RSA® NETWITNESS®
Logs
Implementation Guide

CryptoniteNXT 2.9.0

Daniel Pintal, RSA Partner Engineering
Last Modified: October 25, 2018

RSA READY

Solution Summary

CryptoniteNXT can send alerts and administrative information to RSA NetWitness for display and analysis. This provides greater visibility into potentially malicious activities, misconfiguration, policy decisions, moving target defense violations, and traffic patterns. CryptoniteNXT supports forwarding this data to RSA NetWitness Decoder using CEF messages over Syslog. The RSA NetWitness Decoder may be positioned inside or outside of CryptoniteNXT's protection.

RSA NetWitness Features	
CryptoniteNXT 2.9.0	
Integration package name	Common Event Format
Device display name within NetWitness	cryptonite_nxt
Event source class	Analysis
Collection method	Syslog

RSA NetWitness Community

The RSA NetWitness Community is an online forum for customers and partners to exchange technical information and best practices with each other. All NetWitness customers and partners are invited to register and participate in the **RSA NetWitness Community**.

Release Notes

Release Date	What's New In This Release
10/25/2018	Initial support for CryptoniteNXT.

! > Important: The RSA NetWitness CEF parser is dependent on the partner adhering to the CEF Rules outlined in the *ArcSight Common Event Format (CEF) Guide*. A copy of the Common Event Format guide can be found on http://protect724.hp.com/**.**

Eg. Jan 18 11:07:53 host CEF:Version|Device Vendor|Device Product|Device Version|Signature ID|Name|Severity|[Extension]

Partner Product Configuration

Before You Begin

This section provides instructions for configuring the CryptoniteNXT with RSA NetWitness. This document is not intended to suggest optimum installations or configurations.

It is assumed that the reader has both working knowledge of all products involved, and the ability to perform the tasks outlined in this section. Administrators should have access to the product documentation for all products in order to install the required components.

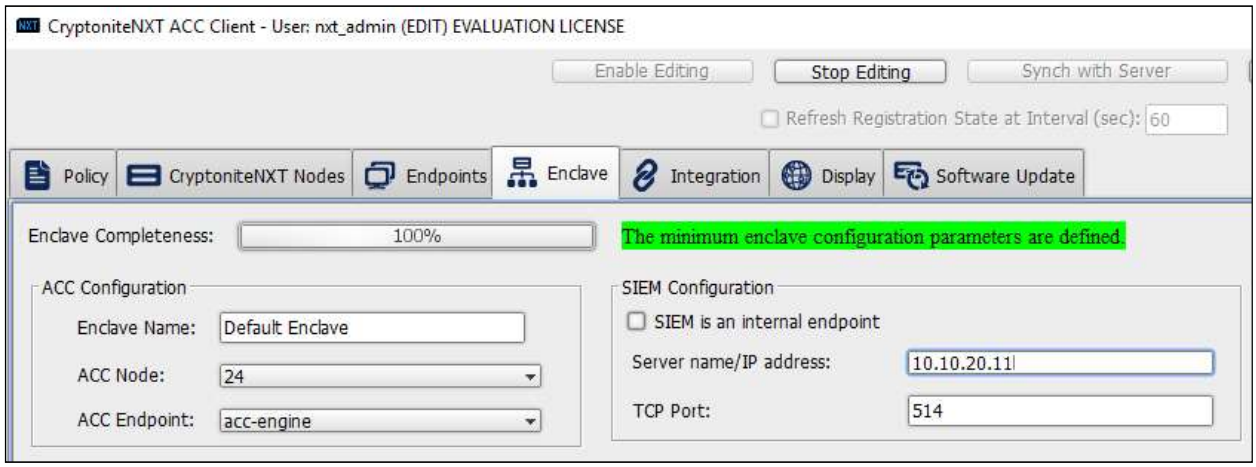
All CryptoniteNXT components must be installed and working prior to the integration. Perform the necessary tests to confirm that this is true before proceeding.

! Important: The configuration shown in this Implementation Guide is for example and testing purposes only. It is not intended to be the optimal setup for the device. It is recommended that customers make sure CryptoniteNXT is properly configured and secured before deploying to a production environment. For more information, please refer to the CryptoniteNXT documentation or website.

CryptoniteNXT Configuration

After completing the previous sections, login to the CryptoniteNXT ACC Client to configure CryptoniteNXT.

1. Click the **Enable Editing** button.
2. Navigate to the **Enclave** tab.



3. If the RSA Netwitness Decoder is inside the CryptoniteNXT enclave, check the **SIEM is an internal endpoint** box and select the decoder's hostname from the list.

   OR

   If the RSA Netwitness Decoder is outside the CryptoniteNXT enclave, uncheck the **SIEM is an internal endpoint** box and enter the decoder's IP.

! > Note: For this configuration, you must also ensure that the ACC Engine has egress policy for port 514 through its configured gateway.

! > Important: The location of other RSA Netwitness components (Concentrator, Admin Server, etc.) is not relevant to this configuration, only the decoder's location applies.

4. Under **SIEM configuration** enter **514** as the TCP port.
5. Click the **Save** button at the bottom of the screen. This change will take effect immediately.

