1"
    eventcategory="1204000000"
    sitetrack="1"
    id1="00110:01"
    id2="HEAD"
    summary="NIC_B_ADDRESS_ACCOUNTING;"
    content="&lt;saddr&gt; &lt;fld5&gt; &lt;username&gt; [&lt;fld7&gt; &lt;timezone&gt;]
&lt;web_method&gt; &lt;webpage&gt; &lt;network_service&gt; &lt;resultcode&gt; &lt;sbytes&gt;
&lt;@:SYSVAL($MSGID,$ID1)&gt;&lt;@event_time:&gt;*EVNTTIME($MSG, '%D/%B/%W:%N:%U:%O', fld7)&gt; " />
```

# Missing Metas

## Log

# This is not a real issue in the existing Apache Parser!

**EVNTTIME(\$MSG, ...)**, which must be defined in the message definition.  
fld7 is used in **EVNTTIME(\$MSG, ...)** but it is not defined anywhere here.

## <MESSAGE

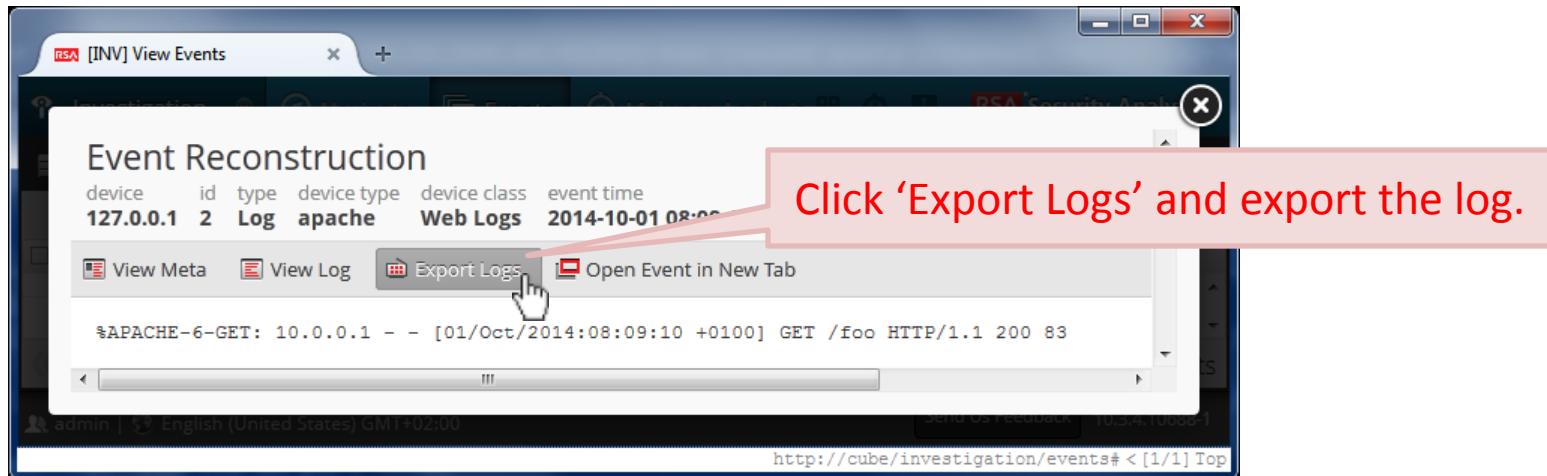
This was an example based on a true story about a custom device parser.



# Missing Metas

Data to provide while creating JIRA ticket when metas are missing:

1. nwtech dump
2. tar -cvzf envision.tar.gz /etc/netwitness/ng/envision
3. The log (Go to Concentrator's Investigate → Events, filter and find the log, click 'View Details' and then click 'Export Logs' to export it. The exported log can be downloaded by going to Profile → Jobs.





# Missing Metas (continued...)

## NetWitness Log Decoder



# Missing Metas

## Log

```
%APACHE-6-HEAD: 10.0.0.1 - - [02/Oct/2014:11:12:13 +0100] HEAD /foo/bar HTTP/1.1 200 0
```

## Message definition in the Apache parser

```
<MESSAGE
    level="6"
    tableid="82"
    parse="1"
    parsedefvalue="1"
    eventcategory="1204000000"
    sitetrack="1"
    id1="00110:01"
    id2="HEAD"
    summary="NIC_B_ADDRESS_ACCOUNTING;"
    content="&lt;saddr&gt; &lt;fld5&gt; &lt;username&gt; [&lt;fld7&gt; &lt;timezone&gt;]
&lt;web_method&gt; &lt;webpage&gt; &lt;network_service&gt; &lt;resultcode&gt; &lt;sbytes&gt;
&lt;@:/*SYSVAL($MSGID,$ID1)&gt;&lt;@event_time:*EVNTTIME($MSG,'%D/%B/%W:%N:%U:%O',fld7)&gt; " />
```

# Missing Metas

The screenshot shows the RSA [INV] View Events window. The top navigation bar includes tabs for Investigation, Navigate, Events, Malware Analysis, and RSA Security Analytics. Below the tabs, there are filters for 'Foo-CONC' and 'Last 3 Hours'. The main pane displays an event detail for a log entry from 2014-10-03T12:04:45. The event type is 'Content.Web Traffic' and it is 153 bytes long. The event details pane lists numerous metadata fields, many of which are collapsed. Some visible fields include sessionid (15), lc.cid (FOO-VLC2), forward.ip (127.0.0.1), device.ip (10.2.0.2), medium (32), device.type (apache), device.class (Web Logs), header.id (0003), level (6), result.code (200), bytes.src (1560), event.time (2014-Oct-01 08:09:10.000), msg.id (00100), event.cat.name (Content.Web Traffic), did (cube), and rid (15). At the bottom of the details pane, there are buttons for 'Hide Additional Meta' and 'View Details'. The footer of the window shows the user is 'admin' in 'English (United States) GMT+00:00', there are 10 events, and the URL is [http://cube/investigation/events# < \[1/1\] Top](http://cube/investigation/events# < [1/1] Top).

The web page '/foo/bar' is missing from the list of metas





# Missing Metas

The web page in the Apache log is parsed as ‘webpage’ field in the ‘apache’ device parser. This is an enVision field name.

enVision field names are mapped to NetWitness meta names in  
</etc/netwitness/ng/envision/etc/table-map.xml>.

Custom mappings may be defined in </etc/netwitness/ng/envision/etc/table-map-custom.xml>.

If there is no mapping for an enVision field, then the value for that field is not written to the disk.

If there is a mapping for an enVision field, but it is flagged as ‘Transient’, then the value for that field is not written to the disk.

**Is there a mapping for enVision field in  
</etc/netwitness/ng/envision/etc/table-map.xml> or  
</etc/netwitness/ng/envision/etc/table-map-custom.xml>?**



# Missing Metas

Is there a mapping for enVision field in `/etc/netwitness/ng/envision/etc/table-map.xml`?

```
# grep 'envisionName="webpage"' /etc/netwitness/ng/envision/etc/table-map.xml
    <mapping envisionName="webpage" nwName="web.page" flags="Transient"
envisionDisplayName="WebPage"/>
```

It is defined, but it is flagged as ‘Transient’!

Is there a mapping for enVision field in `/etc/netwitness/ng/envision/etc/table-map-custom.xml`?

```
# grep 'envisionName="webpage"' /etc/netwitness/ng/envision/etc/table-map-
custom.xml
grep: /etc/netwitness/ng/envision/etc/table-map-custom.xml: No such file or
directoryenvisionDisplayName="WebPage"/>
```

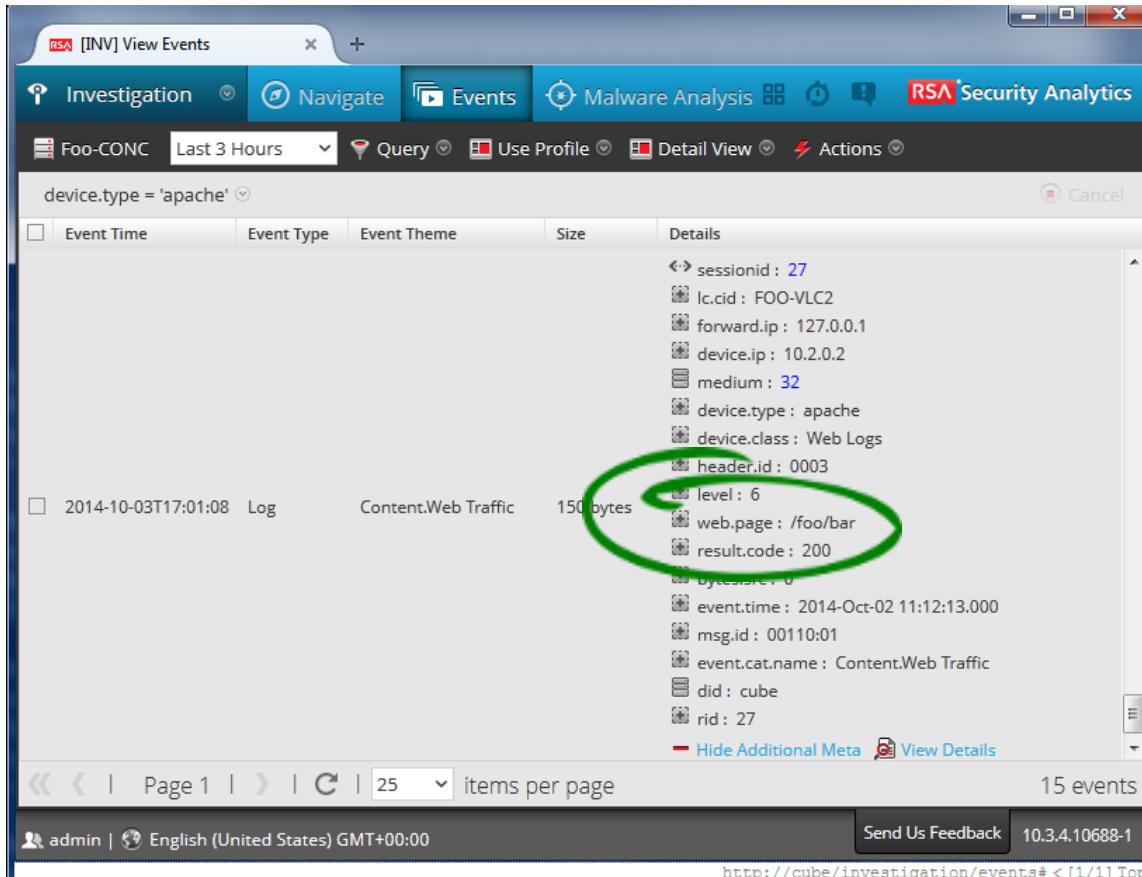
# Missing Metas

If mapping for an enVision field is missing from </etc/netwitness/ng/envision/etc/table-map.xml>, or if it is defined with **flags="Transient"**, the values of such a field is not written to the disk by Log Decoder, and thus not available in the query results.

To make it available, override it by defining it with **flags="None"** in </etc/netwitness/ng/envision/etc/table-map-custom.xml>.

