RSA® NETWITNESS®
Logs
Implementation Guide

ENDGAME.
Endgame 2.5.4

Daniel R. Pintal, RSA Partner Engineering
Last Modified: February 23, 2018

RSA READY

ENDGAME.

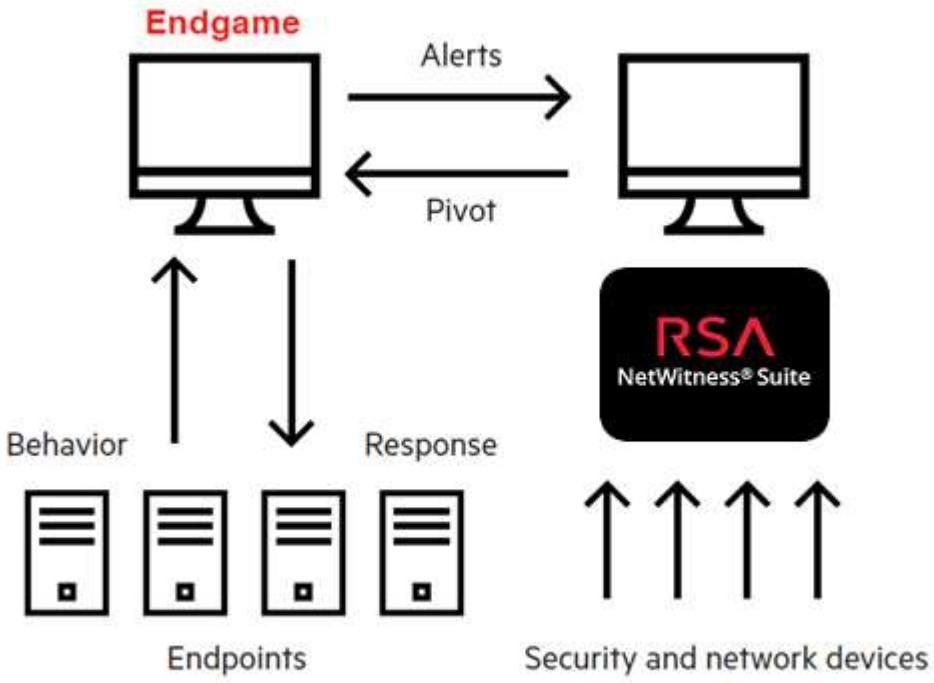
Solution Summary

Endgame is the only endpoint protection platform that provides prevention, detection, and response capabilities to stop targeted attacks before damage and loss, with the people you already have. Their single and autonomous agent replaces AV, NGAV, EDR, exploit protection, and incident response agents to reduce enterprise cost and complexity.

Endgame customers, both commercial and federal, have chosen them for their comprehensive scope of protections, unprecedented speed to stop all attacks, and enhanced skills the platform provides to get the job done with existing people.

Together with RSA NetWitness, the integration provides SOC and NOC Admins' with a solution to actively hunt, monitor and notify personnel about threats that have or are attempting to breach network security within the network infrastructure.

RSA NetWitness Features	
ENDGAME. Endgame 2.5.4	
Integration package name	Common Event Format
Device display name within NetWitness	endgame_endgame
Event source class	Endpoint Security
Collection method	Syslog CEF



RSA READY

ENDGAME.

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
2/23/2018	Initial support for ENDGAME. Endgame 2.5.4.

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.

RSA READY

ENDGAME.

Partner Product Configuration

Before You Begin

This section provides instructions for configuring ENDGAME. 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 Endgame 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 Endgame. Endgame 2.5.4 is properly configured and secured before deploying to a production environment. For more information, please refer to the Endgame. Endgame 2.5.4 documentation or website.**

Endgame RSA NetWitness Connector Configuration Guide

Configuration

Within the Endgame application, the ALERT MANAGEMENT page enables you to automatically export alerts to RSA NetWitness.