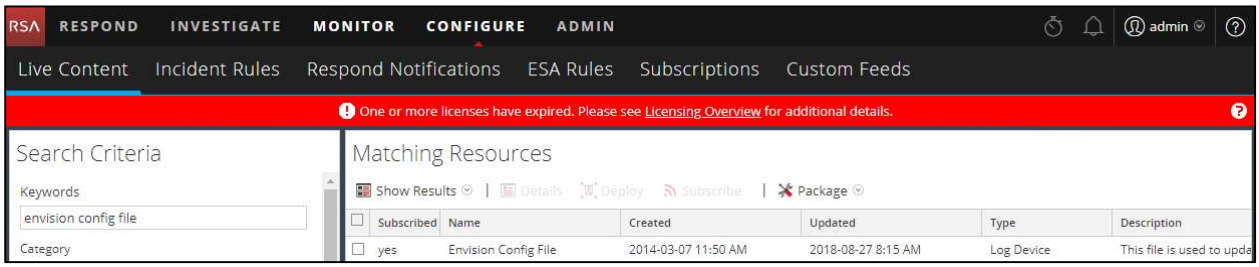
RSA NetWitness Configuration

Deploy the enVision Config File

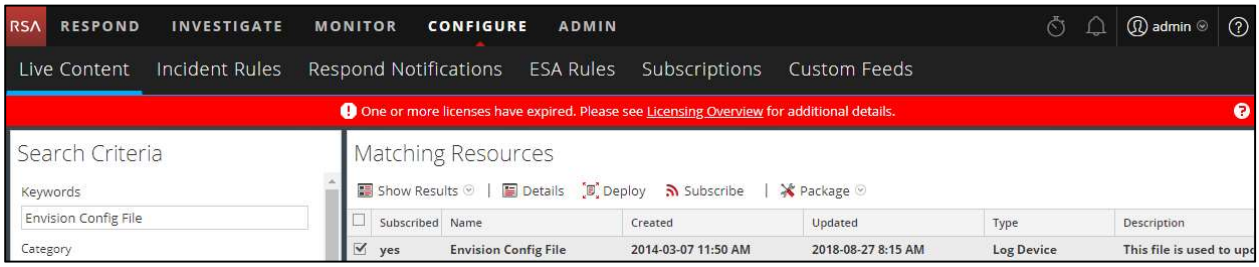
In order to use the RSA Common Event Format, you must first deploy the *enVision Config File* from the **NetWitness Live** module. Log into NetWitness and perform the following actions:

! > Important: Using this procedure will overwrite the existing table_map.xml.

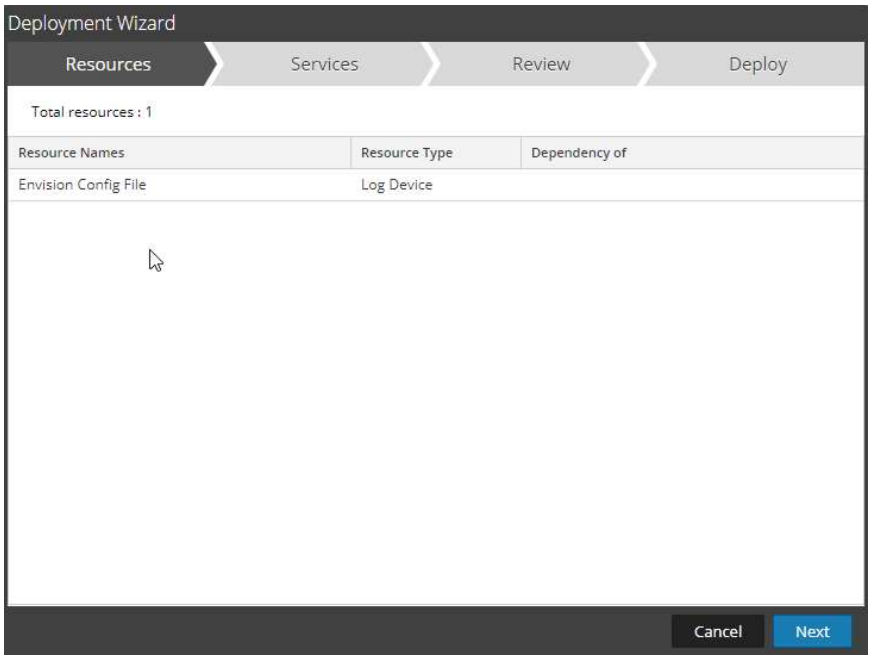
1. From the NetWitness menu, select **Configure > Live Content**.
2. In the keywords field, enter: **enVision**.
3. NetWitness will display the **Envision Config File** in Matching Resources.
4. Select the checkbox next to **Envision Config File**.



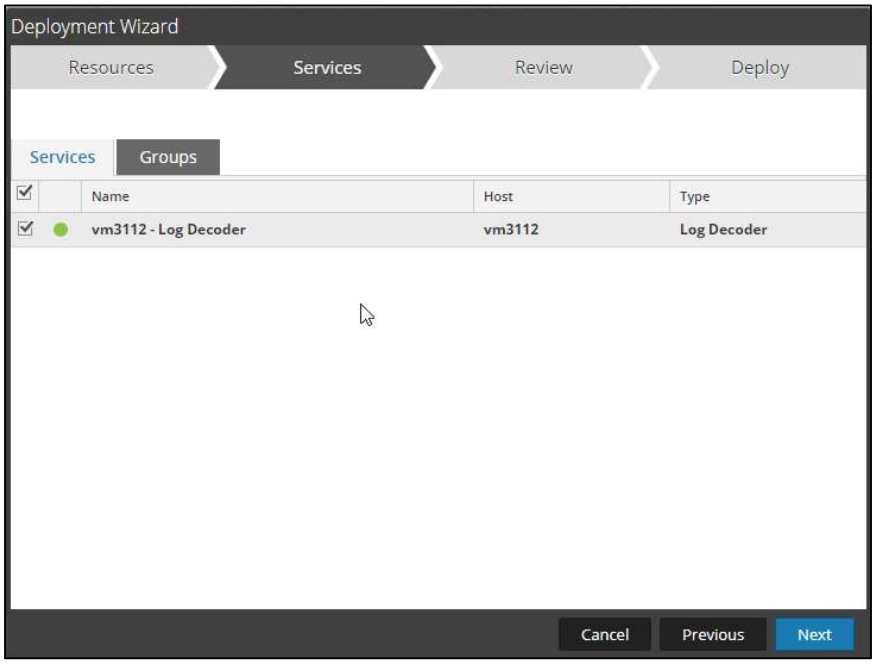
5. Click **Deploy** in the menu bar.



