RSA® NETWITNESS®
Logs
Implementation Guide

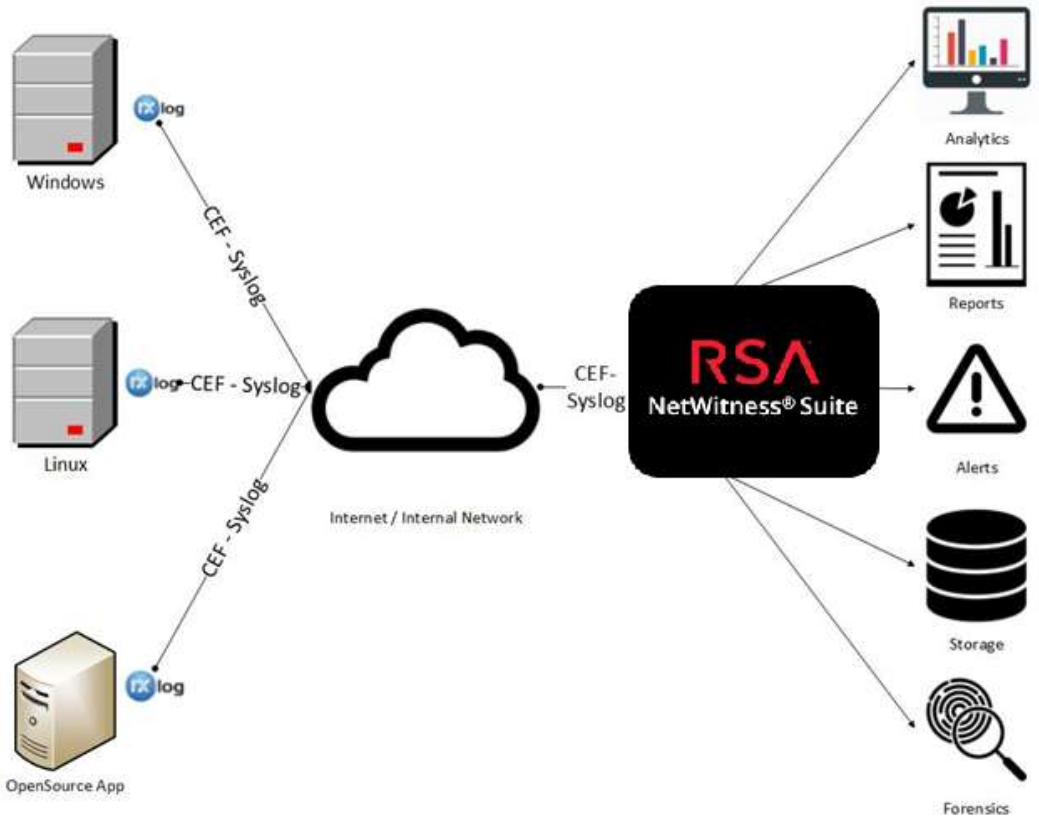
NXLog Enterprise Edition

Daniel Pintal, RSA Partner Engineering
Last Modified: November 19, 2018

RSA READY

Solution Summary

RSA NetWitness Features	
NXLog Enterprise Edition	
Integration package name	Common Event Format
Device display name within NetWitness	NXlog_NXlog
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
10/24/2018	Initial support for NXLog Enterprise Edition.

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]

Important: The time displayed in the CEF log header is parsed into evt.time.str. No other time formats are parsed by default.

Partner Product Configuration

Before You Begin

This section provides instructions for configuring the NXLog Enterprise Edition 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 NXLog Enterprise Edition 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 NXLog Enterprise Edition is properly configured and secured before deploying to a production environment. For more information, please refer to the NXLog Enterprise Edition documentation or website.

NXLog Enterprise Edition Configuration

NXLog Enterprise Edition can generate data in the Arcsight Common Event Format (CEF) which can be ingested and parsed by NetWitness.

Note that the NXLog Enterprise Edition supports various log sources.

This guide provides configuration steps to collect Windows Eventlog only using the *im_msvistalog* input module. The collected data is converted to the CEF syslog format with the *xm_cef* extension module. See **https://nxlog.co/documentation/nxlog-user-guide#xm_cef** for more information.

Follow the instructions on how to deploy the NXLog Enterprise Edition on Windows:

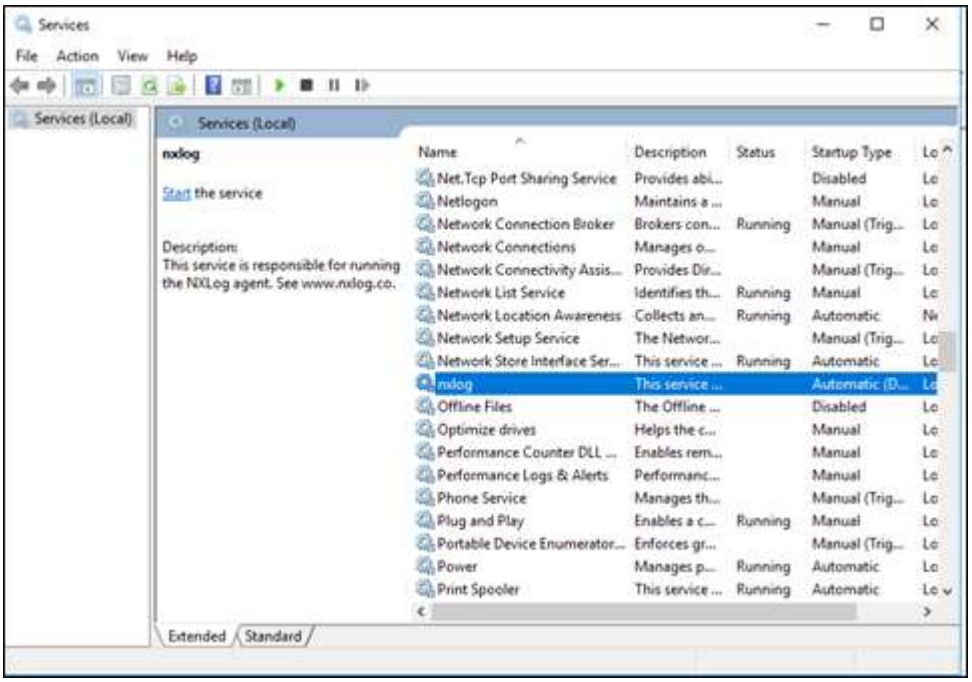
https://nxlog.co/documentation/nxlog-user-guide#deploy-windows