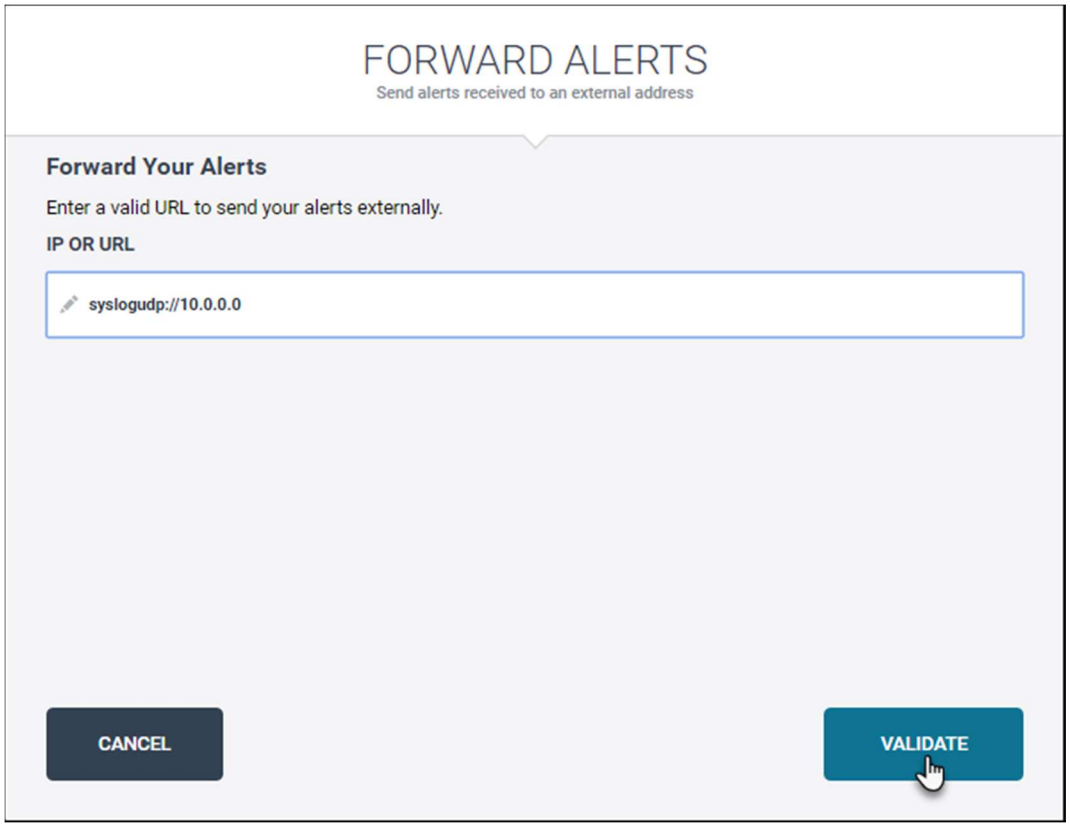
You can forward alerts via the following protocols: User Datagram Protocol (UDP) or Transmission Control Protocol (TCP), or SSL-encrypted TCP. Alerts are exported using the Common Event Format (CEF).

Perform the following steps to start sending alerts to RSA NetWitness.

RSA READY

Configuring TCP or UDP (Plaintext)

1. Click **FORWARD ALERTS**.
2. In the text box, type the **IP address** or **URL** of your syslog listener.
   - For UDP, prepend the url with: syslogtcp://
   - For TCP, prepend the url with: syslogtcp://
   - For SSL-encrypted, please skip to the "configuring TCP (SSL)" section below.
3. Click **VALIDATE** to verify the address is valid. If it is valid, a "Success setting alert export location" message appears. If it isn't valid, a "Failure setting alert export location" message appears.



Configuring TCP (SSL)

To establish an SSL-enabled connection between ENDGAME. and RSA NetWitness, you will need to provide an ssl cert and the associated cert chain.

To perform the install, ssh into the ENDGAME. server and run the following commands, replacing the contents of <> with appropriate values:

```
cd /opt/endgame/
python alert_export_cmd.py --url syslogssl://<LoggingServer_IP>:<port number> --keyfile /<path of machine key>/machine-key.pem --certfile /<path of machine certificate>/machine-cert.pem --cert_reqs 2 --ca_certs /<path of certificate authority>/ca.pem
```

Test Export Location

To verify that the ENDGAME. server can communicate with RSA NetWitness, we provide a validation feature to support troubleshooting. Within the ENDGAME. application, the ALERT MANAGEMENT page enables you to automatically test the connection at any time.

To run the test:

1. Click **TEST EXPORT LOCATION**. If the export was successful, an "Export Validation Successful" message appears in the upper right corner of the page.
2. Click **View Details** to view the test message that was sent to RSA NetWitness.
3. Within RSA NetWitness Investigator, confirm that the test message was received.
4. From within Endgame, click **Clear** to dismiss the message.

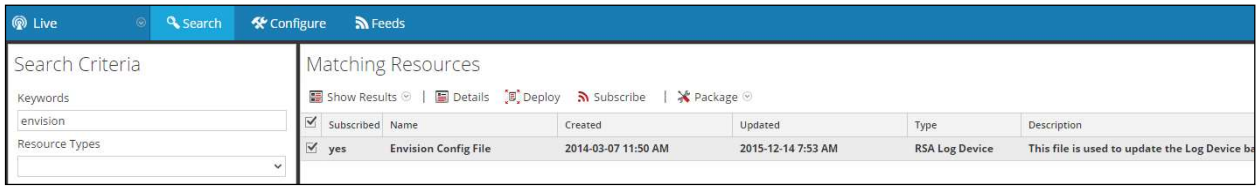
RSA NetWitness Configuration

Deploy the enVision Config File

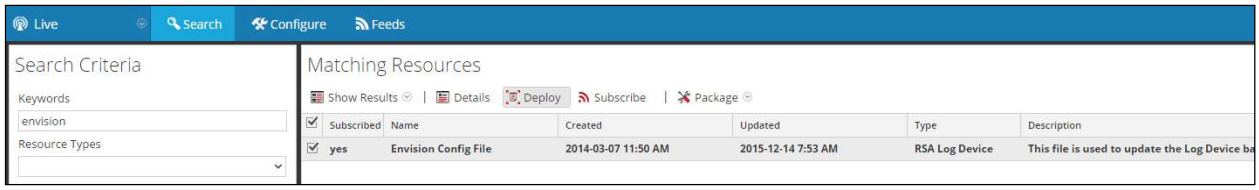
In order to use the RSA Common Event Format, you must first deploy the *enVision Config File* from the **NetWitness Live** module. Log into NetWitness and perform the following actions:

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

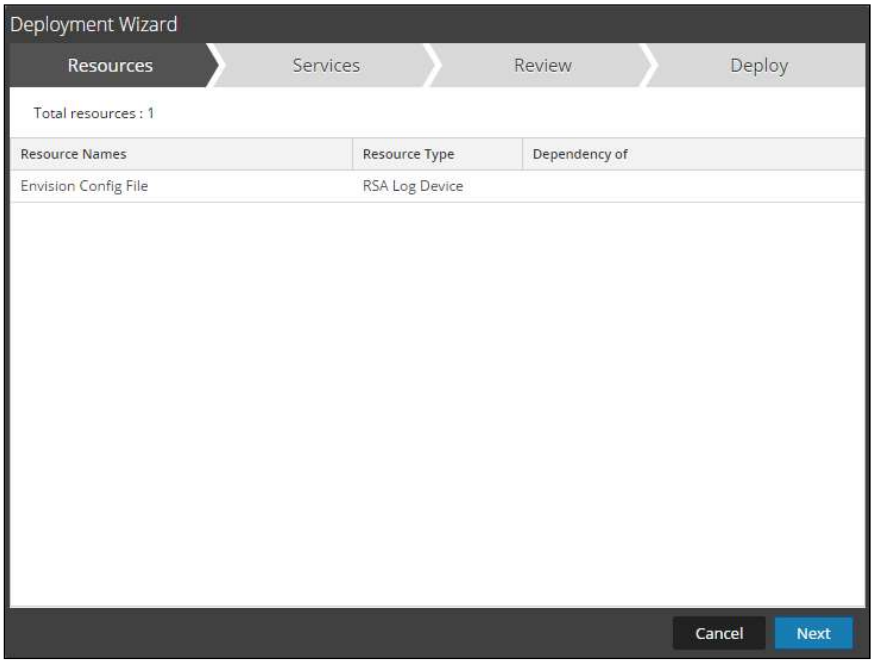
1. From the NetWitness menu, select **Live > Search**.
2. In the keywords field, enter: **enVision**.
3. NetWitness will display the **Envision Config File** in Matching Resources.
4. Select the checkbox next to **Envision Config File**.



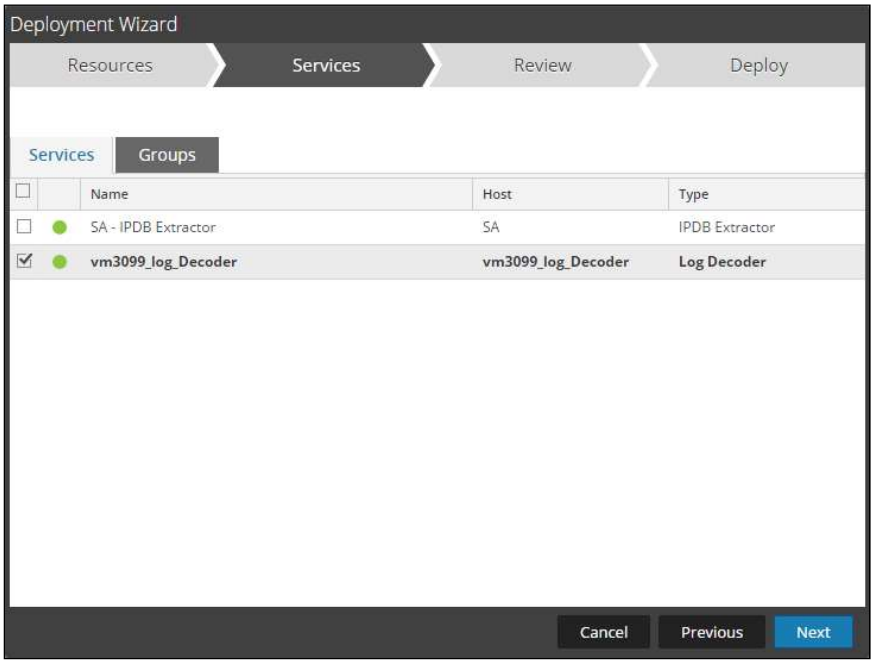
5. Click **Deploy** in the menu bar.