6. Select **Next**.

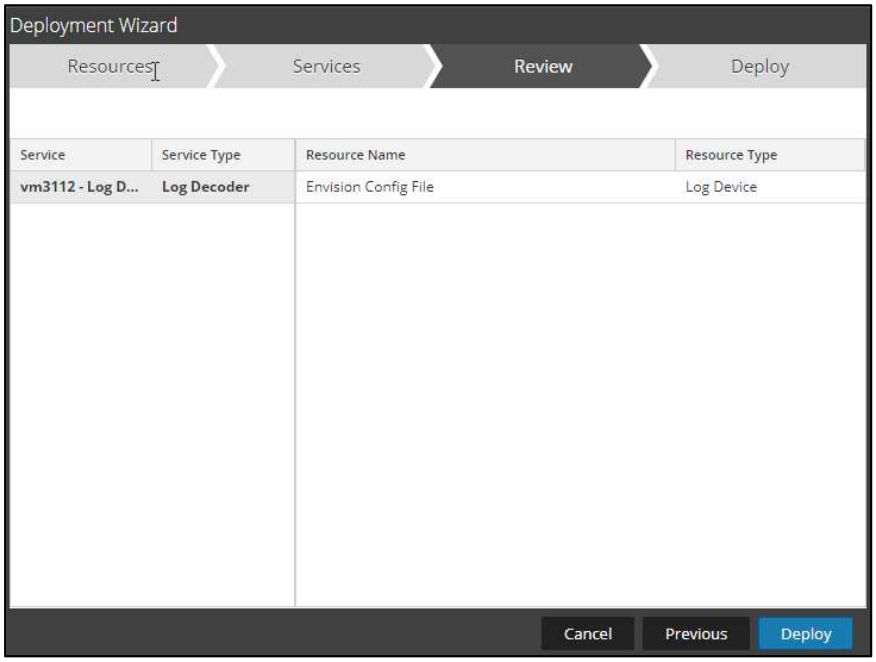


7. Select the **Log Decoder** and select **Next**.

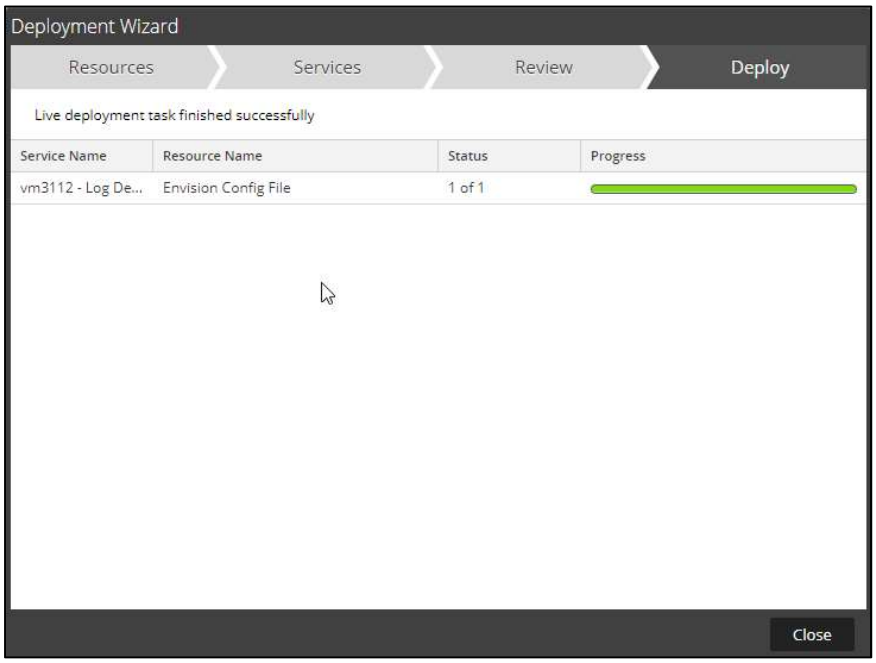


! Important: In an environment with multiple Log Decoders, deploy the Envision Config File to each Log Decoder in your network.

8. Select **Deploy**.



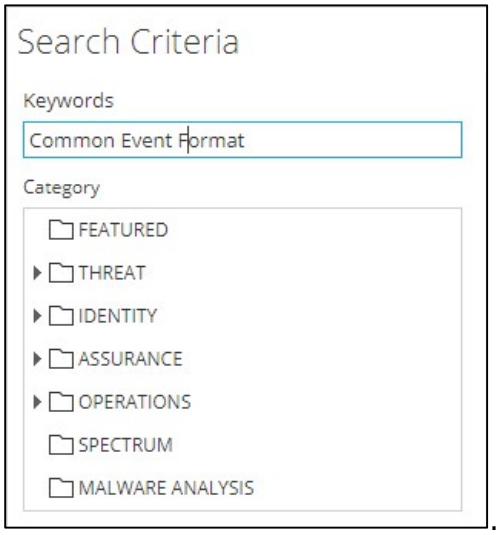
9. Select **Close**, to complete the deployment of the Envision Config file.



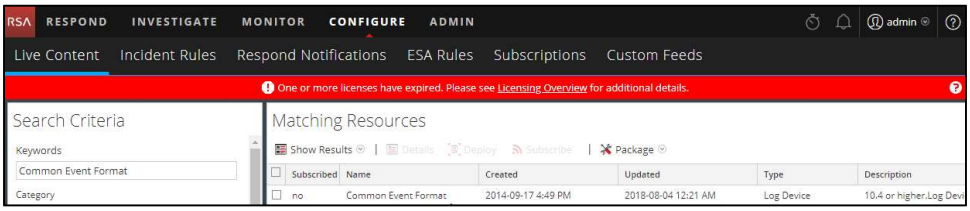
Deploy the Common Event Format

Next, you will need to deploy the *Common Event Format file* from the **NetWitness Live** module. Log into NetWitness and perform the following actions:

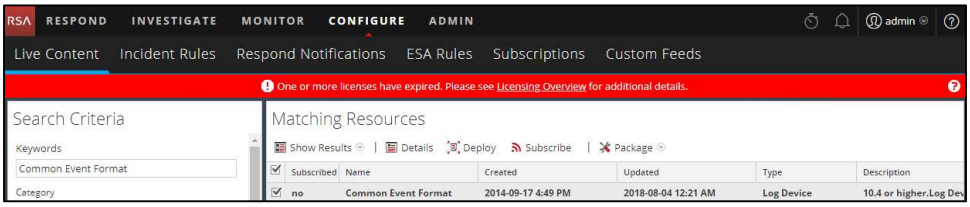
1. From the NetWitness menu, select **Live > Search**.
2. In the keywords field, enter: **Common Event Format**



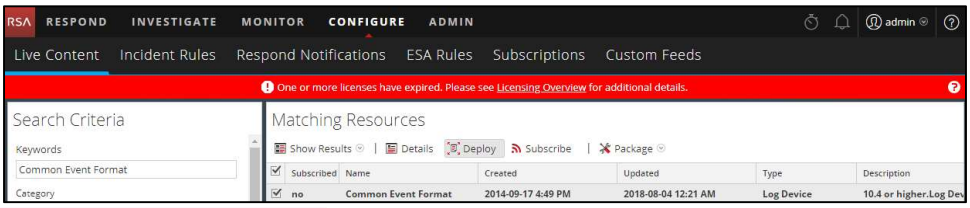
3. RSA NetWitness will display the **Common Event Format** in Matching Resources.



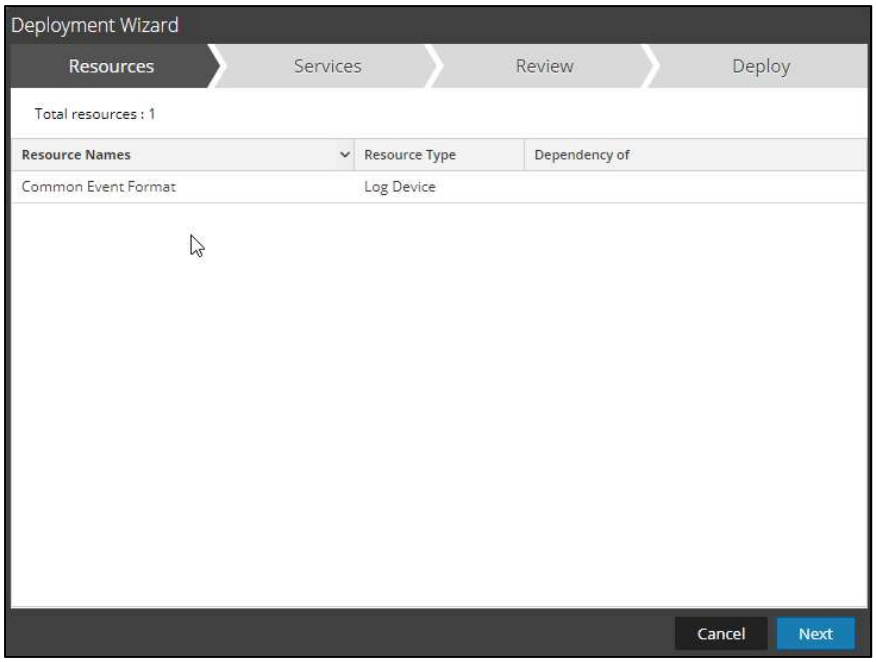
4. Select the checkbox next to **Common Event Format**.



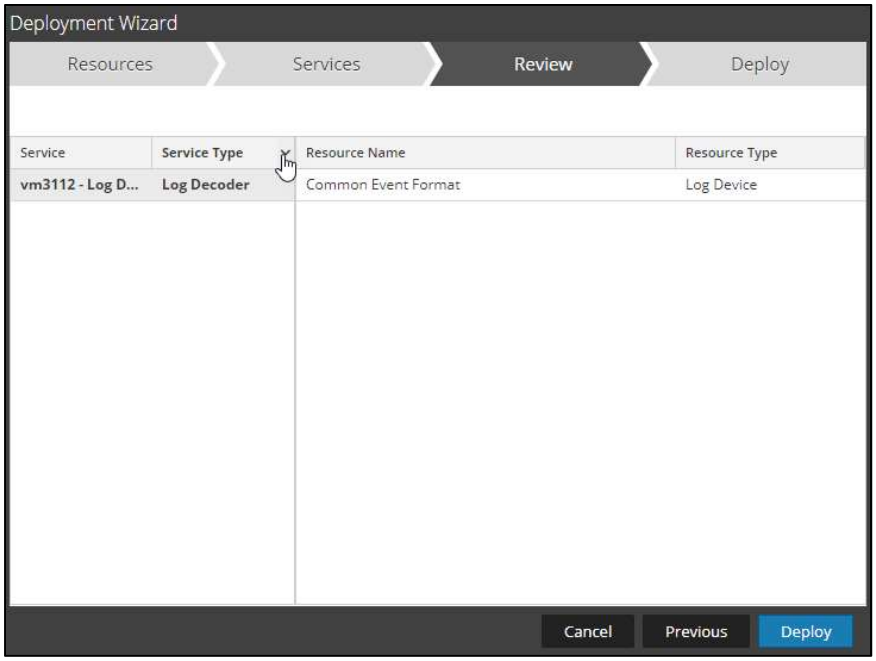
5. Click **Deploy** in the menu bar.