1. Once the msi is deployed edit the *nxlog.conf* configuration file to contain the following module configuration:

Example:

```
#---------------------------------------------------------------------
<Extension cef>
    Module  xm_cef
</Extension>

<Extension syslog>
    Module  xm_syslog
</Extension>

<Input eventlog>
    Module  im_msvistalog
</Input>

<Output udp>
    Module  om_udp
    Host    10.0.0.42
    Port    514
    Exec    $Message = to_cef(); to_syslog_bsd();
</Output>

<Route nw>
    Path    eventlog => udp
</Route>
#---------------------------------------------------------------------
```

2. Start the *nxlog* service on the Windows Server server:



3. Once the service is started please check the *nxlog.log* file for any errors.

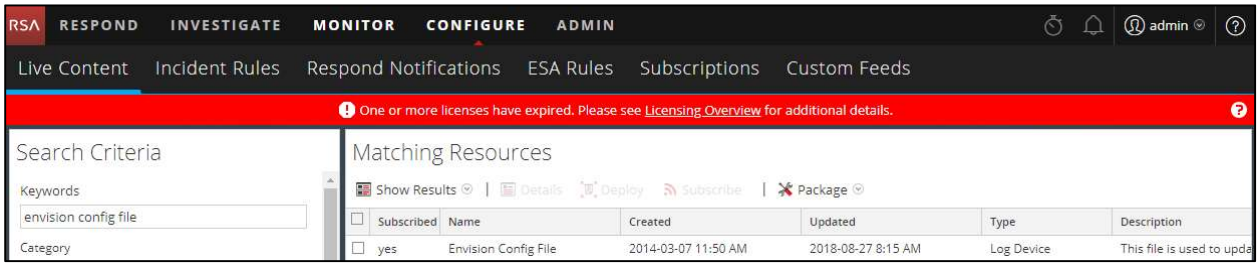
RSA NetWitness Configuration

Deploy the enVision Config File

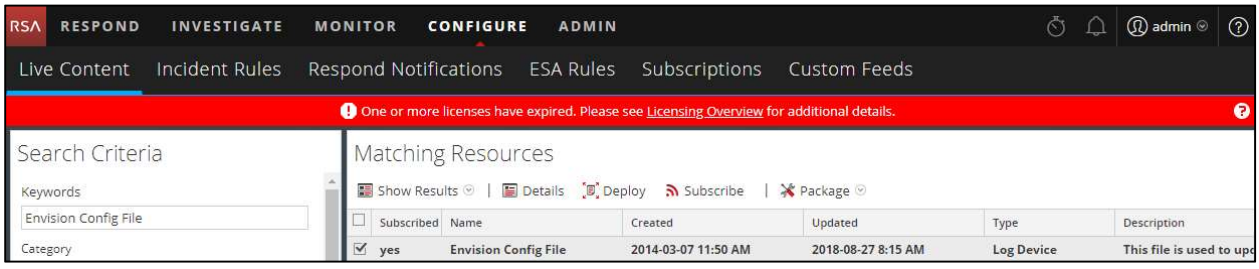
In order to use the RSA Common Event Format, you must first deploy the *enVision Config File* from the **NetWitness Live** module. Log into NetWitness and perform the following actions:

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

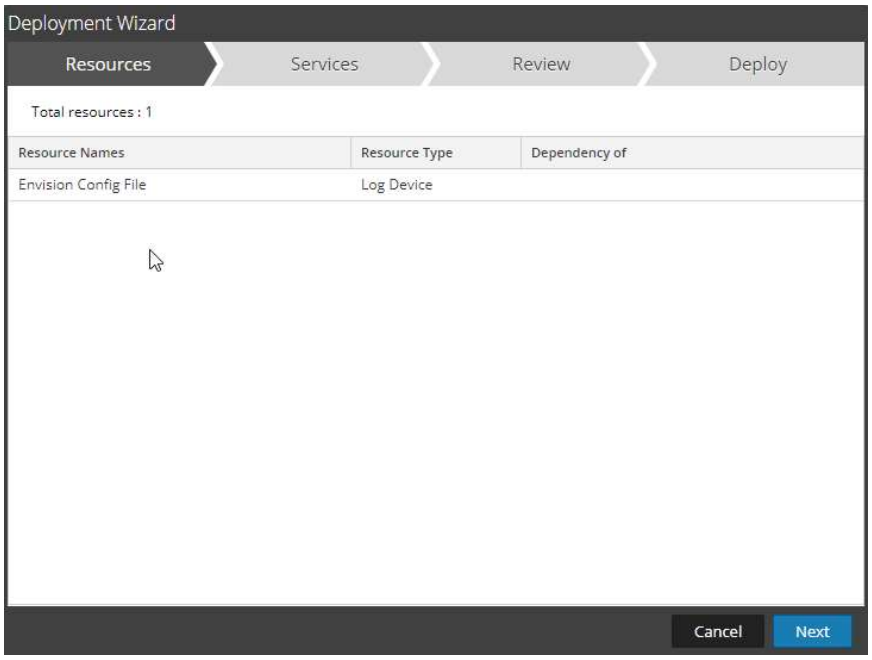
1. From the NetWitness menu, select **Configure > Live Content**.
2. In the keywords field, enter: **enVision**.
3. NetWitness will display the **Envision Config File** in Matching Resources.
4. Select the checkbox next to **Envision Config File**.