6. Select **Next**.

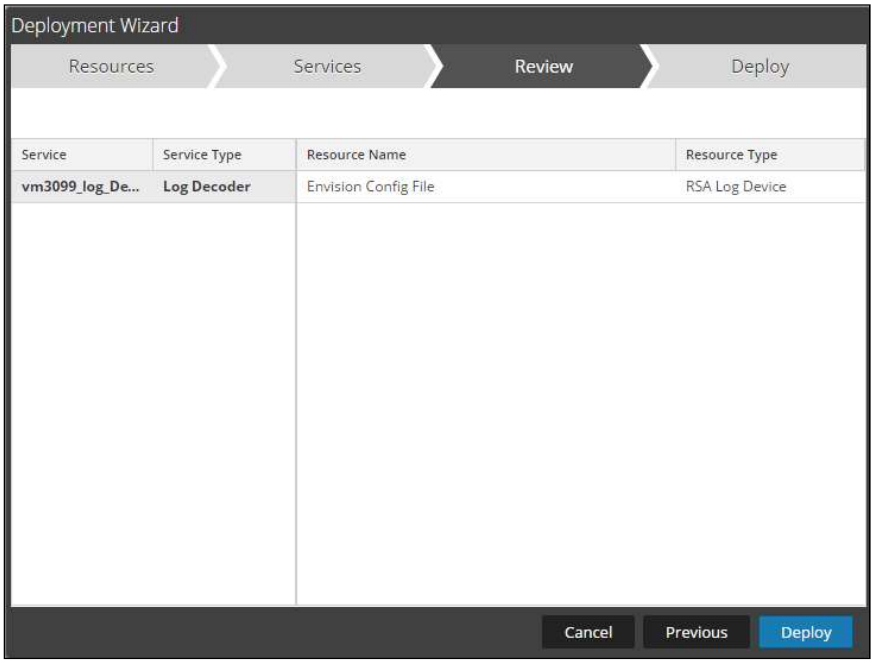


7. Select the **Log Decoder** and select **Next**.

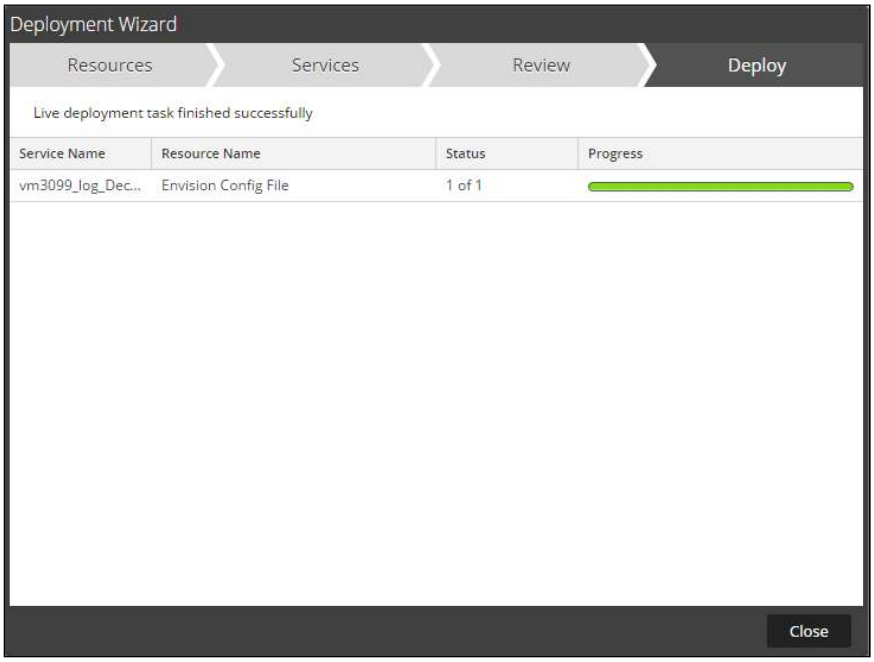


! Important: In an environment with multiple Log Decoders, deploy the Envision Config File to each Log Decoder in your network.

8. Select **Deploy**.



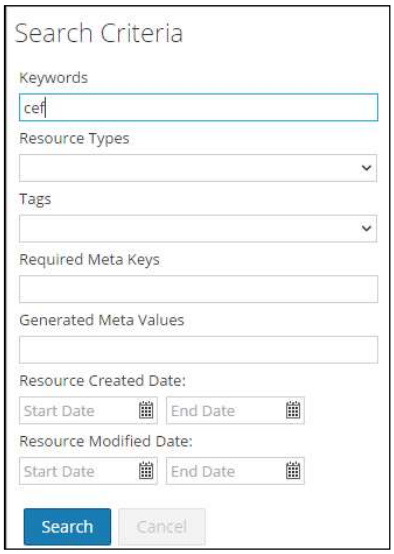
9. Select **Close**, to complete the deployment of the Envision Config file.



Deploy the Common Event Format

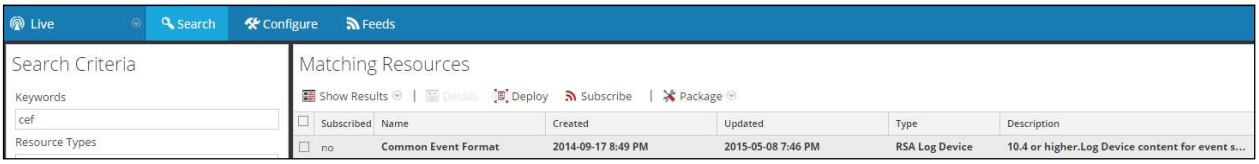
Next, deploy the *Common Event Format file* from the **NetWitness Live** module. Log into NetWitness and perform the following actions:

1. From the NetWitness menu, select **Live > Search**.
2. In the Keywords field, enter: **CEF**

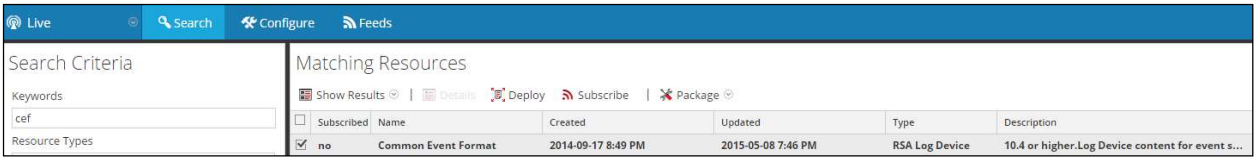


.