```
<?xml version="1.0" encoding="utf-8"?>
<mappings>
    <mapping envisionName="webpage" nwName="web.page" flags="None"
              envisionDisplayName="WebPage"/>
</mappings>
```

# Missing Metas



The screenshot shows the RSA Security Analytics interface. The top navigation bar includes 'Investigation', 'Navigate', 'Events', 'Malware Analysis', and the 'RSA Security Analytics' logo. Below the navigation is a toolbar with 'Foo-CONC', 'Last 3 Hours', 'Query', 'Use Profile', 'Detail View', and 'Actions'. A search bar displays the query 'device.type = "apache"'. The main content area shows a list of events. One event is selected, showing details: 'Event Time' (2014-10-03T17:01:08), 'Event Type' (Log), 'Event Theme' (Content.Web Traffic), and 'Size' (150 bytes). The event details pane lists numerous metadata fields. A green circle highlights the 'web.page' field, which has a value of '/foo/bar'. Other visible fields include sessionid (27), lc.cid (FOO-VLC2), forward.ip (127.0.0.1), device.ip (10.2.0.2), medium (32), device.type (apache), device.class (Web Logs), header.id (0003), level (6), result.code (200), bytessize (0), event.time (2014-Oct-02 11:12:13.000), msg.id (00110:01), event.cat.name (Content.Web Traffic), did (cube), and rid (27). At the bottom of the details pane are buttons for 'Hide Additional Meta' and 'View Details'. The footer of the interface includes 'Page 1' of 15 events, 'items per page' (set to 25), 'Send Us Feedback', '10.3.4.10688-1', and the URL 'http://cube/investigation/events# < [1/1] Top'. The bottom left corner shows the RSA logo, and the bottom right corner shows the EMC<sup>2</sup> logo.

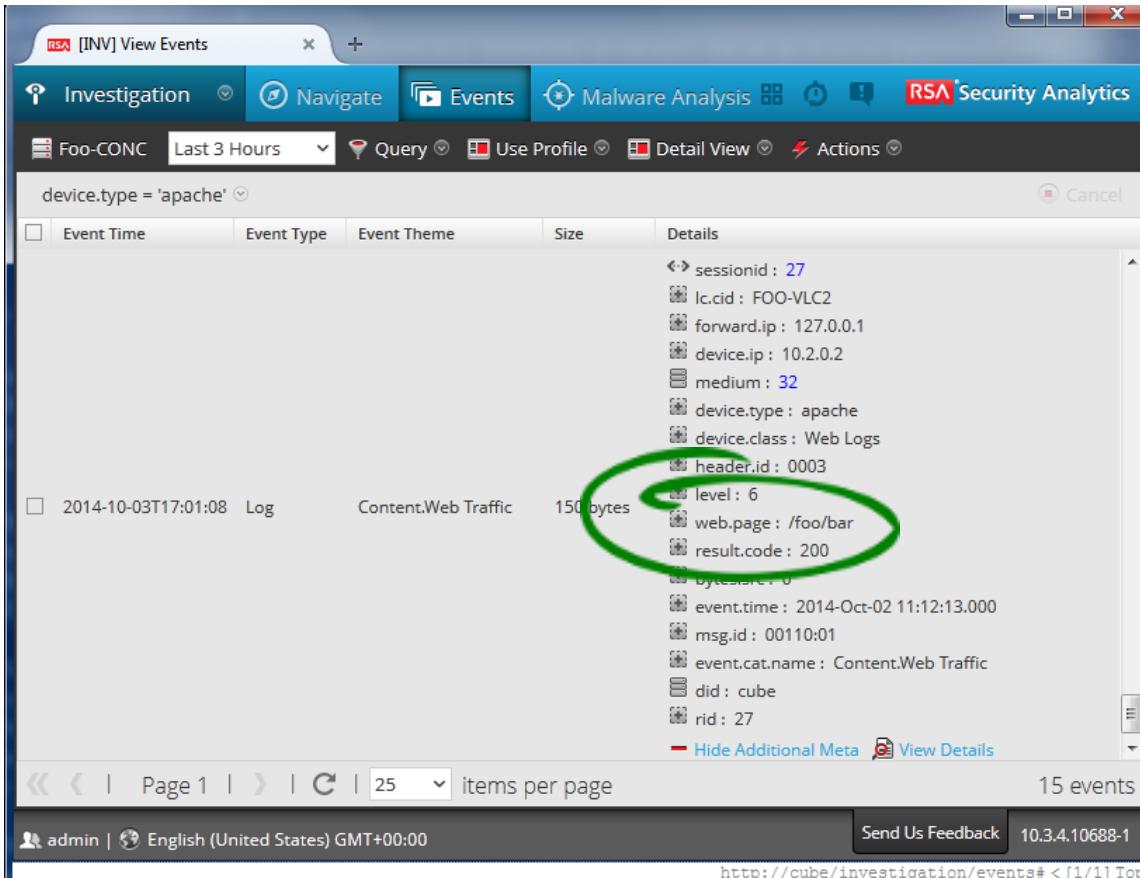
web.page meta is now available



# Meta Query Issues

## NetWitness Log Decoder

# Meta Query Issues



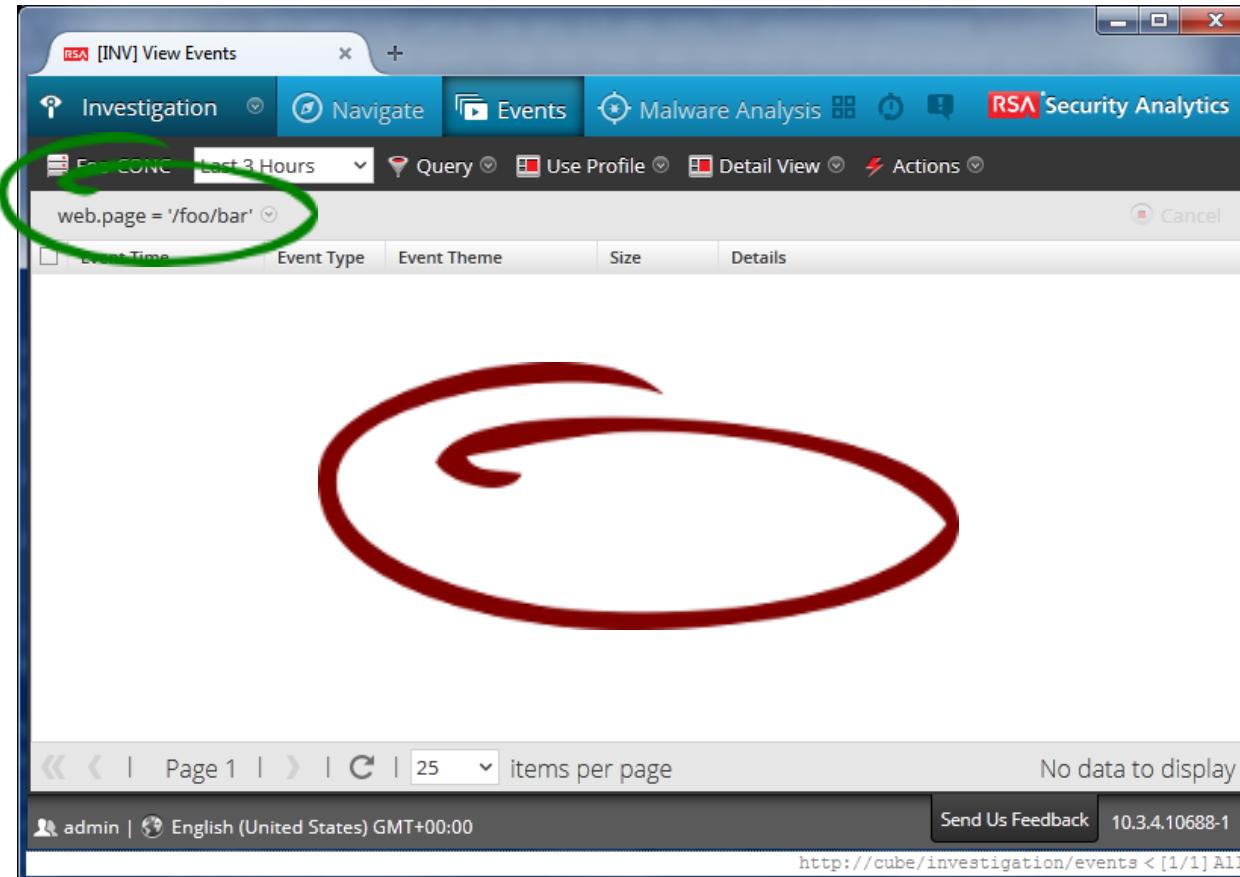
The screenshot shows the RSA Security Analytics interface with the following details:

- Top Bar:** [INV] View Events, Investigation, Navigate, Events, Malware Analysis, RSA Security Analytics.
- Filter Bar:** Foo-CONC, Last 3 Hours, Query, Use Profile, Detail View, Actions.
- Event Details:** device.type = 'apache'  
Event Time: 2014-10-03T17:01:08 Log, Event Type: Content.Web Traffic, Event Theme: 150 bytes.  
Details:
  - sessionid : 27
  - lc.cid : FOO-VLC2
  - forward.ip : 127.0.0.1
  - device.ip : 10.2.0.2
  - medium : 32
  - device.type : apache
  - device.class : Web Logs
  - header.id : 0003
  - level : 6
  - web.page : /foo/bar (This field is circled in green.)
  - result.code : 200
  - byesize : 0
  - event.time : 2014-Oct-02 11:12:13.000
  - msg.id : 00110:01
  - event.cat.name : Content.Web Traffic
  - did : cube
  - rid : 27
- Bottom Navigation:** Page 1, items per page (25), 15 events.
- User Information:** admin | English (United States) GMT+00:00, Send Us Feedback, 10.3.4.10688-1, http://cube/investigation/events# < [1/1] Top.

web.page meta is available...



# Meta Query Issues



... but we cannot use it in a query



# Meta Query Issues

If a meta is available (i.e. we can see the meta value in Concentrator's Investigation → Events page) but we cannot use it in a query to filter results in Concentrator, then it is not a Log Decoder issue.

Log Decoder parsed the message and provided the meta. Log Decoder's job is done. This is a Concentrator indexing issue that needs to be fixed on Concentrator.



# Meta Query Issues

The metas to be indexed are defined in </etc/netWitness/ng/index-concentrator.xml>.

Custom indexing may be defined in </etc/netWitness/ng/index-concentrator-custom.xml>.

If there is no indexing defined for a meta key, or

If the index level for a meta key is defined as ‘IndexNone’, then the meta is NOT indexed.

If the index level for a meta key is defined as ‘IndexKeys’ or ‘IndexValues’, then the meta is indexed.

**Is index level defined as ‘IndexKeys’ or ‘IndexValues’ for the meta in**

**</etc/netWitness/ng/index-concentrator.xml> or**

**</etc/netWitness/ng/index-concentrator-custom.xml>?**



# Meta Query Issues

Is indexing enabled for the meta in /etc/netWitness/ng/index-concentrator.xml?

```
# grep 'name="web.page"' /etc/netWitness/ng/index-concentrator.xml  
#
```

Is indexing enabled for the meta in /etc/netWitness/ng/index-concentrator-custom.xml?

```
# grep 'name="web.page"' /etc/netWitness/ng/index-concentrator-  
custom.xml  
grep: /etc/netWitness/ng/index-concentrator-custom.xml: No such file or  
directory
```

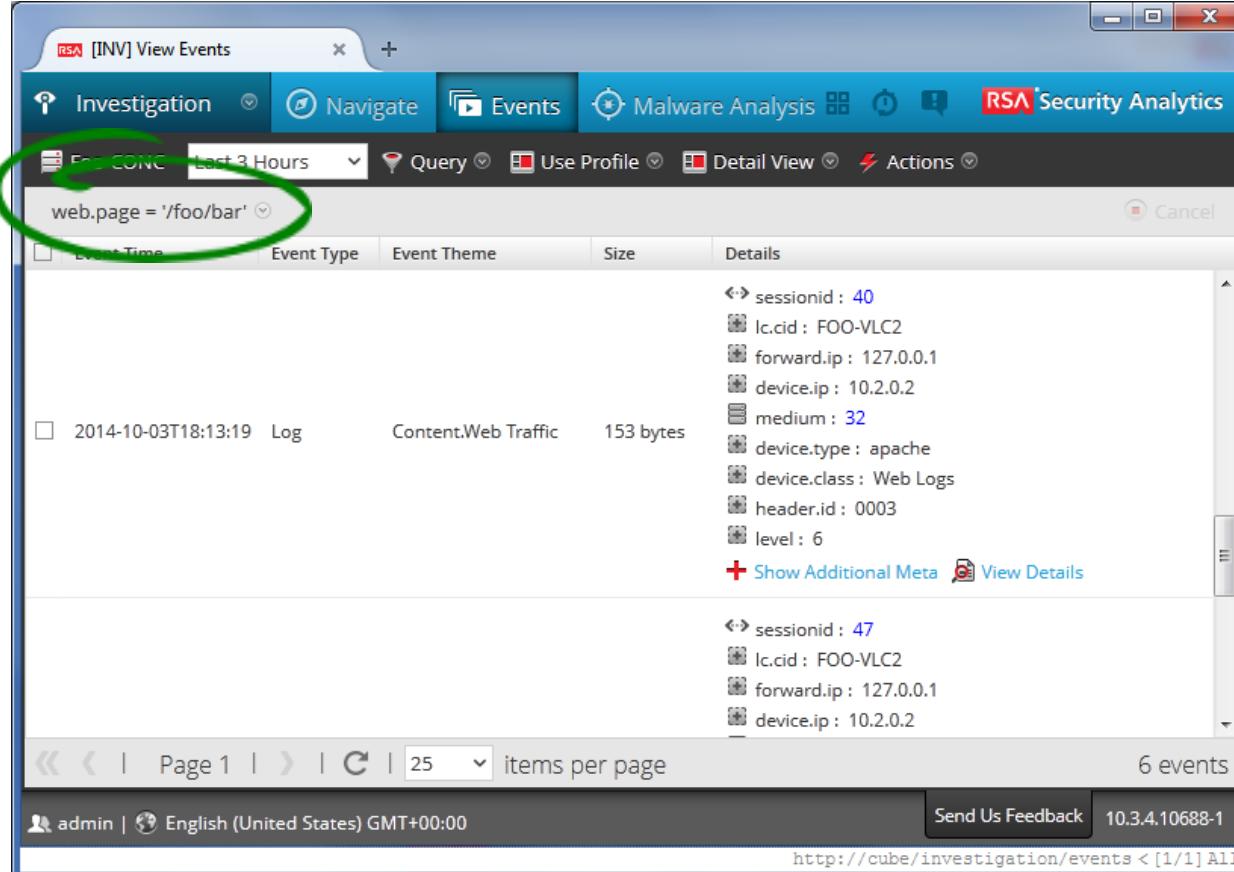
No, indexing is not enabled for the meta.

# Meta Query Issues

We can enable indexing for a meta by adding a definition for it in etc/netwitness/ng/index-concentrator-custom.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<language level="IndexNone" defaultAction="Auto">
    <key description="Web Page" level="IndexValues" name="web.page"
        format="Text" valueMax="10000" />
</language>
```

# Meta Query Issues



The screenshot shows the RSA Security Analytics interface. At the top, there's a navigation bar with tabs for Investigation, Navigate, Events, Malware Analysis, and Security Analytics. Below the navigation bar is a toolbar with various icons and dropdown menus. A search bar at the top left contains the query "web.page = '/foo/bar'". A green circle highlights this search term. The main area displays a table of event results. The first event listed is from October 3, 2014, at 18:13:19, categorized as Content.Web Traffic, with a size of 153 bytes. The event details show sessionid: 40, lc.cid: FOO-VLC2, forward.ip: 127.0.0.1, device.ip: 10.2.0.2, medium: 32, device.type: apache, device.class: Web Logs, header.id: 0003, and level: 6. There are links to "Show Additional Meta" and "View Details". Below this, another event is listed with sessionid: 47, lc.cid: FOO-VLC2, forward.ip: 127.0.0.1, and device.ip: 10.2.0.2. At the bottom of the interface, there are pagination controls (Page 1), a dropdown for items per page (set to 25), and a status bar showing "6 events". The footer includes a "Send Us Feedback" button, the IP address 10.3.4.10688-1, and the URL http://cube/investigation/events < [1/1] All.