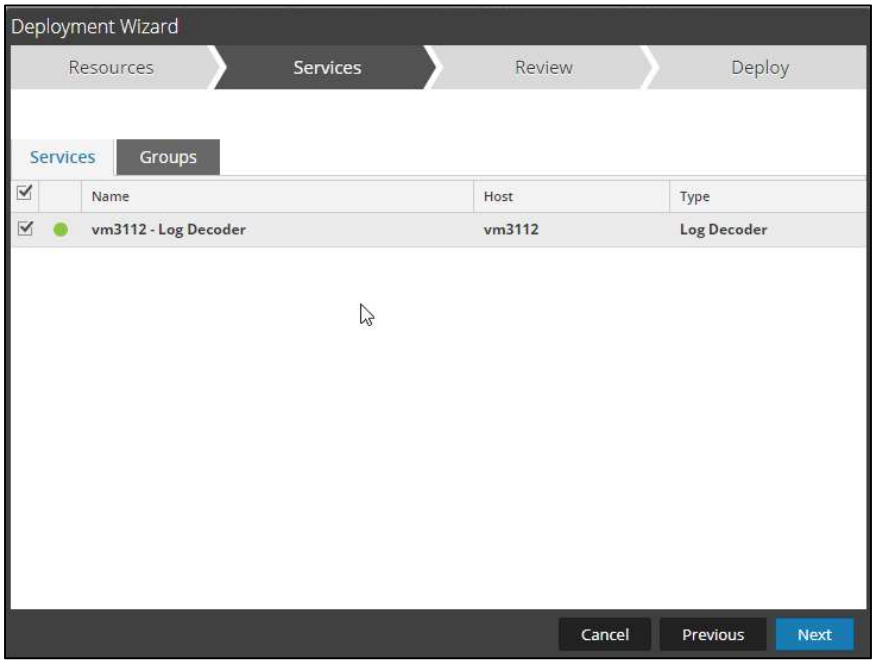
5. Click **Deploy** in the menu bar.



6. Select **Next**.

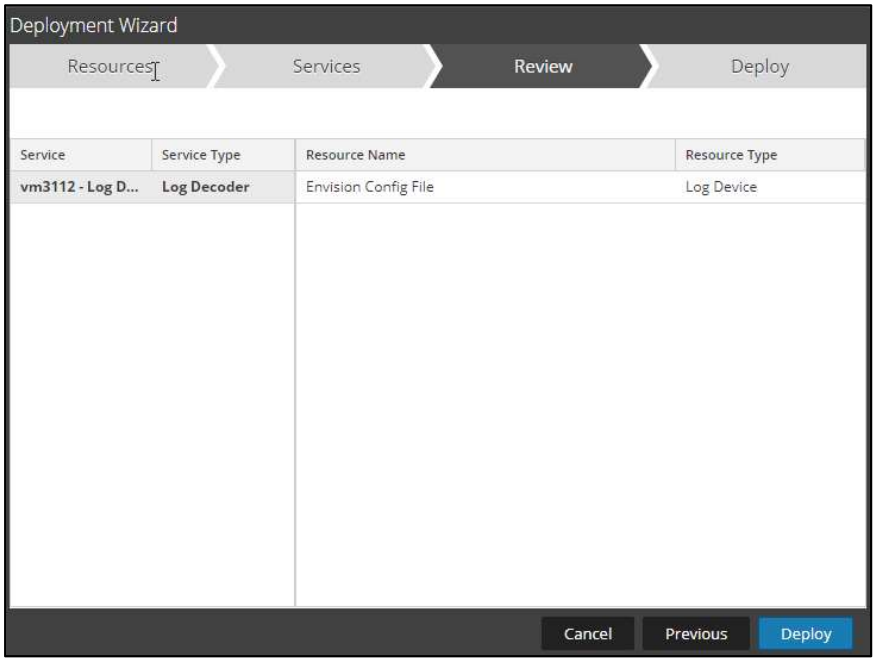


7. Select the **Log Decoder** and select **Next**.

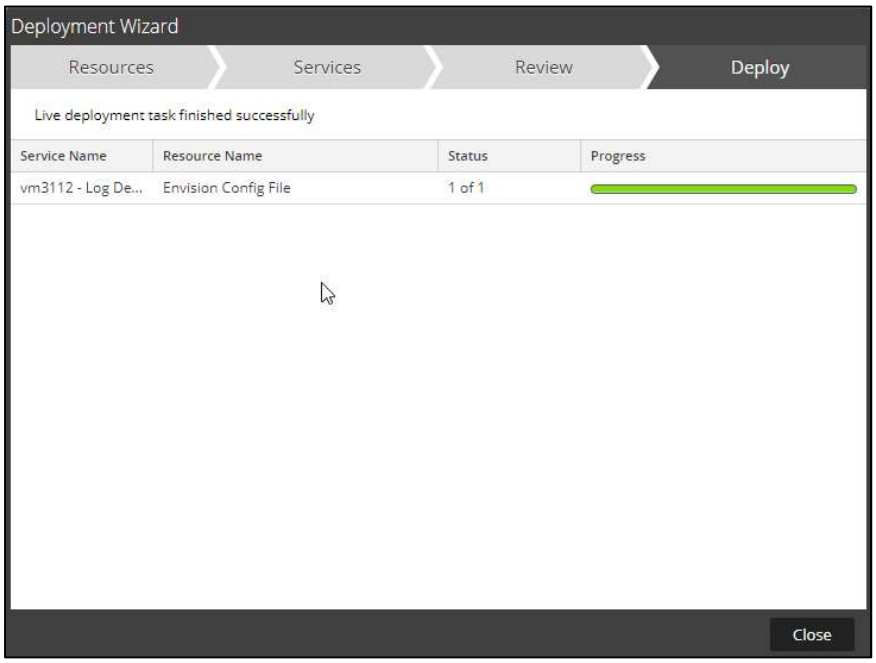


Important: In an environment with multiple Log Decoders, deploy the Envision Config File to each Log Decoder in your network.

8. Select **Deploy**.



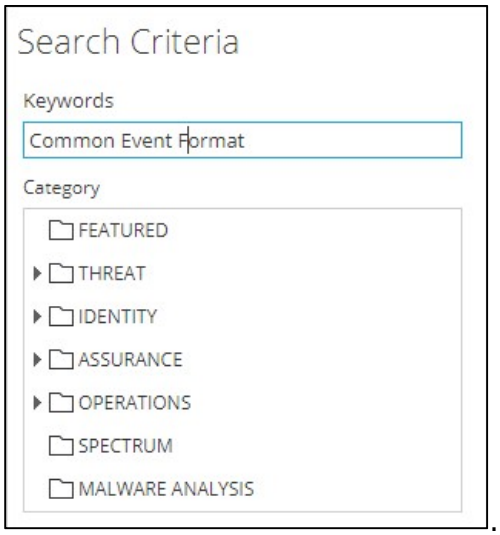
9. Select **Close**, to complete the deployment of the Envision Config file.