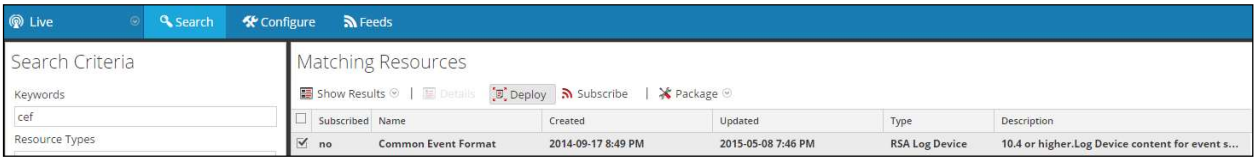
3. RSA NetWitness will display the **Common Event Format** in Matching Resources.



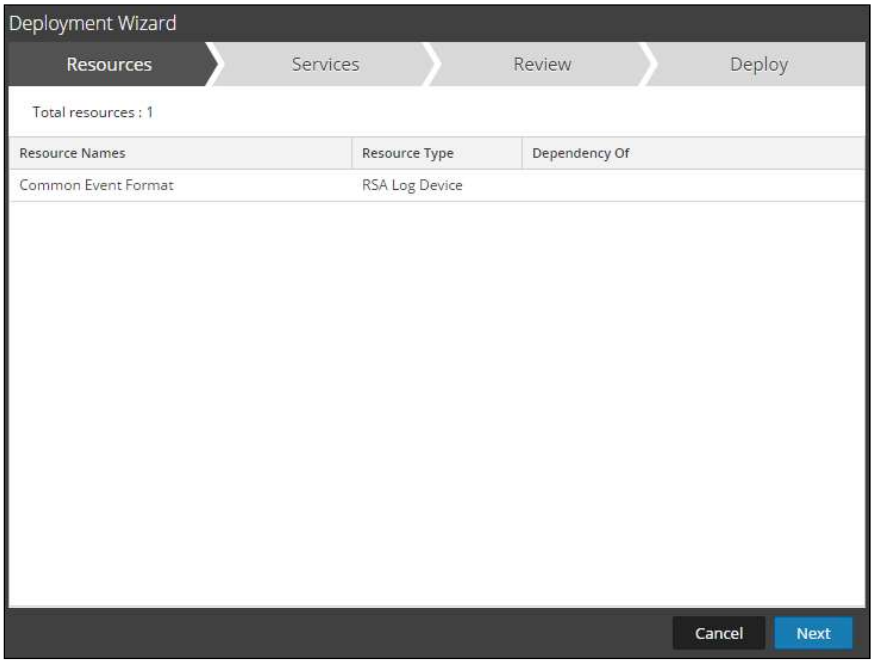
4. Select the checkbox next to **Common Event Format**.



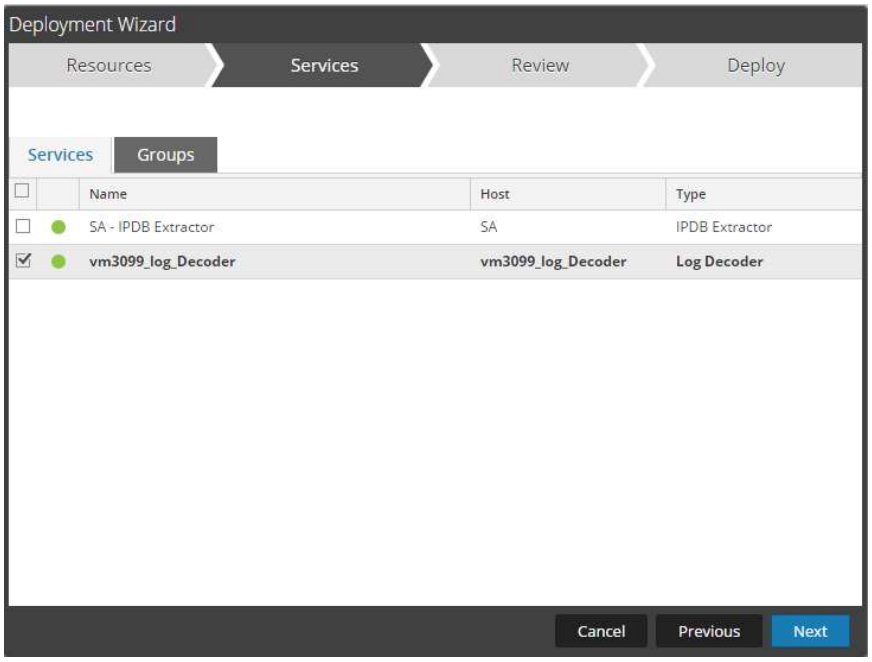
5. Click **Deploy** in the menu bar.