Querying web.page meta works after defining its index level



# Log Forwarding

## NetWitness Log Decoder



# Log Forwarding

Since 10.3.2, Log Decoder can forward logs that it ingests to another destination.

Log Forwarding is done after the logs are parsed and before they are written to the disk.

Three steps must be performed to enable log forwarding:

1. Define a destination
2. Enable syslog forwarding
3. Define an application rule

# Log Forwarding

The screenshot shows the RSA [ADM] "FOO-LD" explorer interface. The left sidebar shows a tree view with "FOO-LD (LOG\_DECODER)" selected, revealing sub-folders like "connections", "database", "decoder", "config" (which is highlighted), "devices", and "parsers". The main pane displays a table of configuration parameters:

/Decoder/config	FOO-LD (Log Decoder)
export.time.ordered	no
export.usage.max	00
logs.forwarding.destination	bar=tcp:10.31.244.44:5000
logs.forwarding.enabled	false
logs.stats.enabled	true

A green oval highlights the "logs.forwarding.destination" row. The bottom status bar shows the user is "admin", the language is "English (United States) GMT+00:00", and the IP address is "10.3.4.10688-1". The URL "http://cube/admin/devices/1/explorer < [1/1] All" is also visible.

Step 1: Define a destination

# Log Forwarding

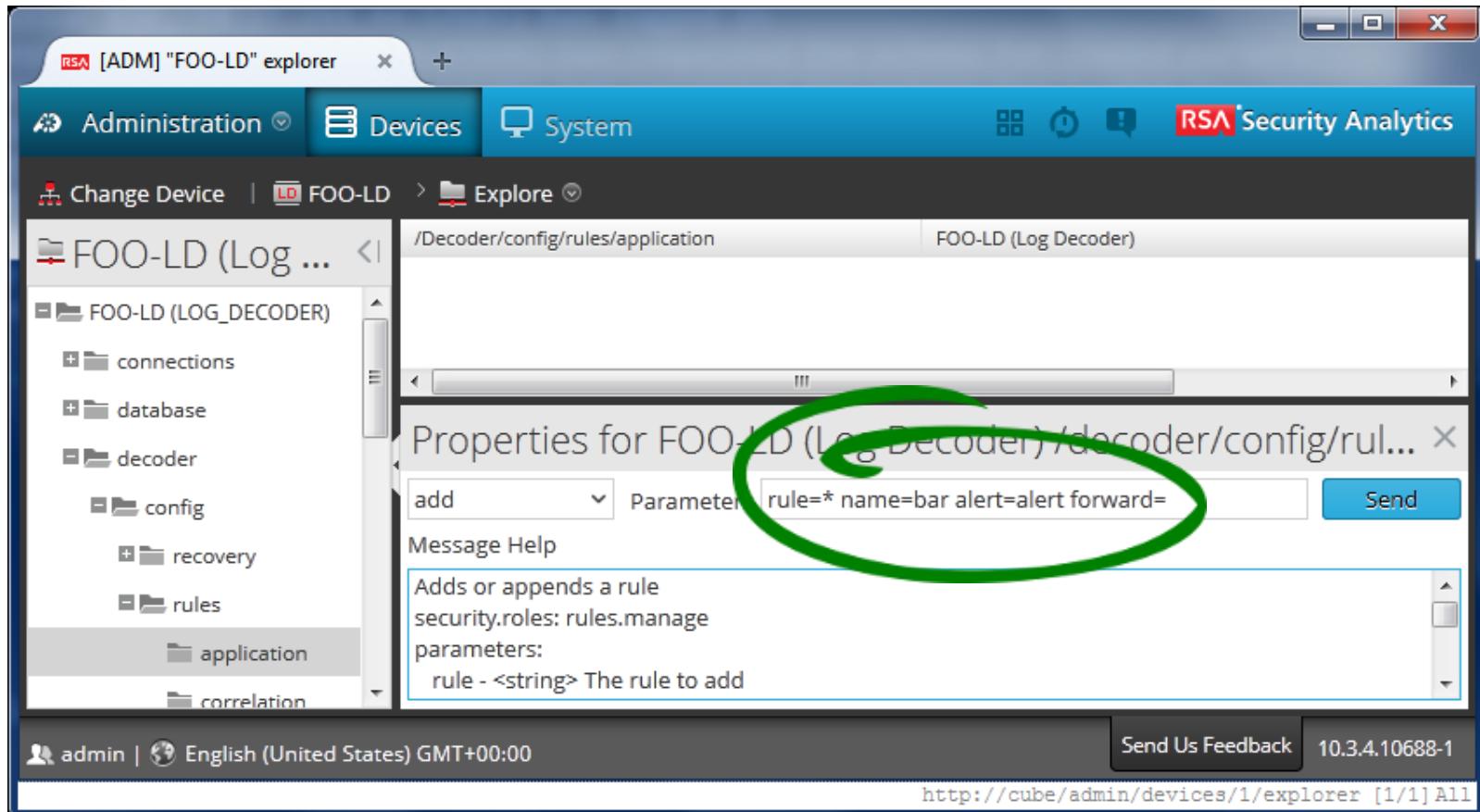
The screenshot shows the RSA Security Analytics administration interface. The top navigation bar includes 'Administration', 'Devices' (selected), and 'System'. The main pane displays the configuration for 'FOO-LD (Log Decoder)'. The left sidebar shows a tree view with 'FOO-LD (LOG\_DECODER)' expanded, revealing 'connections', 'database', 'decoder', 'config' (which is selected), 'devices', and 'parsers'. The right pane lists configuration parameters:

/Decoder/config	FOO-LD (Log Decoder)
export.time.ordered	no
export.usage.max	90
logs.forwarding.destination	bar=tcp:10.31.244.44:5000
logs.forwarding.enabled	true
logs.stats.enabled	true

A green oval highlights the 'true' value for 'logs.forwarding.enabled'. The bottom status bar shows the user is 'admin', the language is 'English (United States) GMT+00:00', and the IP address is '10.3.4.10688-1'. The URL 'http://cube/admin/devices/1/explorer < [1/1] All' is also visible.

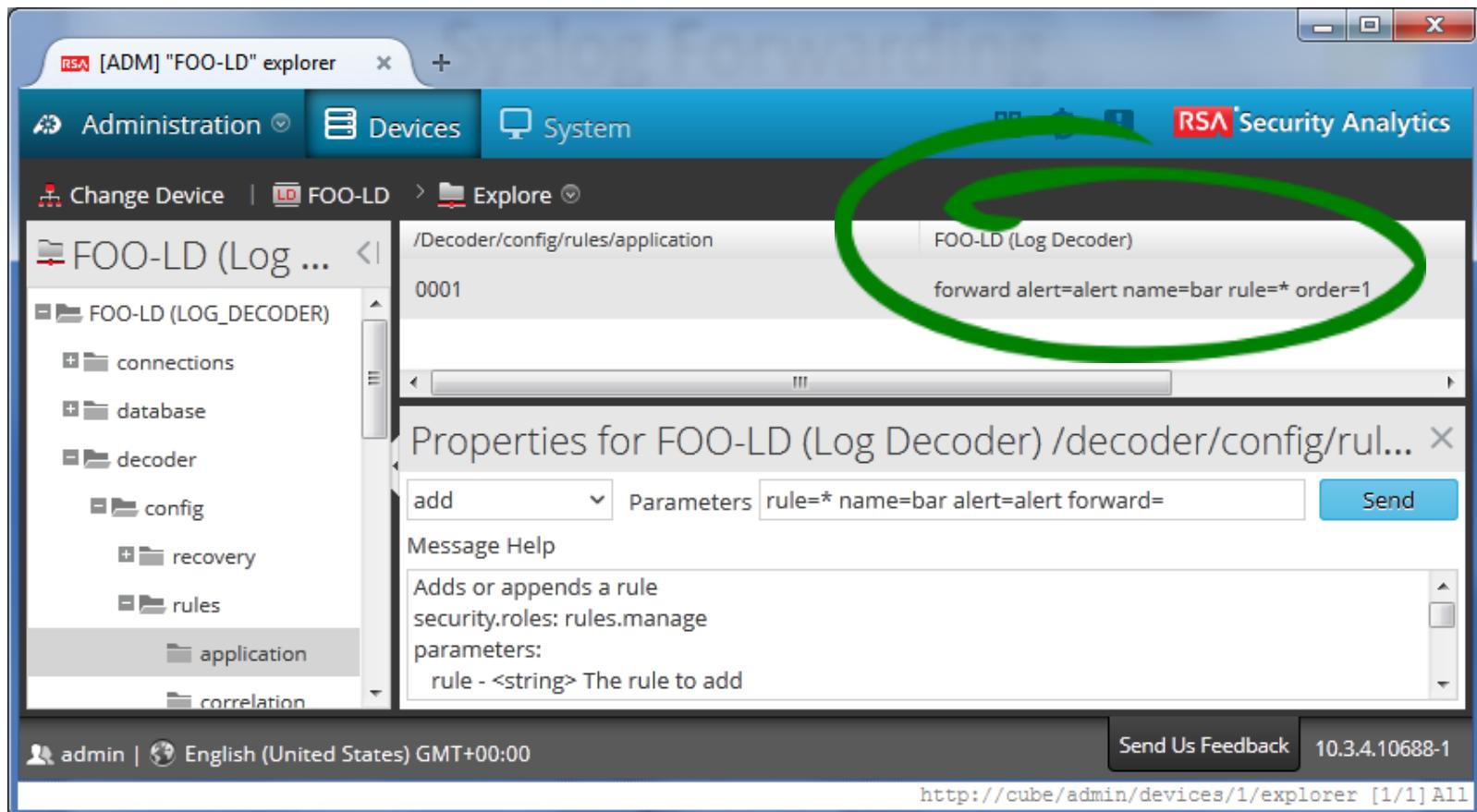
Step 2: Enable syslog forwarding

# Log Forwarding



Step 3: Add an application rule

# Log Forwarding



Configuration complete



# Log Forwarding

We feed two Cisco Pix logs to Log Decoder.

```
# echo "<1> %PIX-1-101001: (PRIORITY) Failover cable OK." > /dev/tcp/127.0.0.1/514
# echo "<1> %PIX-1-101001: (PRIORITY) Bad failover cable." > /dev/tcp/127.0.0.1/514
```

(The above command is run on Log Decoder.)

We see both logs forwarded to the destination

```
# nc -kl 5000
47 <1>%PIX-1-101001: (PRIORITY) Failover cable OK.
48 <1>%PIX-1-101001: (PRIORITY) Bad failover cable.
```

(The above netcat command is run on the destination. It listens on TCP port 5000 and prints all payloads it receives on this port.)



# Log Forwarding

The screenshot shows the RSA Security Analytics interface. The top navigation bar includes tabs for Investigation, Navigate, Events, Malware Analysis, and Security Analytics. Below the navigation is a search bar with the query "device.type = 'ciscopix'". The main content area displays a list of events. One event is selected, showing detailed metadata. A green oval highlights the "alert: bar" entry in the metadata list.