6. Select **Next**.

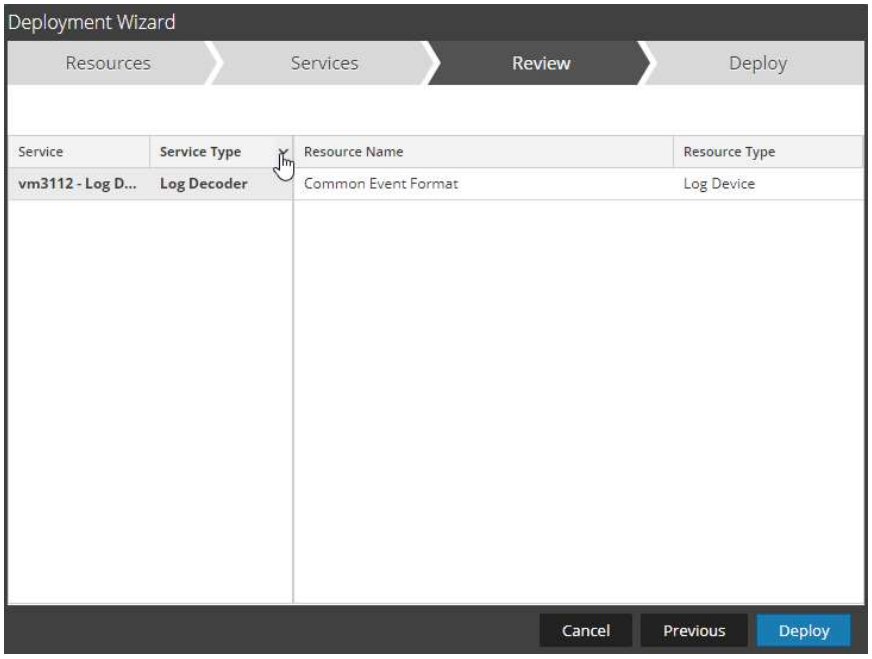


7. Select the **Log Decoder** and Select **Next**.

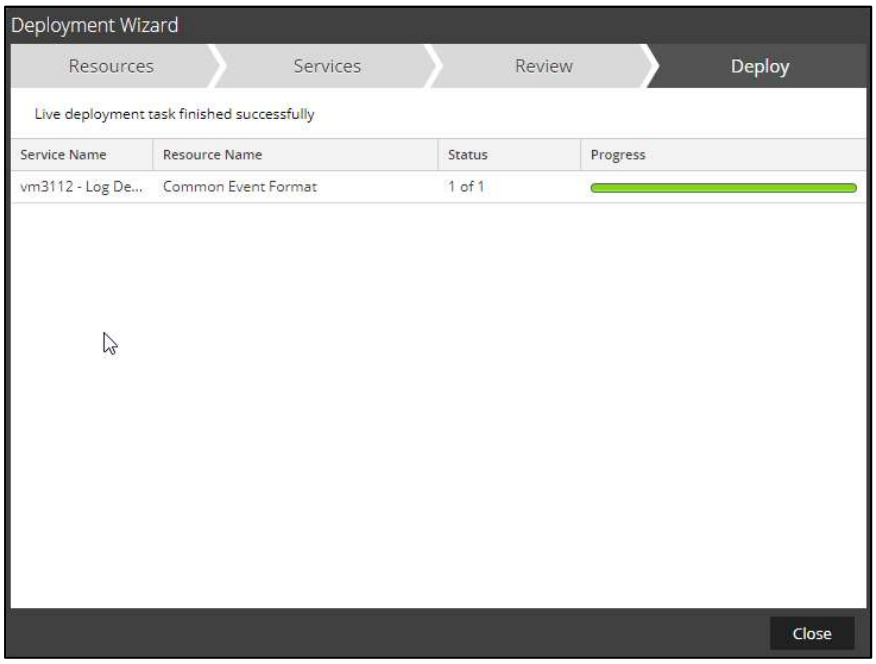


! > Important: In an environment with multiple Log Decoders, deploy the Common Event Format to each Log Decoder in your network.

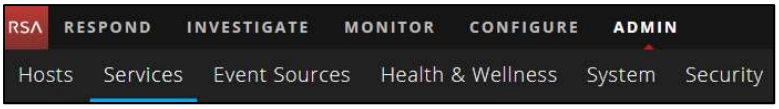
8. Select **Deploy**.



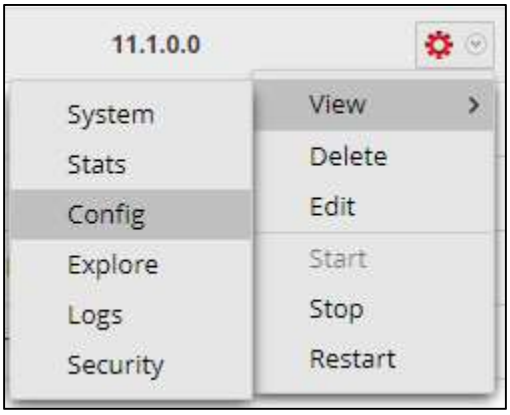
9. Select **Close**, to complete the deployment of the Common Event Format.



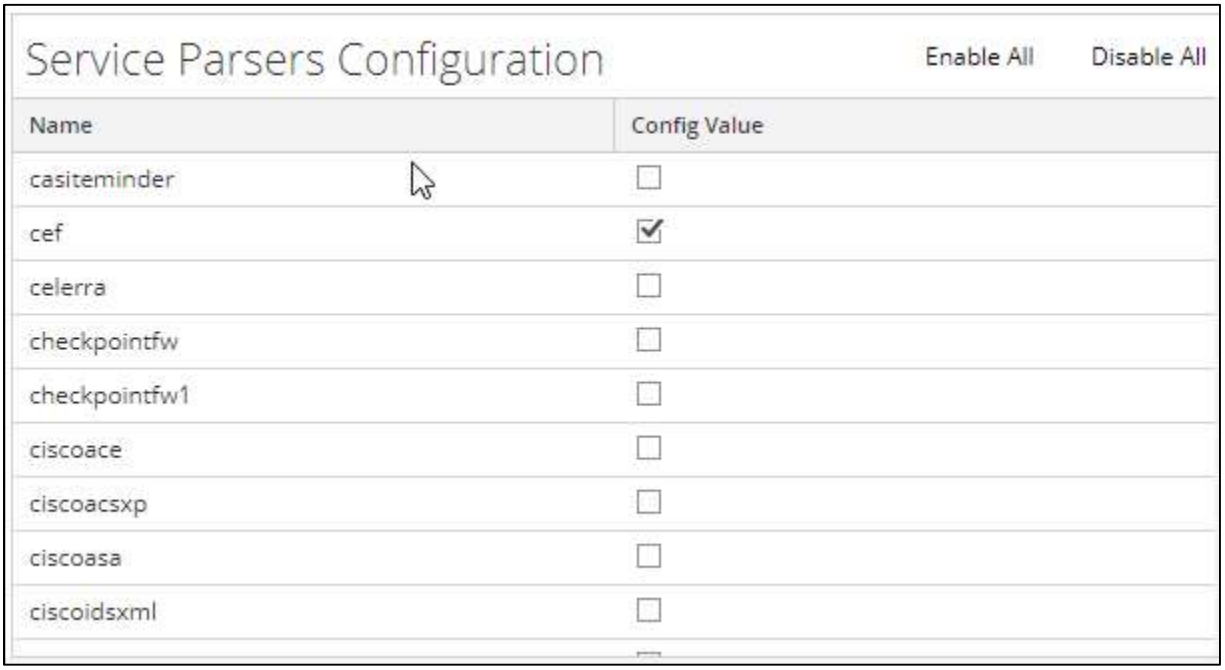
10. Ensure that the CEF Parser is enabled on the Log Decoder(s) by selecting **Admin > Services** from the NetWitness Dashboard.