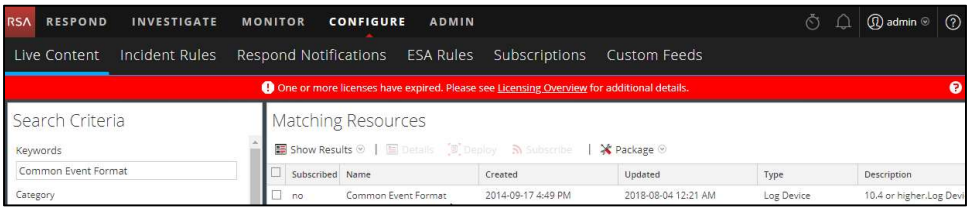
Deploy the Common Event Format

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

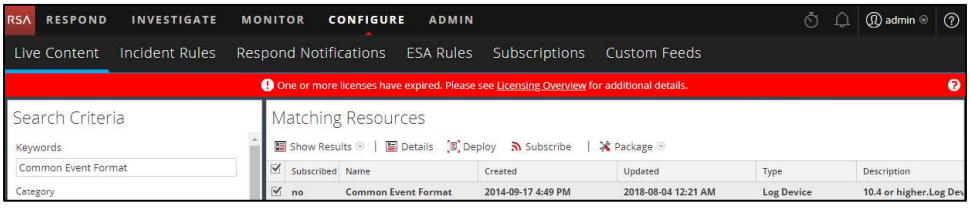
1. From the NetWitness menu, select **Live > Search**.
2. In the keywords field, enter: **Common Event Format**



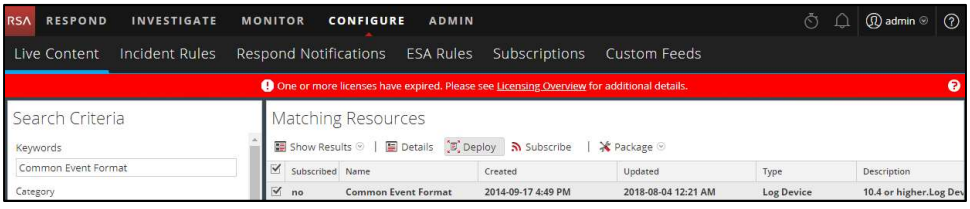
3. RSA NetWitness will display the **Common Event Format** in Matching Resources.



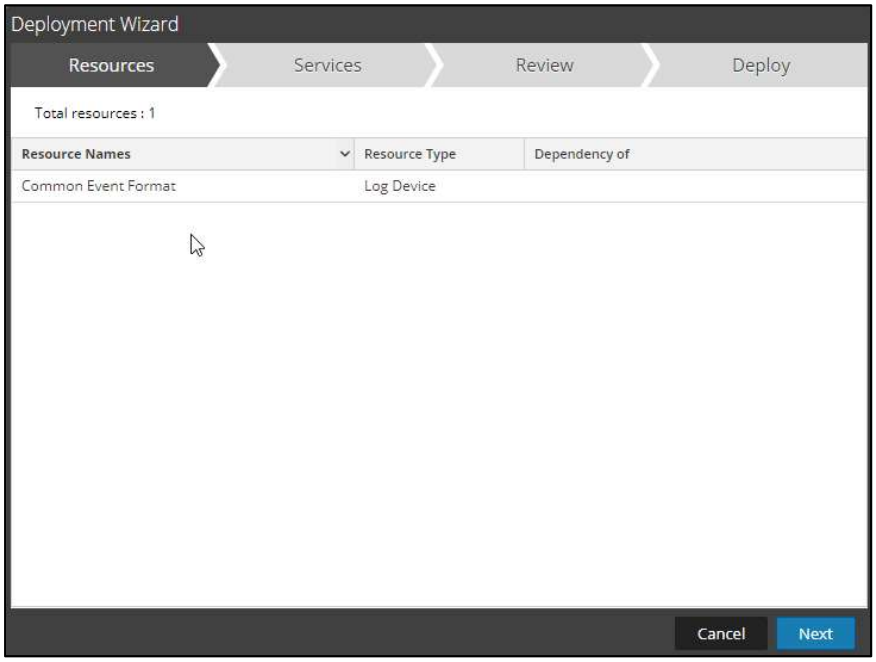
4. Select the checkbox next to **Common Event Format**.



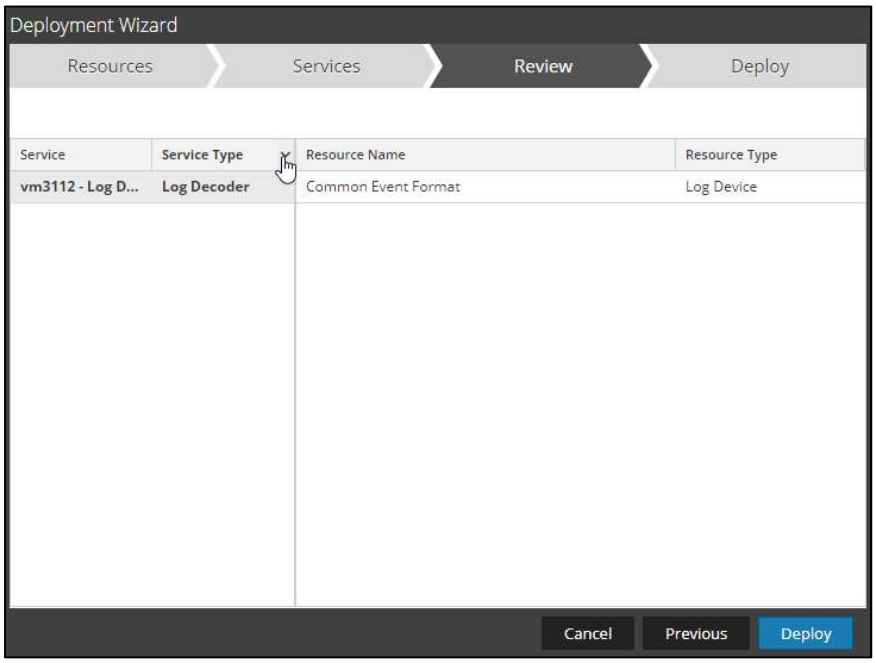
5. Click **Deploy** in the menu bar.