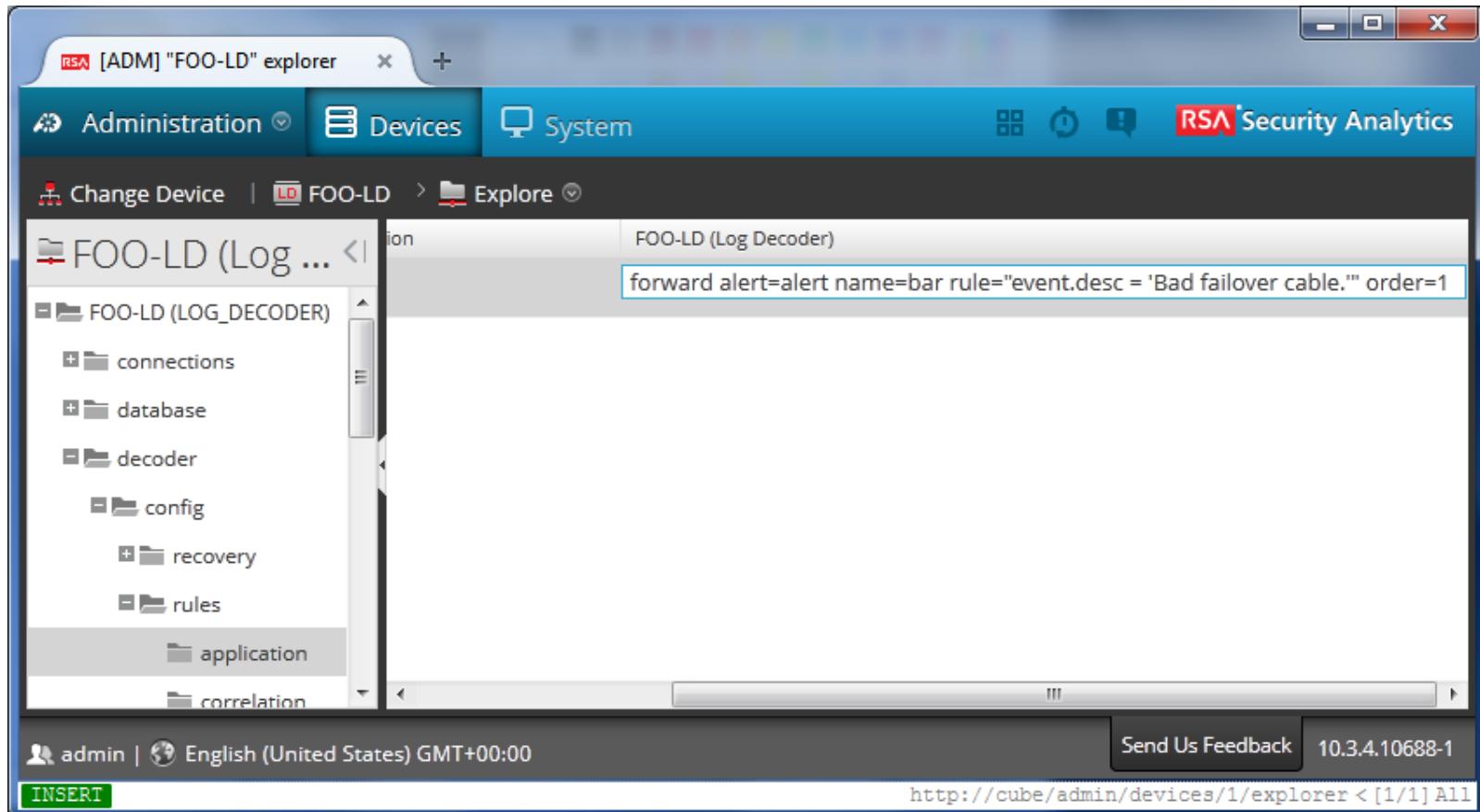
Event Time	Event Type	Event Theme	Size	Details
2014-10-04T20:37:40	Log	System.Heartbeats	85 bytes	<ul style="list-style-type: none"><li>↳ sessionid : 1</li><li>↳ device.ip : 127.0.0.1</li><li>↳ medium : 32</li><li>↳ device.type : ciscopix</li><li>↳ device.class : Firewall</li><li>↳ header.id : 0001</li><li>↳ level : 1</li><li>↳ event.desc : Failover cable OK.</li><li>↳ msg.id : 101001</li><li>↳ event.cat.name : System.Heartbeats</li><li><b>⚠ alert: bar</b></li><li>↳ did : cube</li><li>↳ rid : 1</li></ul> <p><a href="#">Hide Additional Meta</a> <a href="#">View Details</a></p>

Items per page: 25 | Page 1 | 13 events

admin | English (United States) GMT+00:00 | Send Us Feedback | 10.3.4.10688-1 | [http://cube/investigation/events# < \[1/1\] Top](http://cube/investigation/events# < [1/1] Top)

Log forwarding has led to the creation of a new meta called 'alert'

# Log Forwarding



Forward logs with event.desc = 'Bad failover cable.'





# Log Forwarding

We feed two Cisco Pix logs to Log Decoder.

```
# echo "<1> %PIX-1-101001: (PRIORITY) Failover cable OK." > /dev/tcp/127.0.0.1/514
# echo "<1> %PIX-1-101001: (PRIORITY) Bad failover cable." > /dev/tcp/127.0.0.1/514
```

(The above command is run on Log Decoder.)

We see only one log forwarded to the destination, the one that matches the rule:

`event.desc = 'Bad failover cable.'`

```
# nc -kl 5000
48 <1>%PIX-1-101001: (PRIORITY) Bad failover cable.
```

(The above netcat command is run on the destination. It listens on TCP port 5000 and prints all payloads it receives on this port.)

# Log Forwarding

The screenshot shows the RSA Security Analytics interface with the title bar "RSA [INV] View Events". The top navigation bar includes "Investigation", "Events", "Malware Analysis", and "Security Analytics". Below the navigation is a search bar with the query "device.type = 'ciscopix'". The main content area displays a table of events. One event is selected, showing details such as sessionid: 3, device.ip: 127.0.0.1, medium: 32, device.type: ciscopix, device.class: Firewall, header.id: 0001, level: 1, event.desc: Failover cable OK., msg.id: 101001, event.cat.name: System.Heartbeats, did: cube, rid: 3. A link to "View Details" is also present. At the bottom, there are pagination controls (Page 1, 25 items per page), a footer with "admin | English (United States) GMT+00:00", "Send Us Feedback", and the IP address "10.3.4.10688-1", and a URL "http://cube/investigation/events# < [1/1] Top".

Now there is no 'alert' meta for the log that did not match the rule

# Log Forwarding

The screenshot shows the RSA Security Analytics interface with the title bar "RSA [INV] View Events". The top navigation bar includes "Investigation", "Events", "Malware Analysis", and "Security Analytics". Below the navigation is a search bar with the query "device.type = 'ciscopix'". The main content area displays a table of events. One event is selected, showing details: "Event Time" (2014-10-04T21:12:22), "Event Type" (Log), "Event Theme" (System.Heartbeats), and "Size" (86 bytes). The event details pane shows the following fields:  
↳ sessionid : 4  
↳ device.ip : 127.0.0.1  
↳ medium : 32  
↳ device.type : ciscopix  
↳ device.class : Firewall  
↳ header.id : 0001  
↳ level : 1  
↳ event.desc : Bad failover cable.  
↳ msg.id : 101001  
↳ event.cat.name : System.Heartbeats  
⚠ alert : bar  
↳ did : cube  
↳ rid : 4  
A green oval highlights the "event.desc" field, and another green oval highlights the "alert" field. At the bottom of the event details pane are buttons for "Hide Additional Meta" and "View Details".  
At the bottom of the interface, there are navigation controls (back, forward, page numbers 1-25), a "items per page" dropdown set to 25, and a status bar with "4 events", "admin | English (United States) GMT+00:00", "Send Us Feedback", "10.3.4.10688-1", and the URL "http://cube/investigation/events# < [1/1] Top".

Now there is 'alert' meta only for the log that matched the rule

# Log Forwarding

The screenshot shows the RSA Security Analytics administration interface. The title bar reads "rsa [ADM] "FOO-LD" explorer". The top navigation bar includes "Administration", "Devices", and "System" tabs, along with icons for search, refresh, and help, and the "RSA Security Analytics" logo.

The main pane displays a tree view under "FOO-LD (LOG\_DECODER)" with the following structure:

- + connections
- + database
- decoder
  - config
    - + recovery
    - rules
  - application
  - correlation

A tooltip for the "decoder" node shows the configuration: "FOO-LD (Log Decoder) transient forward alert=alert name=bar rule='event.desc = 'Bad failover cable.''"

The bottom status bar shows the user "admin", the language "English (United States) GMT+00:00", a feedback link "Send Us Feedback", the IP address "10.3.4.10688-1", and the URL "http://cube/admin/devices/1/explorer < [1/1] All". A green "INSERT" button is also visible.

Use 'transient' flag, so that the meta is not written to the disk





# Log Forwarding

We feed two Cisco Pix logs to Log Decoder.

```
# echo "<1> %PIX-1-101001: (PRIORITY) Failover cable OK." > /dev/tcp/127.0.0.1/514
# echo "<1> %PIX-1-101001: (PRIORITY) Bad failover cable." > /dev/tcp/127.0.0.1/514
```

(The above command is run on Log Decoder.)

Just like before, we see only one log forwarded to the destination, the one that matches the rule: `event.desc = 'Bad failover cable.'`

```
# nc -kl 5000
48 <1>%PIX-1-101001: (PRIORITY) Bad failover cable.
```

(The above netcat command is run on the destination. It listens on TCP port 5000 and prints all payloads it receives on this port.)



# Log Forwarding

The screenshot shows the RSA Security Analytics interface. The top navigation bar includes 'Investigation', 'Events' (selected), 'Malware Analysis', and 'RSA Security Analytics'. Below the navigation is a search bar with 'FOO-CONC' and 'Last 3 Hours'. The main content area displays a table with a single row of event details:

	Event Time	Event Type	Event Theme	Size	Details
<input type="checkbox"/>	2014-10-04T21:47:42	Log	System.Heartbeats	86 bytes	<a href="#">Show Additional Meta</a> <a href="#">View Details</a> sessionid : 6 device.ip : 127.0.0.1 medium : 32 device.type : ciscopix device.class : Firewall header.id : 0001 level : 1 event.desc : Bad failover cable. msg.id : 101001 event.cat.name : System.Heartbeats did : cube rid : 6

The 'event.desc' field, which contains the value 'Bad failover cable.', is circled in green. At the bottom of the interface, there are navigation controls for pages, items per page (set to 25), and a link to 'Send Us Feedback'. The footer includes the URL 'http://cube/investigation/events# < [1/1] Top'.

No 'alert' meta for the log that matched the rule because it had 'transient' flag set





# Log Forwarding Issues

## NetWitness Log Decoder



# Log Forwarding Issues

Check if the alert meta is being generated for the logs that we expect to be forwarded.

Note: If there is ‘transient’ flag in the application rule, then the alert meta won’t be written to the disk, and thus cannot be seen in the Investigation module in the UI.

Therefore, to ease troubleshooting, remove ‘transient’ flag (if it is present) and check if the alert meta is being generated for the logs that we expect to be forwarded.

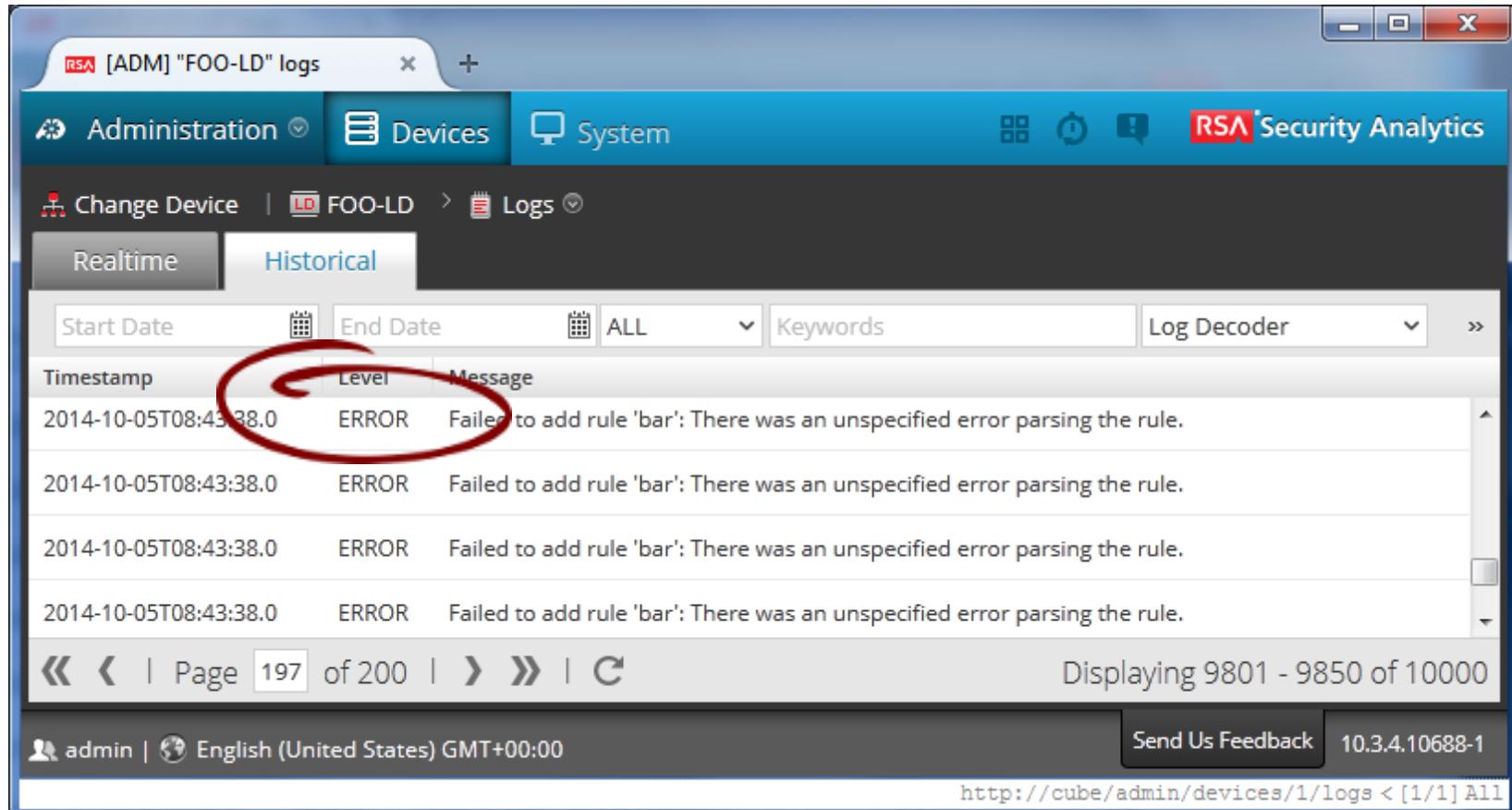


# Log Forwarding Issues

If the ‘alert’ meta is not being generated for the logs that we expect to be forwarded, then there is an issue with the rule.