6. Select **Next**.

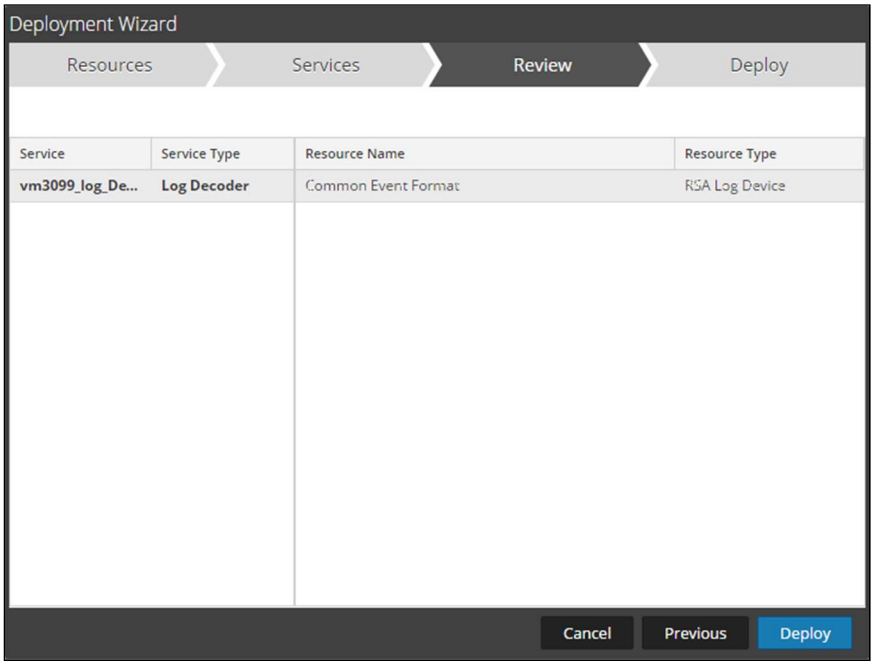


7. Select the **Log Decoder** and Select **Next**.

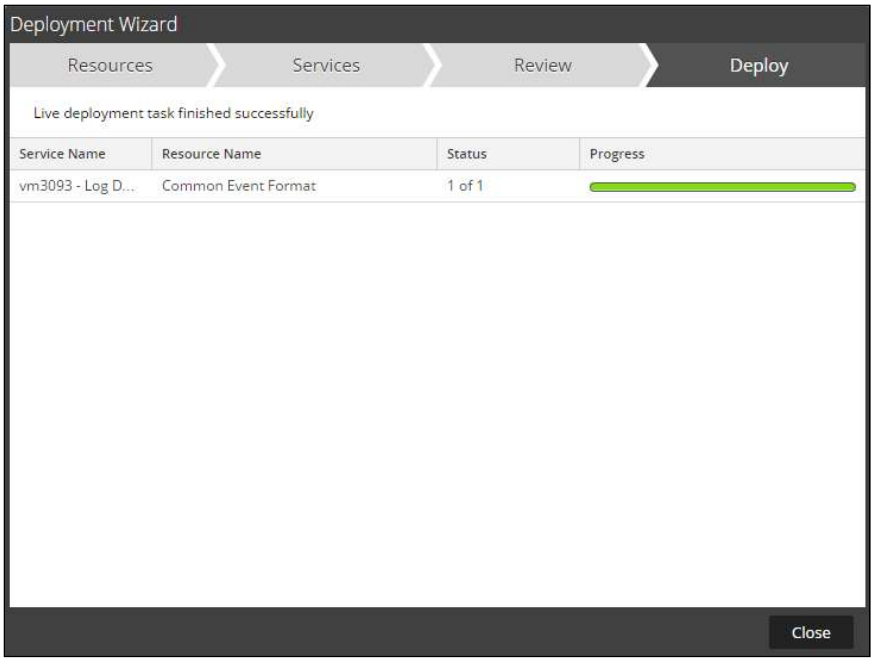


Important: In an environment with multiple Log Decoders, deploy the Common Event Format to each Log Decoder in your network.

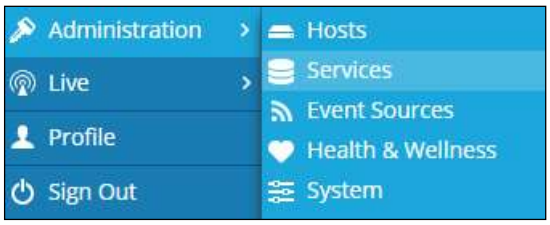
8. Select **Deploy**.



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 **Administration, 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 cef under Config Value within the Service Parsers Configuration and select **Apply**.