6. Select **Next**.

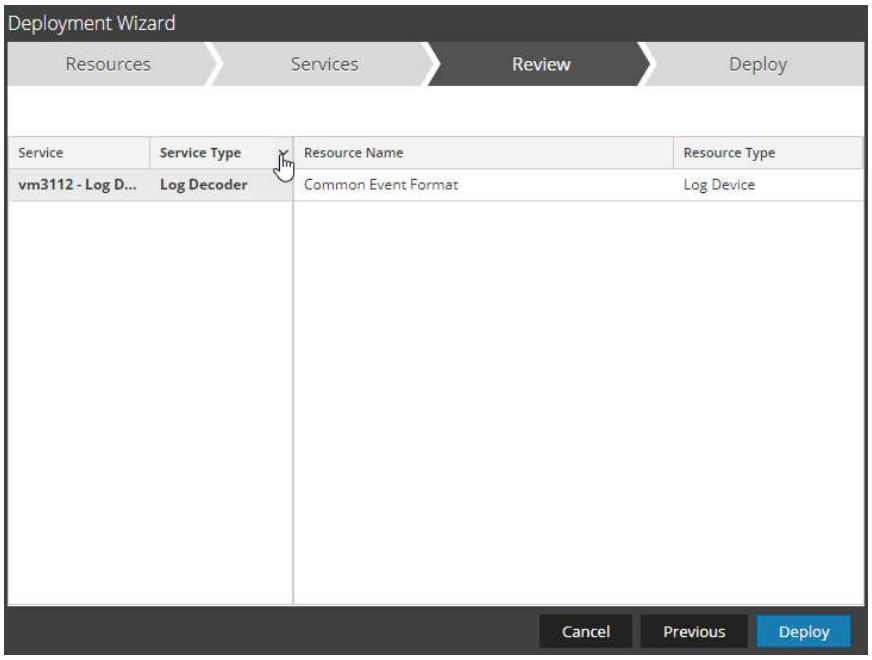


7. Select the **Log Decoder** and Select **Next**.

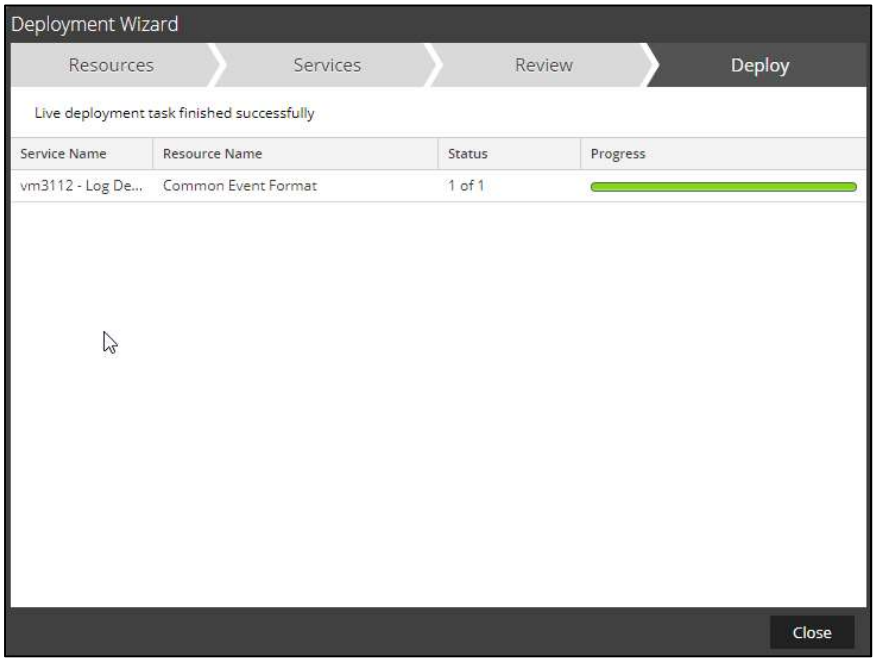


! Important: In an environment with multiple Log Decoders, deploy the Common Event Format to each Log Decoder in your network.

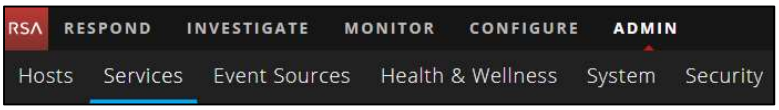
8. Select **Deploy**.



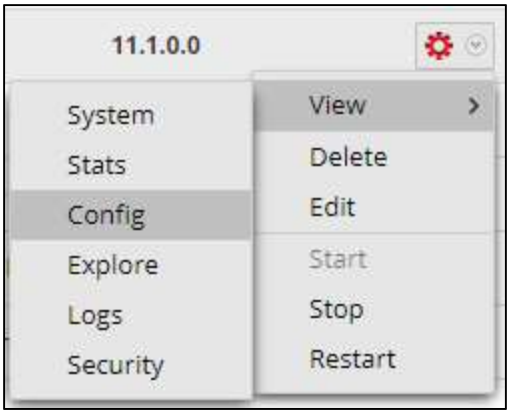
9. Select **Close**, to complete the deployment of the Common Event Format.