1. Check the logs.
2. Check the rule.

# Log Forwarding Issues



The screenshot shows a log viewer interface for the RSA Security Analytics platform. The top navigation bar includes tabs for Administration, Devices, and System, along with a search bar and a date range selector. The main content area displays a table of log entries. A red circle highlights the 'Level' column header, which is set to 'ERROR'. The log entries show four identical errors occurring at the same timestamp (2014-10-05T08:43:38.0) with the message 'Failed to add rule 'bar': There was an unspecified error parsing the rule.'.

Timestamp	Level	Message
2014-10-05T08:43:38.0	ERROR	Failed to add rule 'bar': There was an unspecified error parsing the rule.
2014-10-05T08:43:38.0	ERROR	Failed to add rule 'bar': There was an unspecified error parsing the rule.
2014-10-05T08:43:38.0	ERROR	Failed to add rule 'bar': There was an unspecified error parsing the rule.
2014-10-05T08:43:38.0	ERROR	Failed to add rule 'bar': There was an unspecified error parsing the rule.

Check the logs



# Log Forwarding Issues

The screenshot shows the RSA Security Analytics interface. The top navigation bar includes tabs for Administration, Devices, and System, along with the RSA logo and 'Security Analytics' text. The main content area displays a tree view under 'FOO-LD (Log Decoder)' with nodes for connections, database, decoder, config, recovery, and rules. A right-hand panel shows the configuration for the 'decoder' node, specifically the 'rules' section. The rule listed is:

```
forward alert=alert name=bar rule=event.desc = 'Bad failover cable.' order=1
```

A red callout box highlights the rule condition 'event.desc = 'Bad failover cable.'' with the text: "The rule condition \"event.desc = 'Bad failover cable.'\" should be enclosed in double-quotes."

The bottom status bar shows the user is 'admin', the language is 'English (United States) GMT+00:00', and the IP address is '10.3.4.10688-1'. The URL 'http://cube/admin/devices/1/explorer < [1/1] All' is also visible.

Check the rule



# Log Forwarding Issues

The screenshot shows the RSA Security Analytics interface. The title bar reads "rsa [ADM] "FOO-LD" explorer". The top navigation bar includes "Administration", "Devices" (selected), and "System" tabs, along with icons for user, system status, and help, and the "RSA Security Analytics" logo.

The main content area shows a tree view under "FOO-LD (Log Decoder) <". The tree structure includes:

- FOO-LD (LOG\_DECODER)
  - connections
  - database
  - decoder
    - config
      - recovery
    - rules
- application

A right-hand panel displays the configuration for "FOO-LD (Log Decoder)" with the following rule:

```
forward alert=alert name=bar rule="event.desc = 'Bad failover cable.'" order=1
```

The bottom navigation bar includes "admin", "English (United States) GMT+00:00", "Send Us Feedback", the IP address "10.3.4.10688-1", and the URL "http://cube/admin/devices/1/explorer < [1/1] All".

Fix the rule





# Log Forwarding Issues

One of the most common errors encountered in the field: The ‘alert’ flag is not set.

Both ‘forward’ and ‘alert’ flags must be set.

1. The ‘alert’ flag is necessary to generate the ‘alert’ meta that is required for forwarding rules.
2. The ‘forward’ flag is necessary to set the ‘forward’ flag on any log that matches the rule condition.

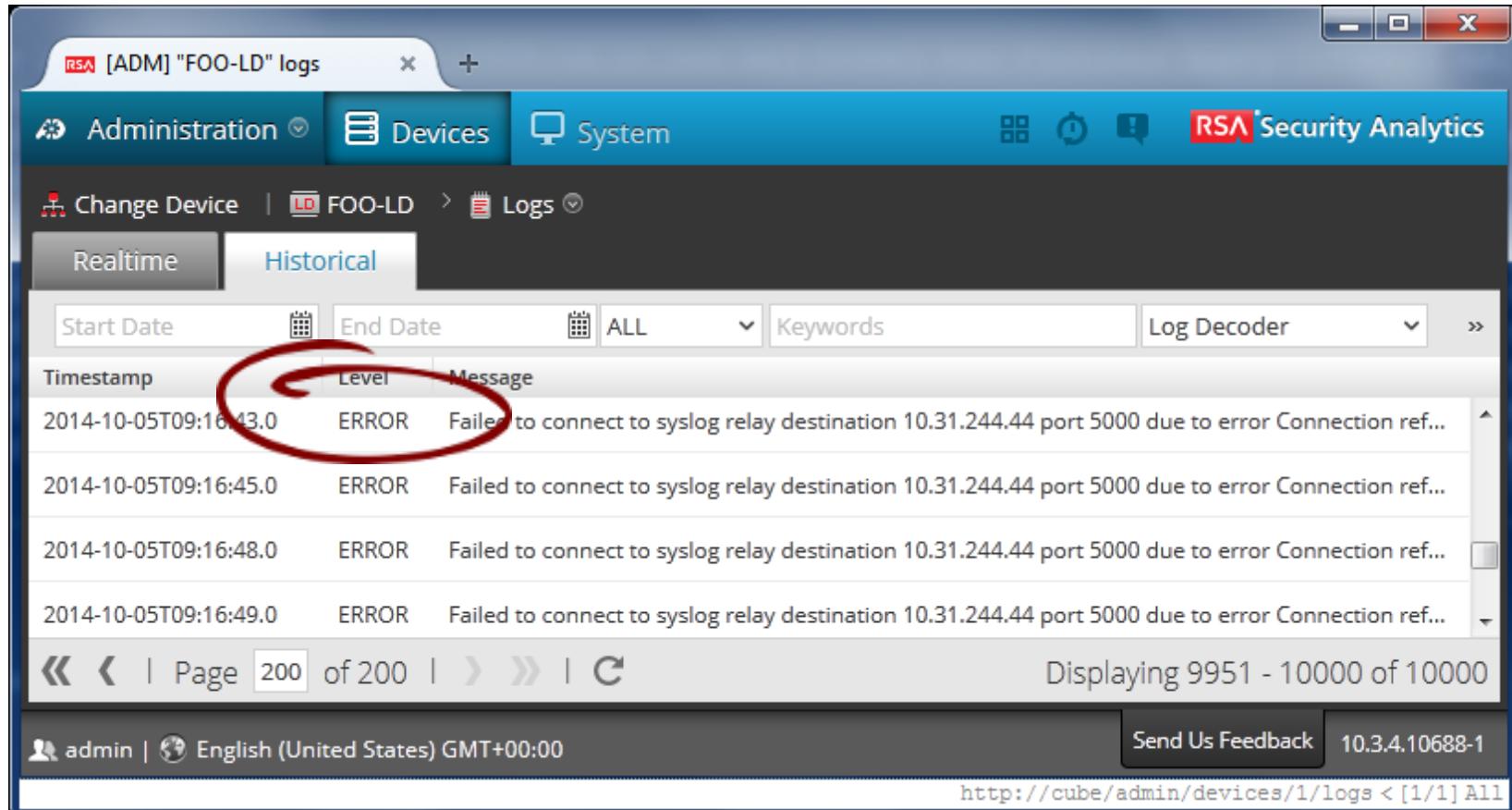


# Log Forwarding Issues

If the rule is good, both ‘forward’ and ‘alert’ flags are set, and the ‘alert’ meta is being generated for the logs that should be forwarded, then check connectivity to the log forwarding destination.

1. Check the logs
2. Test connectivity to the destination

# Log Forwarding Issues



The screenshot shows the RSA Security Analytics interface for device FOO-LD. The 'Logs' tab is selected, and the 'Historical' view is active. A red circle highlights the 'Level' column in the log table, which shows multiple entries of 'ERROR' level. The log table includes columns for Timestamp, Level, and Message. The messages indicate failed attempts to connect to a syslog relay destination at 10.31.244.44 port 5000 due to connection errors.

Timestamp	Level	Message
2014-10-05T09:16:43.0	ERROR	Failed to connect to syslog relay destination 10.31.244.44 port 5000 due to error Connection ref...
2014-10-05T09:16:45.0	ERROR	Failed to connect to syslog relay destination 10.31.244.44 port 5000 due to error Connection ref...
2014-10-05T09:16:48.0	ERROR	Failed to connect to syslog relay destination 10.31.244.44 port 5000 due to error Connection ref...
2014-10-05T09:16:49.0	ERROR	Failed to connect to syslog relay destination 10.31.244.44 port 5000 due to error Connection ref...

Check the logs



# Log Forwarding Issues

Test connectivity to the log forwarding destination.

In Bash, and any POSIX shell, `:` is a null command. It does nothing. Therefore, we can use the `:` command instead of the echo command if we just want to test connectivity to the log forwarding destination without sending any payload to it.

```
# : > /dev/tcp/10.31.244.44/5000  
#
```

(The above command is run on Log Decoder to test connectivity to the log forwarding destination.)

If the connection could be established, then there is no output.

```
# : > /dev/tcp/10.31.244.44/5000  
-bash: connect: Connection refused  
-bash: /dev/tcp/10.31.244.44/5000: Connection refused
```

(The above command is run on Log Decoder to test connectivity to the log forwarding destination.)

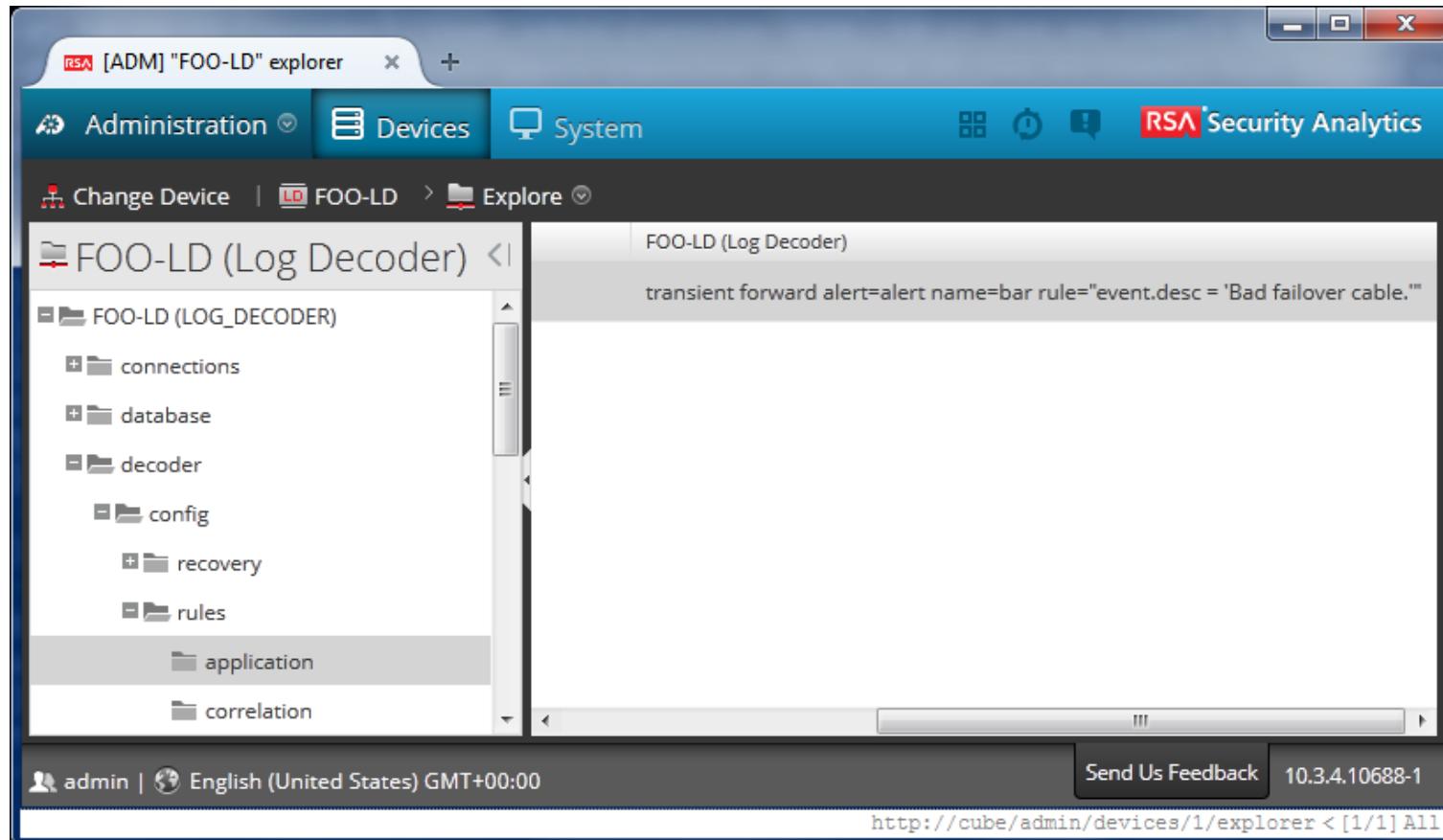
If the connection could not be established, then error messages are displayed.