11. Locate the Log_Decoder and click the gear to the right and select **View>Config**.



12. **Check** the box next to the cef Parser within the Service Parsers Configuration and select **Apply**.



Edit the Common Event Format to collect CryptoniteNXT event times

! Important: The cef.xml file is overwritten by NetWitness Live during updates, it is important to maintain backups of the file in the event of a typing error or unforeseen event.

1. Using WinSCP or other application to access the RSA NetWitness Log Decoder open a connection and locate the **/etc/netwitness/ng/envision/etc/devices/cef** folder. Backup cef.xml and edit the existing CEF.XML file.
2. Locate the end of the **<MESSAGE** section and copy/paste the following lines below into the file after the **/>** of the preceding <MESSAGE and contents;

Example:

```
<HEADER
       id1="0010"
       id2="0010"
       messageid="VENDORMAP(devvendor, product)"
       content="&lt;event_time_string&gt; &lt;hostname&gt;
&lt;hfld1&gt;[&lt;process_id&gt;]: &lt;hfld2&gt;
CEF:&lt;cefversion&gt;|&lt;devvendor&gt;|&lt;product&gt;|&lt;version&gt;|&lt;event_typ
e&gt;|&lt;event_description&gt;|&lt;severity&gt;|&lt;!payload&gt;" />


<MESSAGE
       id1="cryptonite_nxt"
       id2="cryptonite_nxt"
       functions="&lt;@event_time:*EVNTTIME($MSG,'%X',param_starttime)&gt;&lt;@startti
       me:*EVNTTIME($MSG,'%X',param_starttime)&gt;&lt;@endtime:*EVNTTIME($MSG,'%X',par
       am_endtime)&gt;"
              content="&lt;param_starttime&gt;&lt;param_endtime&gt;&lt;msghold&gt;" />
```

Edit the Common Event Format Custom to support custom fields

! > Important: The cef-custom.xml file is not overwritten by NetWitness Live during updates, however it is important to maintain backups of the file in the event of a typing error or unforeseen event.

1. Using WinSCP or other application to access the RSA NetWitness Log Decoder open a connection and locate the **/etc/netwitness/ng/envision/etc/devices/cef** folder. If the cef-custom.xml file does not exist create one. If the file exists create a backup cef-custom.xml and edit the file.
2. If this is a new **cef-custom.xml** file, copy the following into the file, otherwise copy only the required sections.

Example"

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<DEVICEMESSAGES>
<!-- Example Comment
#
# cef-custom.xml Reference: https://community.rsa.com/docs/DOC-79189
#

<HEADER
       id1="0010"
       id2="0010"
       messageid="VENDORMAP(devvendor, product)"
       content="&lt;event_time_string&gt; &lt;hostname&gt;
&lt;hfld1&gt;[&lt;process_id&gt;]: &lt;hfld2&gt;
CEF:&lt;cefversion&gt;|&lt;devvendor&gt;|&lt;product&gt;|&lt;version&gt;|&lt;event_typ
e&gt;|&lt;event_description&gt;|&lt;severity&gt;|&lt;!payload&gt;" />

<MESSAGE
       id1="cryptonite_nxt"
       id2="cryptonite_nxt"
       functions="&lt;@event_time:*EVNTTIME($HDR,'%X',event_time_string)&gt;&lt;@start
time:*EVNTTIME($MSG,'%X',param_starttime)&gt;&lt;@endtime:*EVNTTIME($MSG,'%X',param_en
dtime)&gt;"
       content="&lt;param_starttime&gt;&lt;param_endtime&gt;&lt;msghold&gt;" />

-->

<VendorProducts>
       <Vendor2Device vendor="cryptonite_nxt" product="Cryptonite NXT"
device="cryptonite_nxt" group="Analysis"/>
</VendorProducts>

           <ExtensionKeys>
                    <ExtensionKey cefName="severity" metaName="severity"/>

              <ExtensionKey cefName="cn1" metaName="cn_fld">
                    <device2meta device="trendmicrodsa" metaName="result" label="Host
ID"/>
                    <device2meta device="cryptonite_nxt" metaName="ipversion"/>
              </ExtensionKey>
              <ExtensionKey cefName="cn1Label" metaName="cs_fld"/>