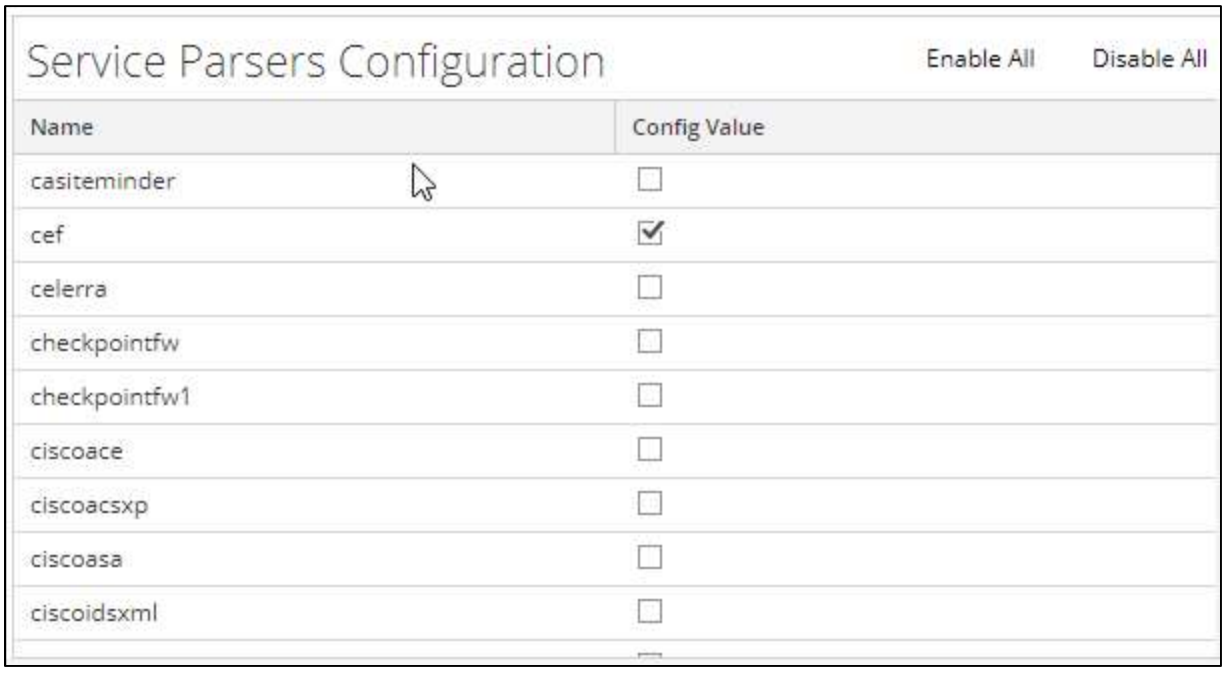
10. Ensure that the CEF Parser is enabled on the Log Decoder(s) by selecting **Admin > Services** from the NetWitness Dashboard.



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**.



13. Restart the **Log Decoder services**.

Edit the Common Event Format to collect NXLog 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:

```
<MESSAGE
        id1="NXLog_NXLog"
        id2="NXLog_NXLog"
        eventcategory="1612000000"
    functions="&lt;@msg:*PARMVAL($MSG)&gt;&lt;@event_time:*EVNTTIME($MSG,'%R %F
%Z',event_time_string)&gt;&lt;@endtime:*EVNTTIME($MSG,'%W-%D-%G
%Z',param_endtime)&gt;&lt;@starttime:*EVNTTIME($MSG,'%W-%G-
%FT%Z',param_starttime)&gt;"
content="&lt;param_endtime&gt;&lt;param_starttime&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>
<!--
#
# cef-custom.xml Reference: https://community.rsa.com/docs/DOC-79189
#
--> cef-custom.xml

<VendorProducts>
       <Vendor2Device vendor="NXlog" product="NXLog Enterprise Edition"
device="NXLog_NXLog" group="Analysis"/>
</VendorProducts>

           <ExtensionKeys>
                    <ExtensionKey cefName="Keywords" metaName="Keywords"/>
                    <ExtensionKey cefName="Severity" metaName="Severity"/>
                    <ExtensionKey cefName="SeverityValue" metaName="SeverityValue"/>
                    <ExtensionKey cefName="externalID" metaName="externalID"/>
                    <ExtensionKey cefName="SourceName" metaName="SourceName"/>
                    <ExtensionKey cefName="ProviderGuid" metaName="ProviderGuid"/>
                    <ExtensionKey cefName="TaskValue" metaName="TaskValue"/>
                    <ExtensionKey cefName="AccountName" metaName="AccountName"/>
                    <ExtensionKey cefName="Domain" metaName="Domain"/>
                    <ExtensionKey cefName="UserID" metaName="UserID"/>
                    <ExtensionKey cefName="AccountType" metaName="AccountType"/>
                    <ExtensionKey cefName="Namespace" metaName="Namespace"/>
                    <ExtensionKey cefName="payload" metaName="cs_payload"/>
                    <ExtensionKey cefName="ConnectionType" metaName="ConnectionType"/>
                    <ExtensionKey cefName="OpcodeValue" metaName="OpcodeValue"/>
                    <ExtensionKey cefName="RecordNumber" metaName="RecordNumber"/>
                    <ExtensionKey cefName="ExecutionProcessID"
metaName="ExecutionProcessID"/>
                    <ExtensionKey cefName="ExecutionThreadID"
metaName="ExecutionThreadID"/>
                    <ExtensionKey cefName="param1" metaName="param1"/>
                    <ExtensionKey cefName="param2" metaName="param2"/>
                    <ExtensionKey cefName="param3" metaName="param3"/>
                    <ExtensionKey cefName="param4" metaName="param4"/>
                    <ExtensionKey cefName="SourceModuleName"
metaName="SourceModuleName"/>
                    <ExtensionKey cefName="SourceModuleType"
metaName="SourceModuleType"/>
                    <ExtensionKey cefName="EventReceivedTime"
metaName="param_starttime"/>
                    <ExtensionKey cefName="msg" metaName="msg">
                           <device2meta device="trendmicrodsa" metaName="info"/>
                           <device2meta device="NXLog_NXLog" metaName="info"/>
                    </ExtensionKey>
             </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="severity" nwName="severity" flags="None"
format="Text"/>
       <mapping envisionName="Keywords" nwName="Keywords" flags="None"
format="Text"/>
       <mapping envisionName="Severity" nwName="Severity" flags="None"
format="Text"/>
       <mapping envisionName="SeverityValue" nwName="SeverityValue" flags="None"
format="Text"/>
       <mapping envisionName="externalID" nwName="externalID" flags="None"
format="Text"/>

       <mapping envisionName="dvcpid" nwName="dvcpid" flags="None"
format="Text"/>
       <mapping envisionName="hardware_id" nwName="hardware.id" flags="None"
format="Text"/>
       <mapping envisionName="SourceName" nwName="SourceName" flags="None"
format="Text"/>
       <mapping envisionName="ProviderGuid" nwName="ProviderGuid" flags="None"
format="Text"/>
       <mapping envisionName="TaskValue" nwName="TaskValue" flags="None"
format="Text"/>
       <mapping envisionName="OpcodeValue" nwName="OpcodeValue" flags="None"
format="Text"/>
       <mapping envisionName="AccountName" nwName="AccountName" flags="None"
format="Text"/>
       <mapping envisionName="Domain" nwName="domain" flags="None"
format="Text"/>
       <mapping envisionName="UserID" nwName="UserID" flags="None"
format="Text"/>
       <mapping envisionName="AccountType" nwName="AccountType" flags="None"
format="Text"/>
       <mapping envisionName="msg" nwName="msg" flags="None" format="Text"/>
```

```
        <mapping envisionName="Namespace" nwName="Namespace" flags="None"
format="Text"/>
        <mapping envisionName="cs_payload" nwName="cs_payload" flags="None"
format="UInt32"/>
        <mapping envisionName="ConnectionType" nwName="ConnectionType"
flags="None" format="UInt32"/>
        <mapping envisionName="RecordNumber" nwName="RecordNumber" flags="None"
format="Text"/>
        <mapping envisionName="ExecProcID" nwName="ExecProcID" flags="None"
format="Text"/>
        <mapping envisionName="ExecThreadID" nwName="ExecThreadID" flags="None"
format="Text"/>
        <mapping envisionName="cs_devfacility" nwName="deviceFacility"
flags="None" format="Text"/>
        <mapping envisionName="info" nwName="info" flags="None" format="Text"/>
        <mapping envisionName="param1" nwName="param1" flags="None"
format="Text"/>
        <mapping envisionName="param2" nwName="param2" flags="None"
format="Text"/>
        <mapping envisionName="param3" nwName="param3" flags="None"
format="Text"/>
        <mapping envisionName="param4" nwName="param4" flags="None"
format="Text"/>
        <mapping envisionName="SourceModuleName" nwName="SourceModuleName"
flags="None" format="Text"/>
        <mapping envisionName="SourceModuleType" nwName="SourceModuleType"
flags="None" format="Text"/>
        <mapping envisionName="param_endtime" nwName="end" flags="None"
format="TimeT"/>
        <mapping envisionName="param_starttime" nwName="start" flags="none"
format="TimeT"/>

</mappings>
```