# Log Forwarding Flag Issues

## NetWitness Log Decoder

# Log Forwarding Flag Issues



We have a properly configured log forwarding rule here ...

# Log Forwarding Flag Issues

The screenshot shows the RSA Security Analytics web interface. The title bar reads "RSA [ADM] 'FOO-LD' config". The navigation bar includes "Administration", "Devices" (selected), and "System". The main pane shows a device named "FOO-LD" under "Config". A sub-menu for "App Rules" is open, showing actions like "Import", "Export", "Push", and "History". The status bar at the bottom indicates the user is "admin" in English (United States) GMT+00:00, and the URL is "http://cube/admin/devices/1/config < [1/1] All".

... but, in 10.3.4, if we perform any action, the 'forward' and 'transient' flags are lost



# Log Forwarding Flag Issues

The screenshot shows the RSA Security Analytics web interface. The title bar reads "rsa [ADM] "FOO-LD" config". The top navigation bar includes "Administration", "Devices" (selected), and "System". On the right, there are icons for user, system status, and help, along with the "RSA Security Analytics" logo.

The main content area shows a breadcrumb path: "Change Device" > "FOO-LD" > "Config". Below this is a navigation menu with tabs: General, Files, App Rules (selected), Correlation Rules, Feeds, Parsers, and Appliance Service.

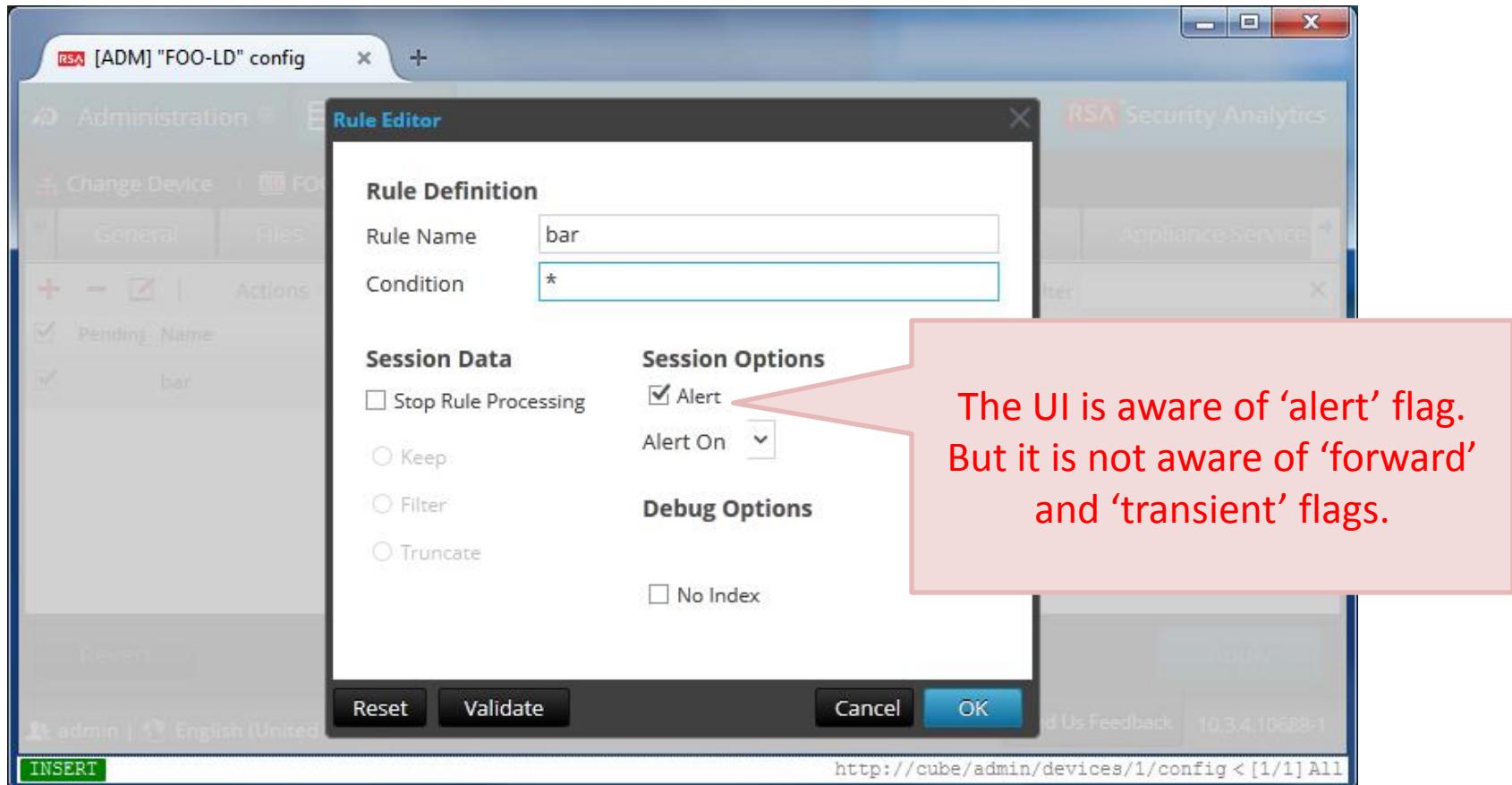
The "App Rules" tab is active, displaying a table with one row. The table columns are: Pending Name, Condition, Session Data, Alert, and Status. The row contains "Pending Name" (checkbox checked), "Condition" (checkbox checked, value "event.desc = 'Bad failover cable.'"), "Session Data" (checkbox checked), "Alert" (checkbox checked, value "alert"), and "Status" (green icon).

At the bottom left are "Revert" and "Apply" buttons. At the bottom right are links for "Send Us Feedback" and the IP address "10.3.4.10688-1". The URL in the address bar is "http://cube/admin/devices/1/config < [1/1] All".

In 10.3.4, editing the rule also causes the 'forward' and 'transient' flags to be lost

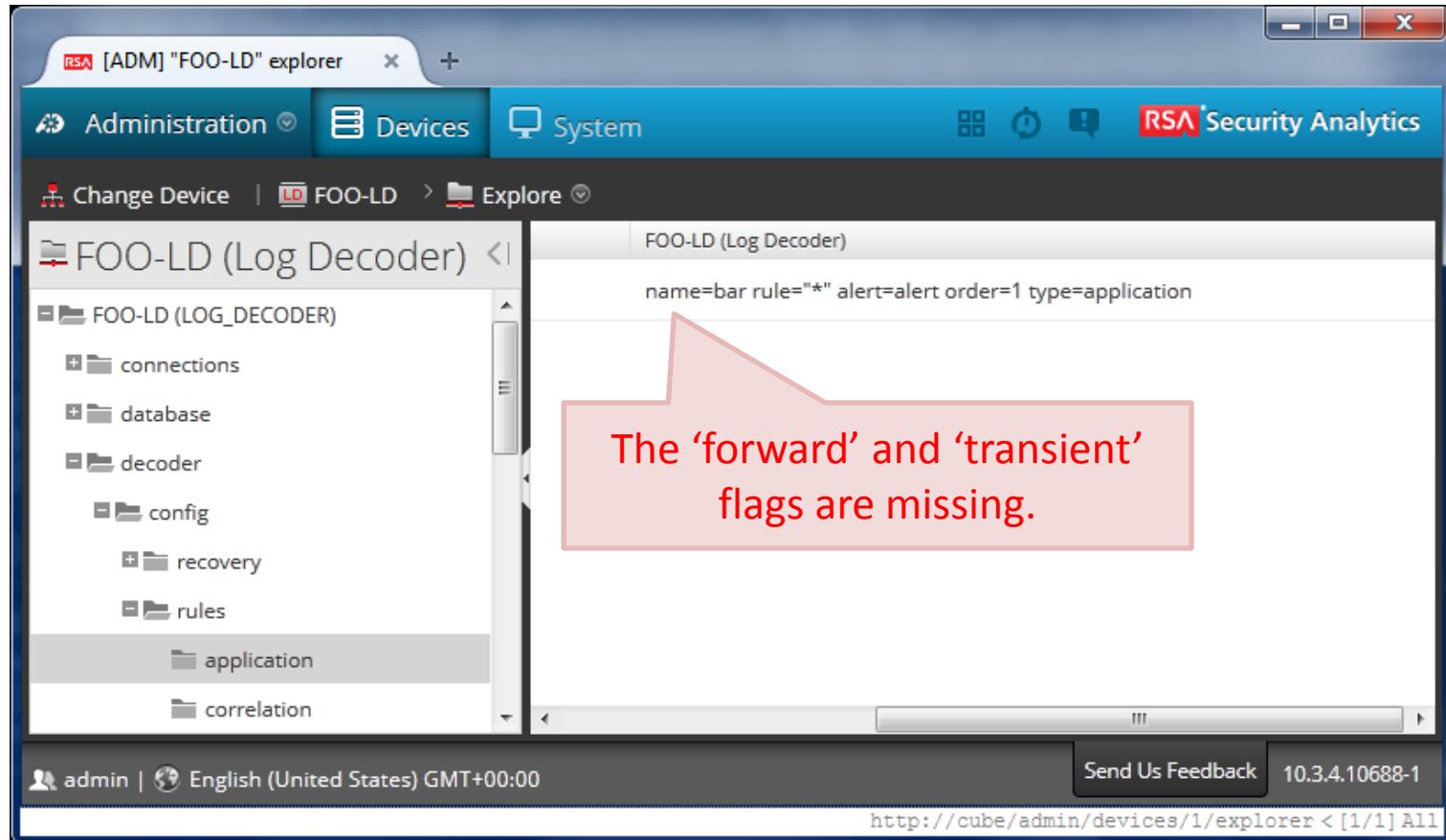


# Log Forwarding Flag Issues



In 10.3.4, the UI is not aware of the 'forward' and 'transient' flags

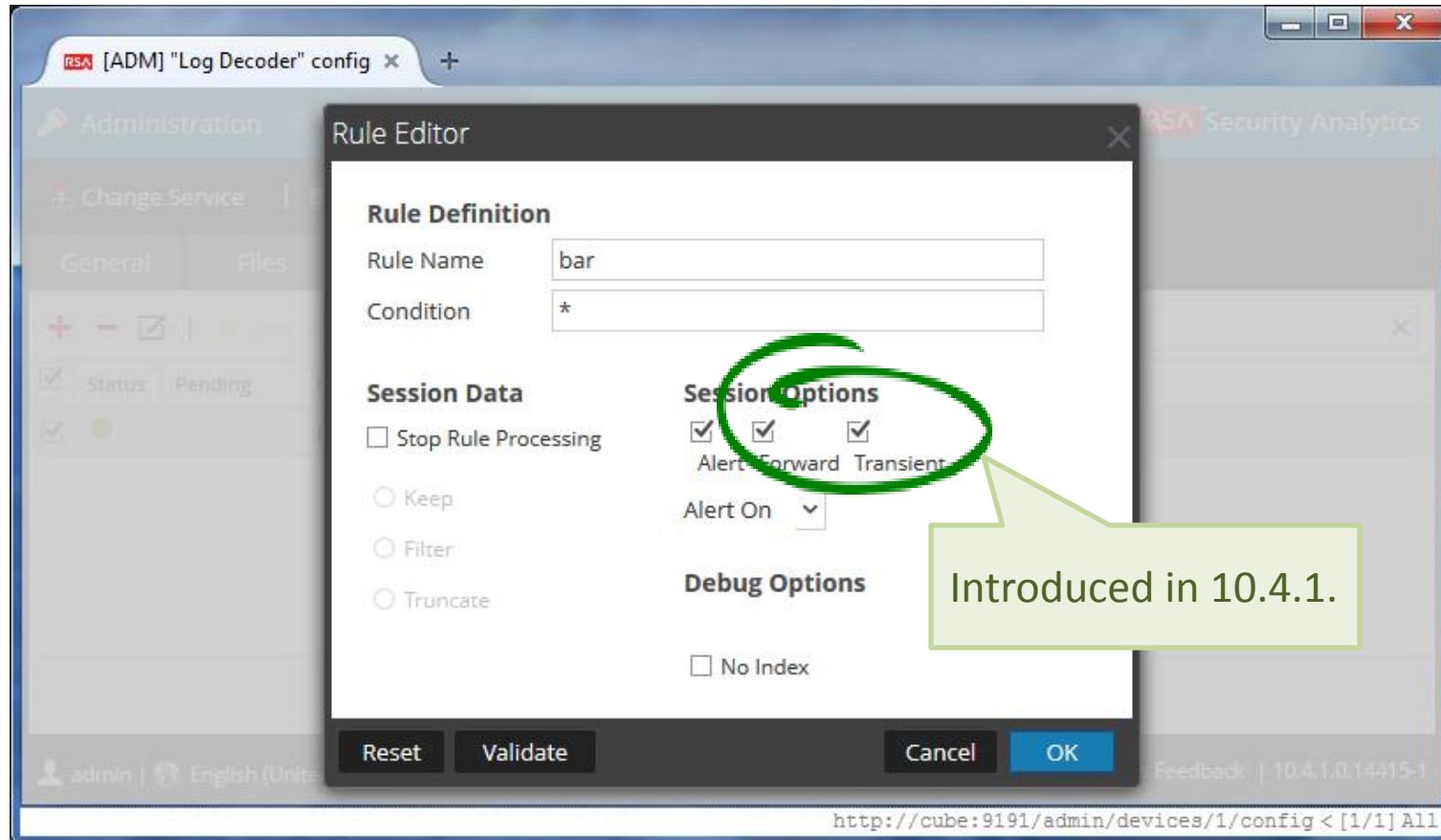
# Log Forwarding Flag Issues



In 10.3.4, after editing a rule in UI, the 'forward' and 'transient' flags are missing