              </ExtensionKeys>

</DEVICEMESSAGES>
```

Edit the NetWtness Table-Map-Custom.xml file

! > Important: The Table-Map-Custom.xml file is not overwritten by NetWitness Live during updates, however it is important to maintain backups of the file in the event of a typing error or unforeseen event.

1. Using WinSCP or other application to access the RSA Netwitness Log Decoder open a connection and locate the **/etc/netwitness/ng/envision/etc/** folder.
2. If one exists, backup the **table-map-custom.xml** and then edit the existing table-map-custom.xml file.
3. Copy and paste the entire section below into a new file or only the lines between the **<mappings>…</mappings>** if the **table-map-custom.xml** file exists;

Example:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
# attributes:
#      envisionName: The name of the column in the universal table
#      nwName:                    The name of the NetWitness meta field
#      format:                    Optional. The language key data type. See
LanguageManager. Defaults to "Text".
#      flags:              Optional. One of None|File|Duration|Transient. Defaults to
"None".
#      failureKey:         Optional. The name of the NW key to write data if
conversion fails. Defaults to system generated "parse.error" meta.
#      nullTokens:         Optional. The list of "null" tokens. Pipe separated.
Default is no null tokens.
-->
<mappings>

       <mapping envisionName="outcome" nwName="result" flags="None" format="Text"
       envisionDisplayName="outcome|Result|Volume|Information|Reason|Succeed/Failed"/>
       <mapping envisionName="protocol" nwName="protocol" flags="None" format="Text"
       envisionDisplayName="Protocol"/>

       <mapping envisionName="severity" nwName="severity" flags="none" format="Text"/>
       <mapping envisionName="ipversion" nwName="ipversion" flags="None" format="Text"
       envisionDisplayName="ipversion"/>
       <mapping envisionName="hardware_id" nwName="hardware.id" flags="none"
       format="Text"/>
       <mapping envisionName="sinterface" nwName="sinterface" flags="none"
       format="Text"/>
       <mapping envisionName="stransaddr" nwName="stransaddr" flags="none"
       format="Text"/>
       <mapping envisionName="sport" nwName="ip.srcport" flags="none" format="UInt16"
       nullTokens="-|(null)|N/A"/>
       <mapping envisionName="event_counter" nwName="event.counter" flags="none"
       format="Int32"/>
       <mapping envisionName="endtime" nwName="endtime" flags="none" format="TimeT"/>
       <mapping envisionName="event_time_string" nwName="event.time.str" flags="none"
       format="Text"/>

</mappings>
```

4. Restart the **Log Decoder services** to begin log collection.

CryptoniteNXT Collection Example from NetWitness Investigator:

☐ 2018-10-24T13:55:58	Log	cryptonite_nxt	454 bytes	192.168.0.55 -> 10.1.11.210 sessionid : 289576 device.ip : 10.100.169.146 medium : 32 device.type : cryptonite_nxt device.class : Analysis event.time.str : 1534780551643054 cryp-54000002 NXT0[1209]: alias.host : NXT3 version : 2.7.0_RC2-11-gd06f9de event.type : 30100 event.desc : attempted random token scan from endpoint severity : 8 hardware.id : 54000002 sinterface : 6/0 host.src : cryp2 user.src : hismirnioglou stransaddr : 10.10.21.93 ipversion : 4 netname : private src netname : private dst direction : lateral protocol : 6 ip.srcport : 45772 ip.dstport : 3000 event.counter : 1 event.time : 2018-Aug-20 15:55:51.000 starttime : 2018-Aug-20 15:50:19.000 endtime : 2018-Aug-20 15:50:19.000 msg.id : cryptonite_nxt device.disc : 100 did : vm3112 rid : 289568 ip.all : 10.100.169.146 host.all : NXT3 eth.all : 9C:EB:E8:28:04:B6 host.all : cryp2 user.all : hismirnioglou ip.all : 192.168.0.55 ip.all : 10.1.11.210

Certification Checklist for RSA NetWitness

Date Tested: October 25, 2018

Certification Environment		
Product Name	**Version Information**	**Operating System**
RSA NetWitness	11.2	Virtual Appliance
CryptoniteNXT	2.9.0	Virtual or Hardware Appliance

NetWitness Test Case	Result
Device Administration	
Partner's device name appears in Device Parsers Configuration	✓
Device can be enabled from Device Parsers Configuration	✓
Device can be disabled from Device Parsers Configuration	✓
Device can be removed from Device Parsers Configuration	✓
Investigation	
Device name displays properly from Device Type	✓
Displays Meta Data properly within Investigator	✓

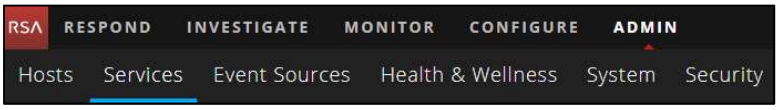
✓ = Pass ✗ = Fail N/A = Non-Available Function

Appendix

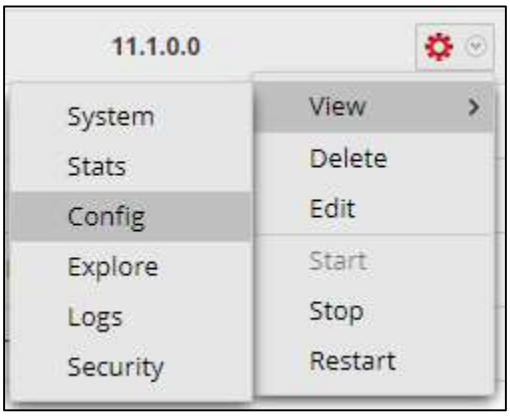
NetWitness Disable the Common Event Format Parser

To disable the NetWitness Common Event Format Parser and not delete it perform the following:

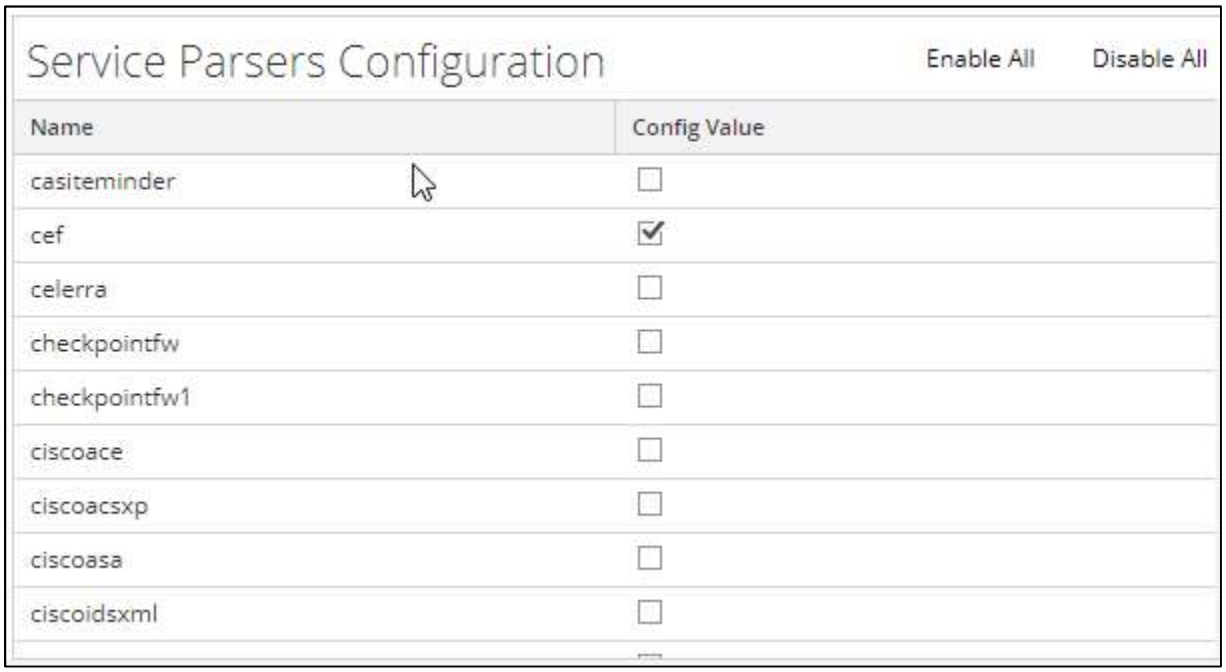
1. Select the NetWitness **Admin > Services**.



5. Select the Log Decoder, then select **View > Config.**



6. From the **Service Parses Configuration** window, scroll down to the CEF parser and uncheck the Config Value checkbox.

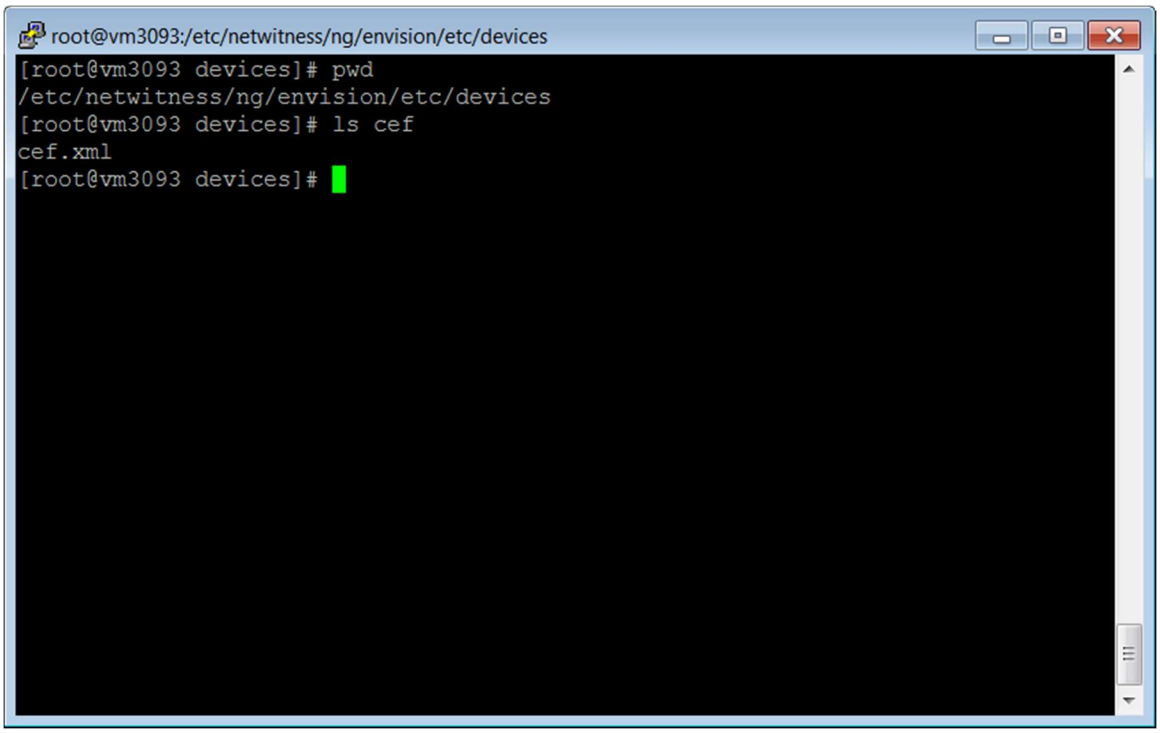


7. Click **Apply** to save settings.

NetWitness Remove Device Parser

To remove the NetWitness Integration Package files from the environment, perform the following:

1. Connect to the NetWitness Log Decoder/Collector Server using SSH and open the **/etc/netwitness/ng/envision/etc/devices** folder.



8. Search for and delete the CEF folder and its contents.

Known Issues

List of custom CEF values not captured by RSA NetWitness

CryptoniteNXT has internal logic to de-duplicate individual events within a time window into a single CEF message. To achieve this, a count, window start time, and window end time are provided with each CEF message. RSA NetWitness only uses the start time. The event.time and starttime fields in RSA NetWitness are identical for these messages. The endtime and event.counter fields are displayed and available for filtering but are generally unused. The implication is that some searches may miss events or graphs may show incorrect counts. For example, if a single CryptoniteNXT reported CEF message collapses events over a window with a start time of 1pm and an end time of 3pm, a search for all events between 12:59pm and 1:01pm will match this event, but a search for all events between 1:01 and 4pm will not match this event even though the event's window overlaps this time range.