Edit the Common Event Format to collect Endgame 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. 10.6.x

```
<MESSAGE
                level="4"
                parse="1"
                parsedefvalue="1"
                tableid="74"
                id1="endgame_endgame"
                id2="endgame_endgame"
                eventcategory="1612000000"

        content="&lt;@event_name:*HDR(event_description)&gt;&lt;@msg:*PARMVAL($MS
G)&gt;&lt;@endtime:*EVNTTIME($MSG,'%B %D %W %Z
%L',param_event_time)&gt;&lt;@starttime:*EVNTTIME($MSG,'%B %D %W %Z
%L',param_starttime)&gt;&lt;msghold&gt;&lt;param_event_time&gt;&lt;param_startt
ime&gt;" />
```

Example. 11.x

```
<MESSAGE
        id1="endgame_endgame"
        id2="endgame_endgame"
        eventcategory="1612000000"
        functions="&lt;@event_name:*HDR(event_description)&gt;&lt;@msg:*PARMVAL($
MSG)&gt;&lt;@endtime:*EVNTTIME($MSG,'%B %D %W %Z
%L',param_event_time)&gt;&lt;@starttime:*EVNTTIME($MSG,'%B %D %W %Z
%L',param_starttime)&gt;"
        content="&lt;msghold&gt;&lt;param_event_time&gt;&lt;param_starttime&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.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<DEVICEMESSAGES>
<!--
#
# cef-custom.xml Reference: https://community.rsa.com/docs/DOC-79189
#
#
<MESSAGE
       id1="endgame_endgame"
       id2="endgame_endgame"
       eventcategory="1612000000"
       functions="&lt;@event_name:*HDR(event_description)&gt;&lt;@msg:*PARMVAL($
MSG)&gt;&lt;@endtime:*EVNTTIME($MSG,'%B %D %W %Z
%L',param_event_time)&gt;&lt;@starttime:*EVNTTIME($MSG,'%B %D %W %Z
%L',param_starttime)&gt;"
       content="&lt;msghold&gt;&lt;param_event_time&gt;&lt;param_starttime&gt;"
/>
-->

<VendorProducts>
    <Vendor2Device vendor="Endgame" product="Endgame" device="endgame_endgame"
group="Endpoint Security"/>
</VendorProducts>

          <ExtensionKeys>
             <ExtensionKey cefName="Version" metaName="version"/>
             <ExtensionKey cefName="level" metaName="severity"/>

             <ExtensionKey cefName="cn1" metaName="cn_fld">
                    <device2meta device="trendmicrods" metaName="hostid"
label="Host ID"/>
                    <device2meta device="trendmicrodsa" metaName="hostid"
label="Host ID"/>
                    <device2meta device="mcafeewg" metaName="result"
label="Block Reason"/>
                    <device2meta device="endgame_endgame" metaName="tid"/>

             </ExtensionKey>
             <ExtensionKey cefName="cn1Label" metaName="cs_fld"/>

             <ExtensionKey cefName="cs1" metaName="cs_fld" >
                    <device2meta device="trendmicrodsa" metaName="context"/>
                    <device2meta device="bluecat" metaName="action"
label="query"/>
                    <device2meta device="websense" metaName="policyname"
label="Policy"/>
                    <device2meta device="mcafeewg" metaName="virusname"
label="Virus Name"/>
                    <device2meta device="bit9" metaName="checksum" label="File
Hash"/>
                    <device2meta device="mcafeereconnex"
metaName="policyname"/>
```

```
                        <device2meta device="endgame_endgame"
metaName="machineId"/>
                </ExtensionKey>
                <ExtensionKey cefName="cs1Label" metaName="cs_fld" />

                <ExtensionKey cefName="cs2" metaName="cs_fld">
                        <device2meta device="bit9" metaName="v_instafname"
label="installerFilename"/>
                        <device2meta device="endgame_endgame"
metaName="endgameUrl"/>
                </ExtensionKey>
                <ExtensionKey cefName="cs2Label" metaName="cs_fld"/>

                <ExtensionKey cefName="cs3" metaName="cs_fld">
                        <device2meta device="websense" metaName="content_type"
label="ContentType"/>
                        <device2meta device="bit9" metaName="policyname"/>
                        <device2meta device="mcafeereconnex"
metaName="content_type"/>
                        <device2meta device="endgame_endgame"
metaName="endgameType"/>
                </ExtensionKey>
                <ExtensionKey cefName="cs3Label" metaName="cs_fld"/>

                <ExtensionKey cefName="dpid" metaName="dpid"/>
                <ExtensionKey cefName="externalId" metaName="hardware_id"/>
                <ExtensionKey cefName="msg" metaName="msg"/>

                </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="starttime" nwName="start" flags="None"
      format="TimeT" envisionDisplayName="StartTime"/>
            <mapping envisionName="endtime" nwName="endtime" flags="None"
      format="TimeT" envisionDisplayName="EndTime,rt,end"/>

            <mapping envisionName="tid" nwName="tid" flags="None"/>
            <mapping envisionName="machineId" nwName="machineId" flags="None"/>
            <mapping envisionName="endgameUrl" nwName="endgameUrl" flags="None"/>
            <mapping envisionName="endgameType" nwName="endgameType" flags="None"/>
            <mapping envisionName="version" nwName="version" flags="None"/>
            <mapping envisionName="severity" nwName="severity" flags="None"
      envisionDisplayName="Severity|SeverityLevel"/>
            <mapping envisionName="dpid" nwName="dpid" flags="None"/>
            <mapping envisionName="hardware_id" nwName="hardware.id" flags="None"/>
            <mapping envisionName="msg" nwName="msg" flags="None" format="Text"
      envisionDisplayName="Message"/>

      </mappings>
```