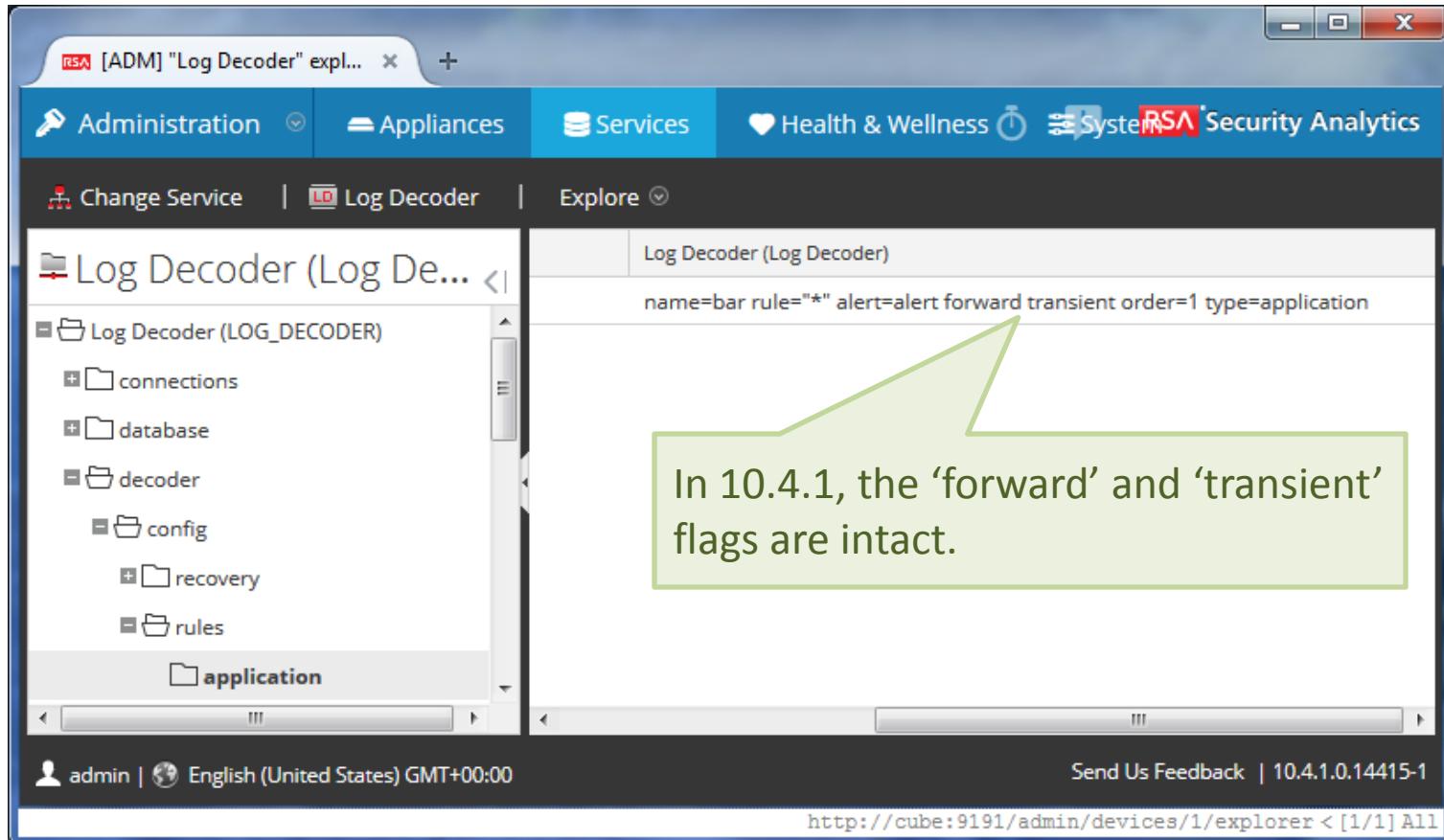


# Log Forwarding Flag Issues



Since 10.4.1, the UI recognizes the 'forward' and 'transient' flags

# Log Forwarding Flag Issues



The screenshot shows the RSA Security Analytics interface. The top navigation bar includes 'Administration', 'Appliances', 'Services', 'Health & Wellness', 'System', and the 'RSA Security Analytics' logo. Below the navigation is a breadcrumb menu: 'Change Service' → 'Log Decoder' → 'Explore'. The left sidebar displays a tree structure under 'Log Decoder (Log Decoder)': 'Log Decoder (LOG\_DECODER)' (with 'connections', 'database', 'decoder', 'config' (containing 'recovery'), and 'rules'), and an 'application' folder. The main content area shows a configuration rule named 'Log Decoder (Log Decoder)'. The rule definition is: `name=bar rule="*" alert=alert forward transient order=1 type=application`. A green callout box highlights this rule with the text: 'In 10.4.1, the 'forward' and 'transient' flags are intact.' The bottom of the screen shows the user 'admin', the language 'English (United States) GMT+00:00', a feedback link 'Send Us Feedback | 10.4.1.0.14415-1', and the URL 'http://cube:9191/admin/devices/1/explorer < [1/1] All'.

In 10.4.1, after editing a rule, the 'forward' and 'transient' flags are intact

# Log Forwarding Flag Issues

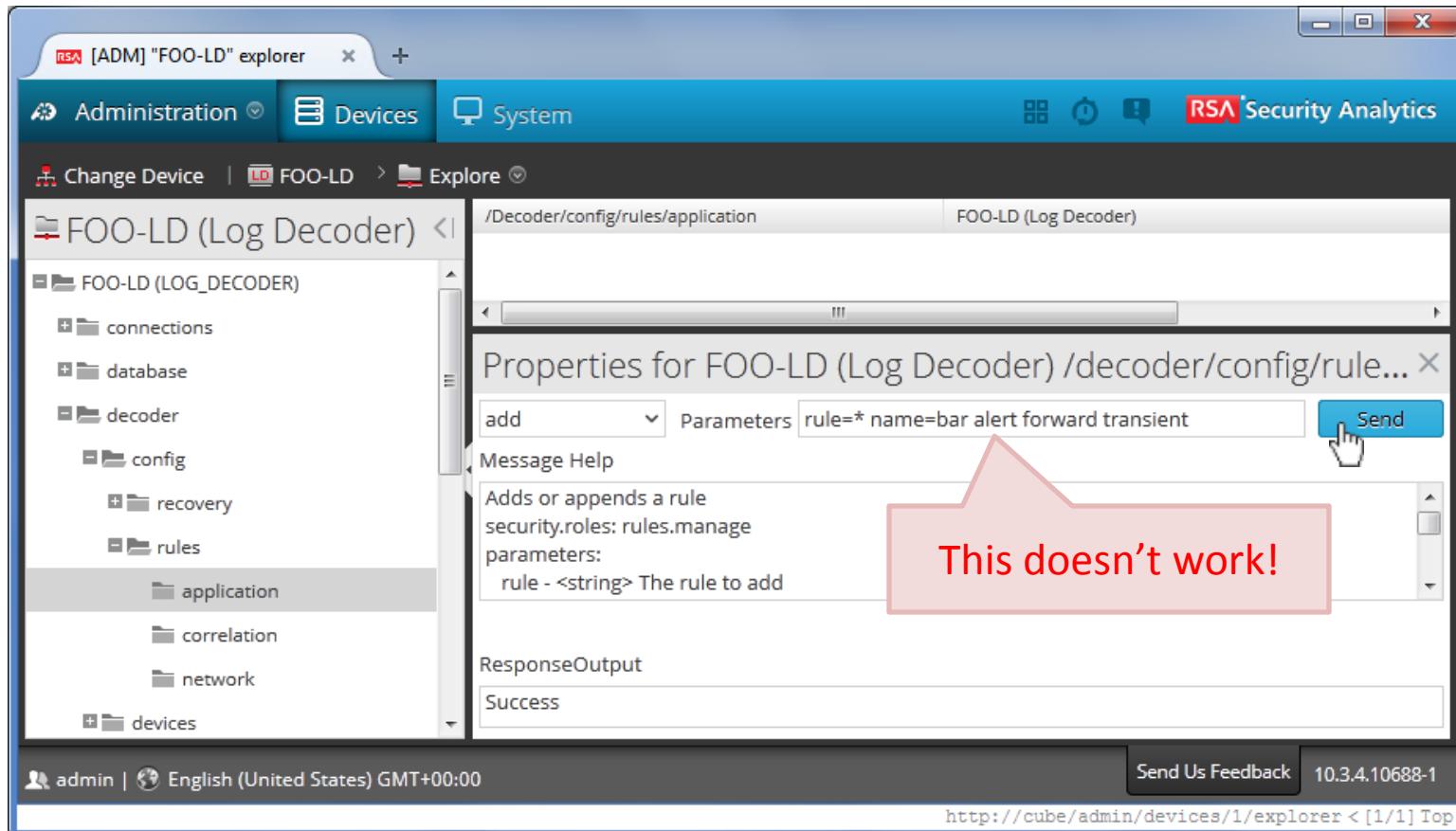
In 10.3.4 and prior versions, 10.4.0 and 10.4.0.1, whenever any action is performed on a log forwarding rule in UI → LD → Config → App Rules, the rule loses the ‘forward’ flag and the ‘transient’ flag, if present.

While losing the ‘transient’ flag does not affect log forwarding, losing the ‘forward’ flag causes log forwarding to break.

Since 10.4.1, there is a fix to make the UI recognize the ‘forward’ and ‘transient’ flags and retain it.

For earlier versions, stick to LD → Explore → /decoder/config/rules for defining and editing log forwarding rules.

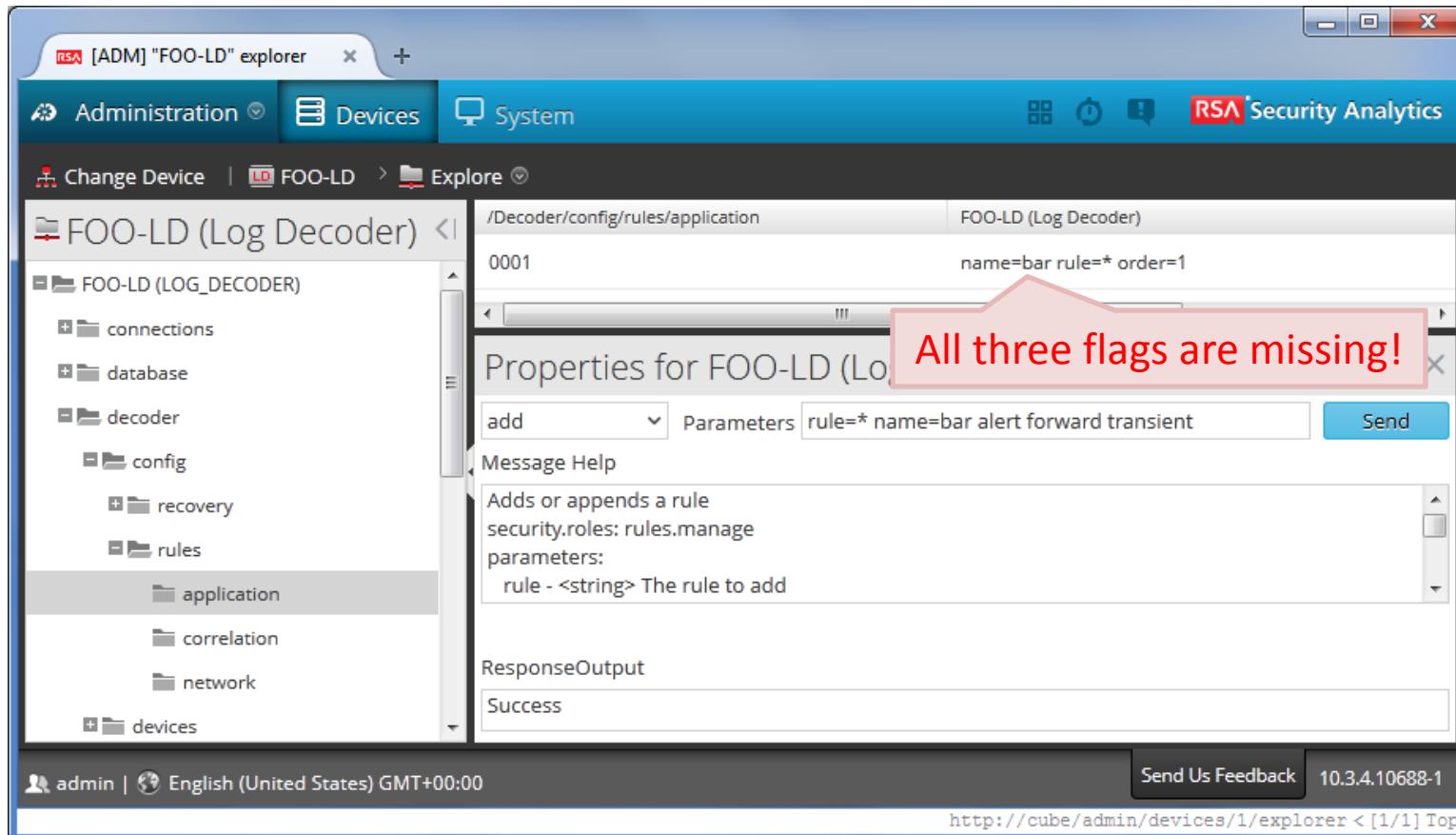
# Log Forwarding Flag Issues



We want to add a rule with 'alert', 'forward' and 'transient' flags via 'Explore' view