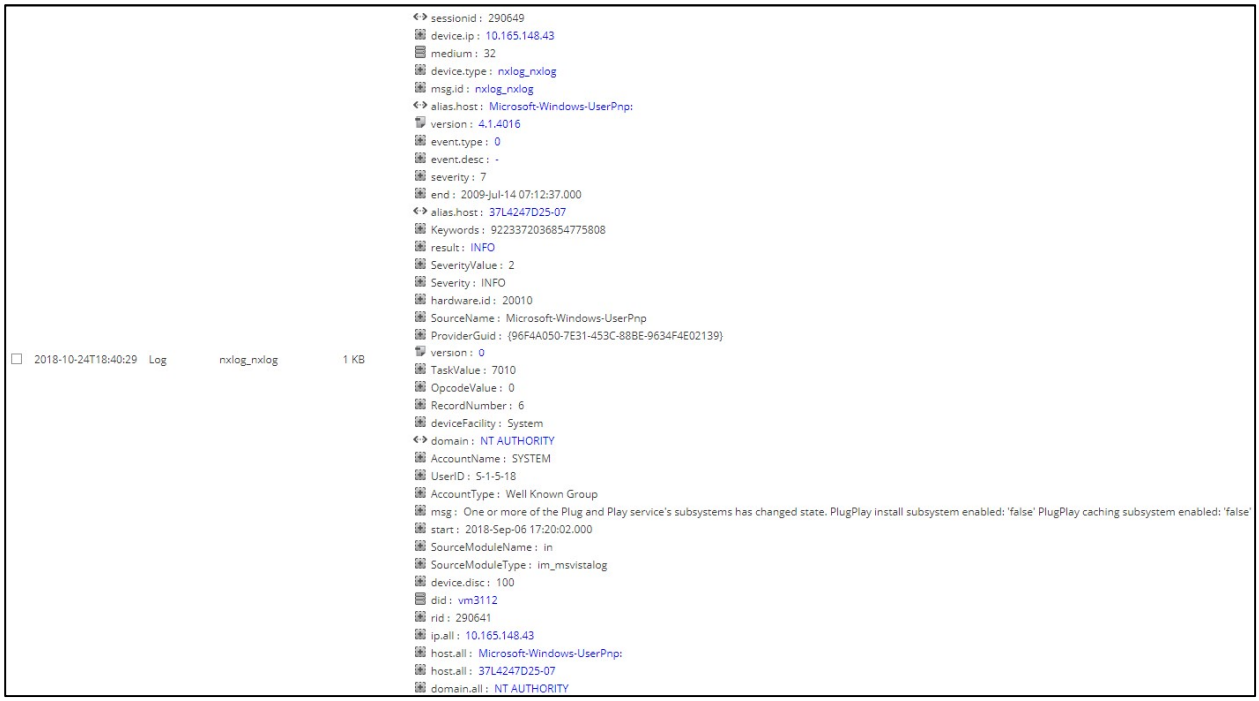
4. Restart the **Log Decoder services** to begin log collection.

> **Important: Other log sources supported by NXLog generate a different set of fields which may not be captured by NetWitness unless the required mapping is configured in the NetWitness parser customizations.**

Custom fields generated by the various NXLog input modules are listed in the NXLog Enterprise Edition User Gude. For example, the im_checkpoint input module generates the fields listed at **https://nxlog.co/documentation/nxlog-user-guide#im_checkpoint_fields**.

Some modules can generate fields which are not known in advance and depends on the logs source configuration, e.g. *im_odbc* generates fields based on the column names in the database table.