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

Endgame Collection Example within NetWitness Investigator:

Certification Checklist for RSA NetWitness

Date Tested: February 23, 2018

Certification Environment		
Product Name	**Version Information**	**Operating System**
RSA NetWitness	10.6.4 & 11	Virtual Appliance
ENDGAME. Endgame	2.5.4	Cloud Environment

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	✓

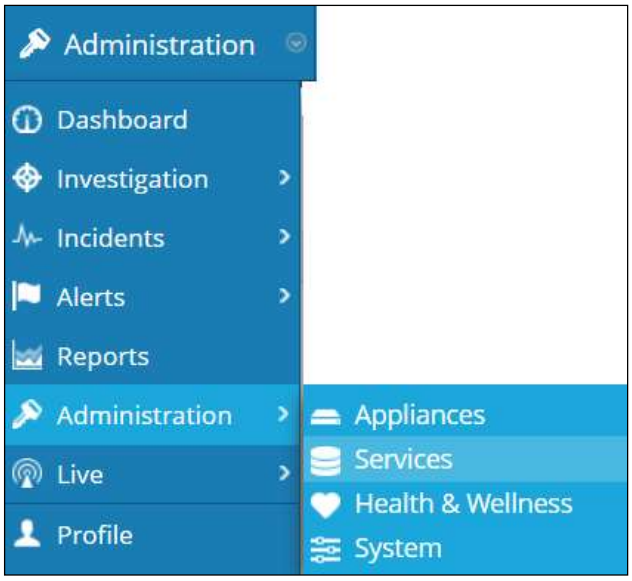
✓ = Pass ✗ = Fail N/A = Non-Available Function

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 **Administration > Services menu**.



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

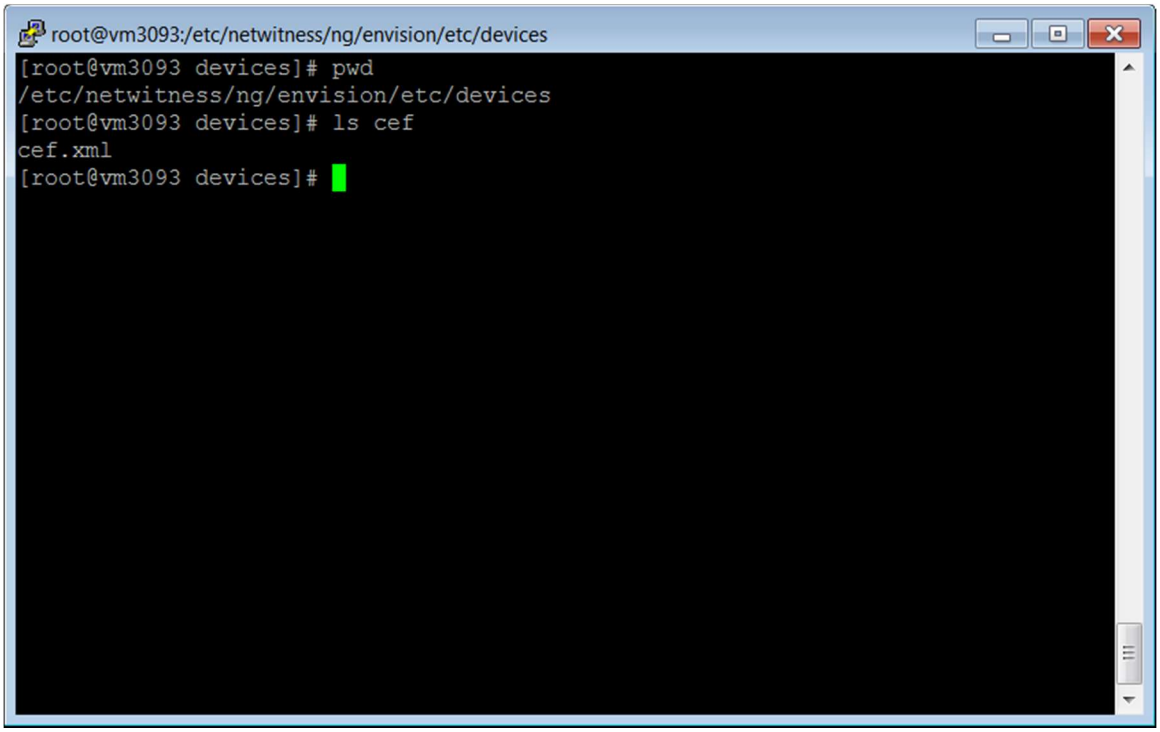
3. From the **Service Parses Configuration** window, scroll down to the CEF parser and uncheck the Config Value checkbox.

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.