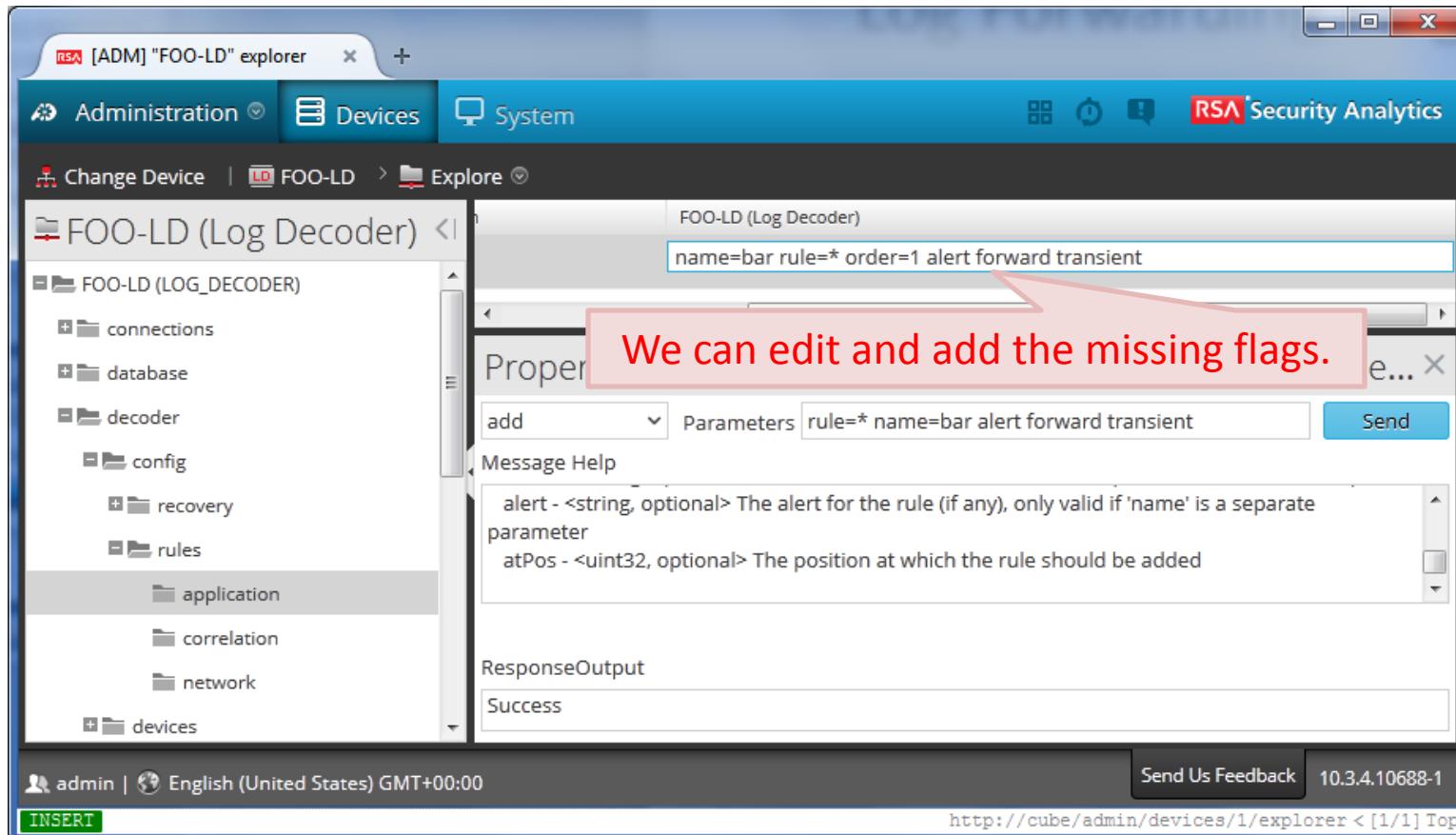


# Log Forwarding Flag Issues



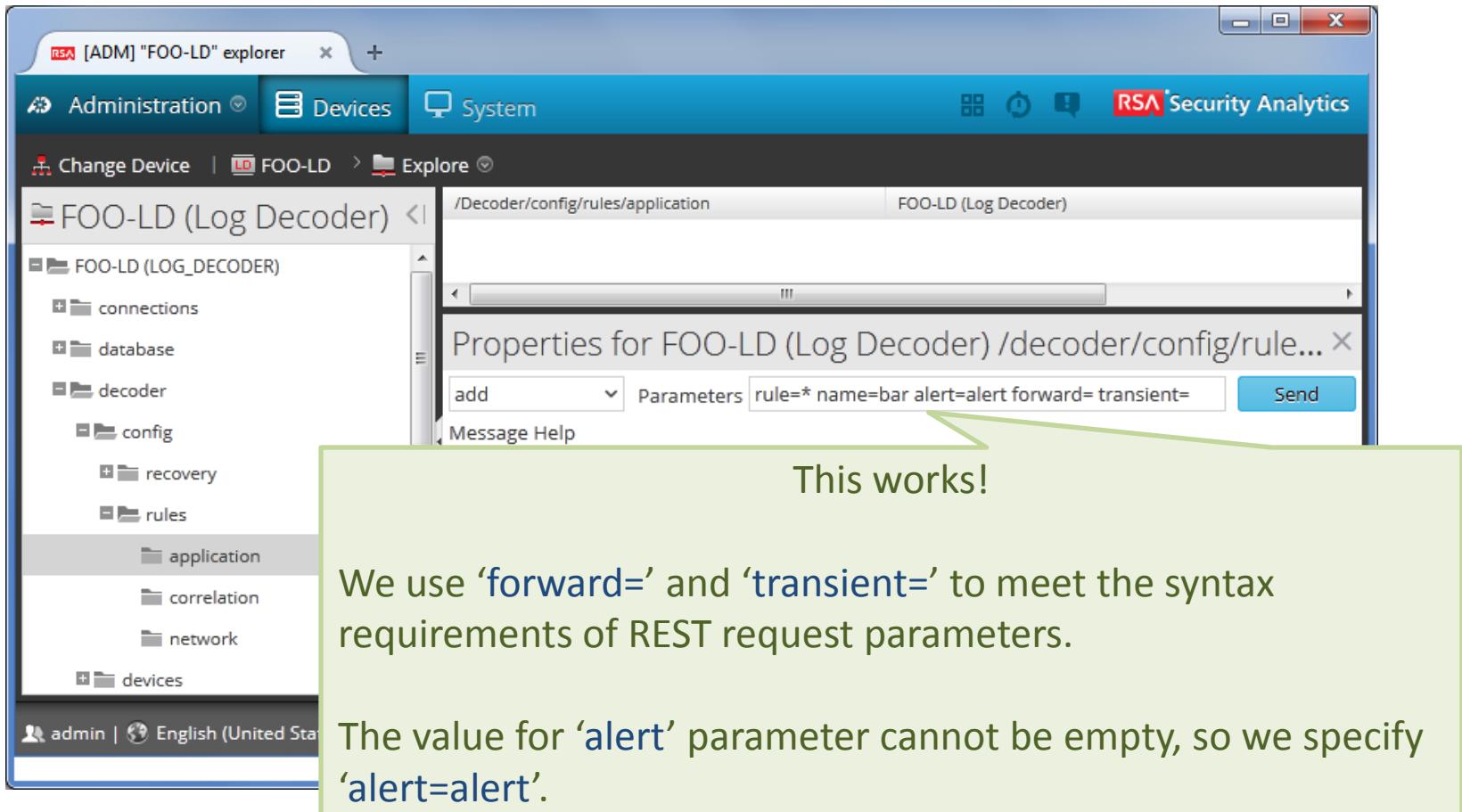
We lost the flags!

# Log Forwarding Flag Issues



We can always edit the rule after editing, but there is another way ...

# Log Forwarding Flag Issues



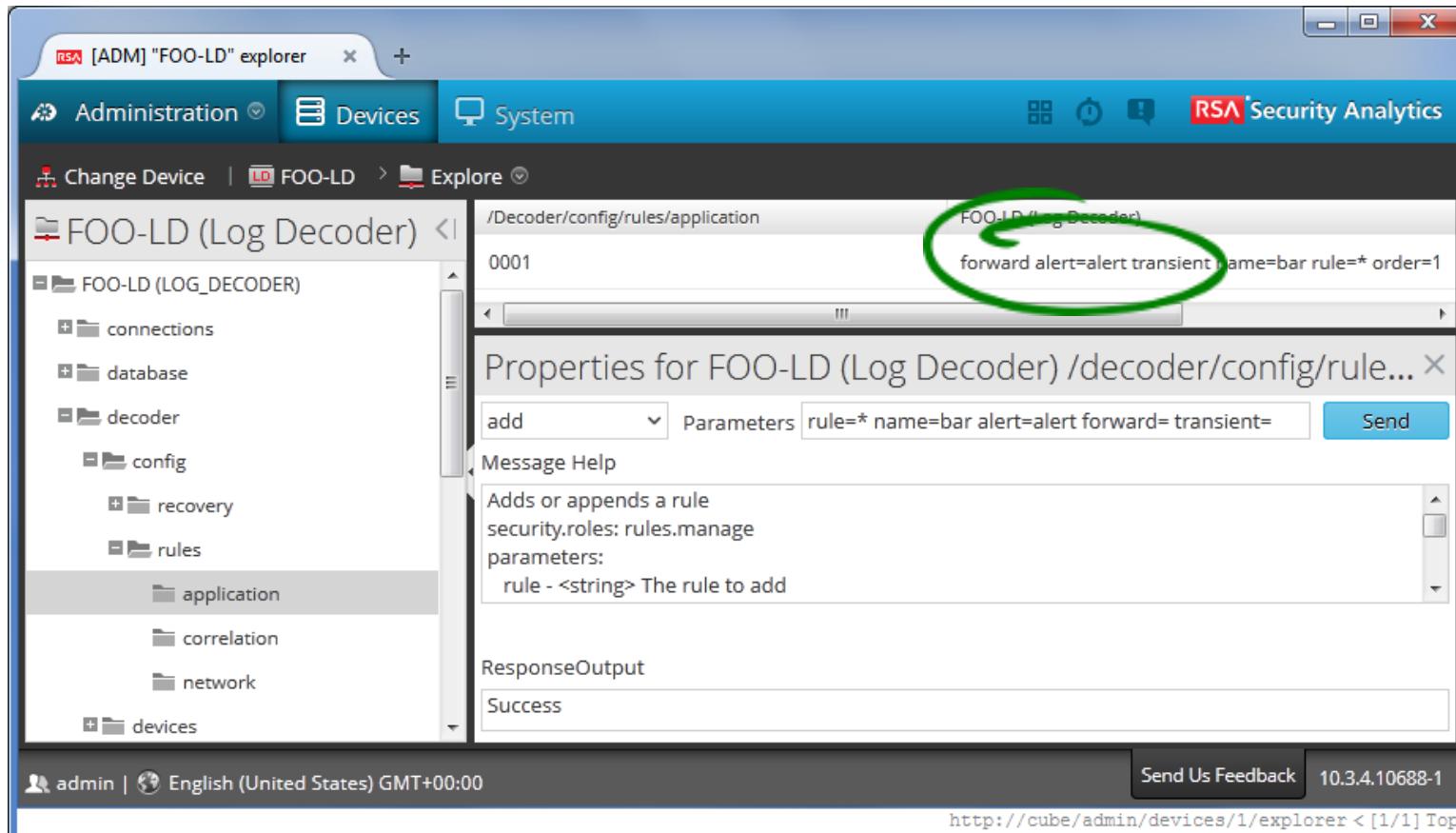
The screenshot shows the RSA Security Analytics administration interface. The left sidebar shows a tree structure under 'FOO-LD (Log Decoder)' with nodes like 'connections', 'database', 'decoder', 'config' (which contains 'recovery' and 'rules'), and 'application'. The 'rules' node is currently selected. The main pane displays a REST API endpoint: '/Decoder/config/rules/application'. A form is present with a dropdown menu set to 'add', a 'Parameters' field containing 'rule=\*&name=bar&alert=alert&forward=transient=' (with 'forward=' and 'transient=' highlighted), and a 'Send' button. A green callout box points to the 'forward=' and 'transient=' parameters with the text 'This works!'. Below this, another callout box contains the text: 'We use 'forward=' and 'transient=' to meet the syntax requirements of REST request parameters.' and 'The value for 'alert' parameter cannot be empty, so we specify 'alert=alert''. The top right corner of the slide features a decorative graphic of overlapping colored squares.

This works!

We use 'forward=' and 'transient=' to meet the syntax requirements of REST request parameters.

The value for 'alert' parameter cannot be empty, so we specify 'alert=alert'.

# Log Forwarding Flag Issues



All flags added



# Thank You!