Example NXLog Collection from NetWitness Investigator:



Certification Checklist for RSA NetWitness

Date Tested: November 19, 2018

Certification Environment		
Product Name	**Version Information**	**Operating System**
RSA NetWitness	11	Virtual Appliance
NXLog Enterprise Edition	4.1	Microsoft Windows Server 2016

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	✓

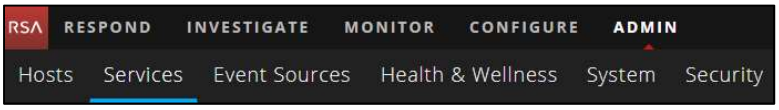
✓ = Pass ✗ = Fail N/A = Non-Available Function

Appendix

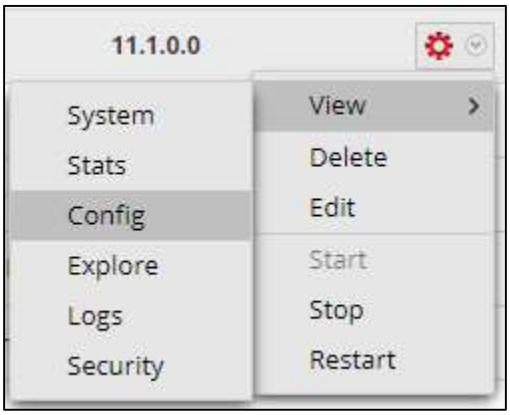
NetWitness Disable the Common Event Format Parser

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

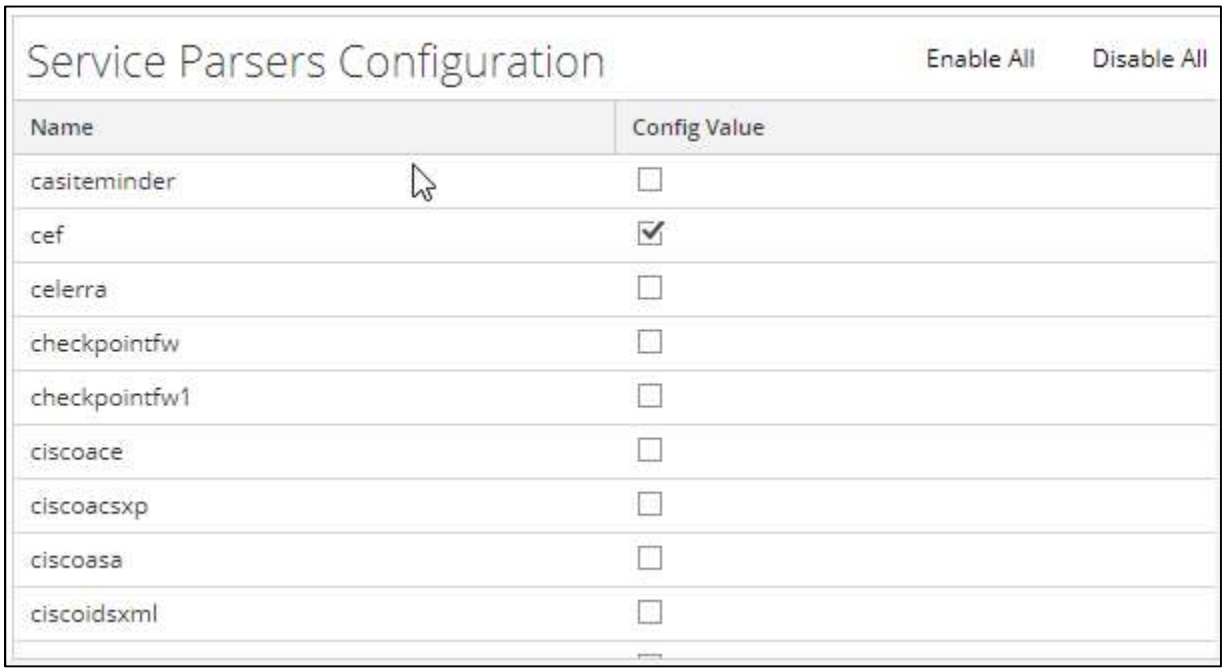
1. Select the NetWitness **Admin > Services**.



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



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

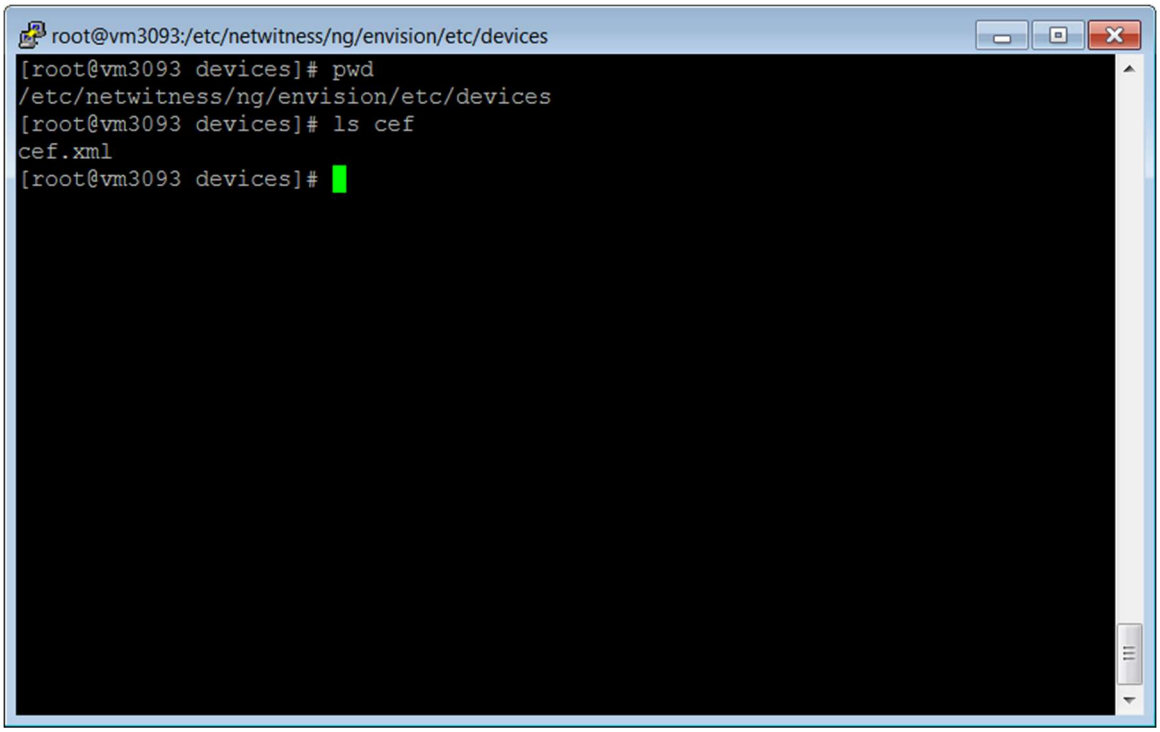


4. Click **Apply** to save settings.

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.



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