NetWitness® Platform

DeepInspect Event Source Log Configuration Guide

DeepInspect

Event Source Product Information:

Vendor: DeepInspect

Event Source: DeepInspect

Versions: 2.2

NetWitness Product Information:

Supported On: NetWitness Platform 12.2 and higher

Event Source Log Parser: deepinspect

Collection Method: Syslog

Event Source Class.Subclass: ICS

Contact Information

NetWitness Community at https://community.netwitness.com contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

RSA and other trademarks are trademarks of RSA Security LLC or its affiliates ("RSA"). For a list of RSA trademarks, go to https://www.rsa.com/en-us/company/rsa-trademarks. Other trademarks are trademarks of their respective owners.

License Agreement

This software and the associated documentation are proprietary and confidential to RSA Security LLC or its affiliates are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability.

This software is subject to change without notice and should not be construed as a commitment by RSA.

Third-Party Licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed on the product documentation page on NetWitness Community. By using this product, a user of this product agrees to be fully bound by terms of the license agreements.

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Use, copying, and distribution of any RSA Security LLC or its affiliates ("RSA") software described in this publication requires an applicable software license.

RSA believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." RSA MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Miscellaneous

This product, this software, the associated documentations as well as the contents are subject to NetWitness' standard Terms and Conditions in effect as of the issuance date of this documentation and which can be found at https://www.netwitness.com/standard-form-agreements/.

January, 2024

Contents

Introduction to DeepInspect

DeepInspect's offers significant advantages in Detection and Asset Discovery in OT environment, seamlessly streamlining complex operations with a simple click.

Leveraging its robust asset identification capabilities, DeepInspect implements a sophisticated anomaly detection mechanism: it integrates a robust statistical module into its detection framework, analyzing protocol fields and data patterns to identify cybersecurity incidents through anomalies and deviations.

The system adopts a distinctive approach by monitoring variances from the golden image, promptly recognizing any divergence from the established baseline.

Extending its core functionalities, the solution seamlessly integrates with Netwitness, enhancing its capabilities within the cybersecurity framework.

One of DeepInspect's standout features is its adaptability to operate effectively in diverse environments, thanks to its adaptable nature, facilitated by type-approved hardware certified for both maritime and railway applications. This ensures strict compliance with the rigorous standards set within these industries.

Its data processing capability is embedded in all devices, allowing it to function optimally even in air-gapped settings. This distinctive quality ensures maximum reliability without necessitating an internet connection, making DeepInspect a resilient and dependable cybersecurity solution across various operational scenarios.

Configure DeepInspect Event Source

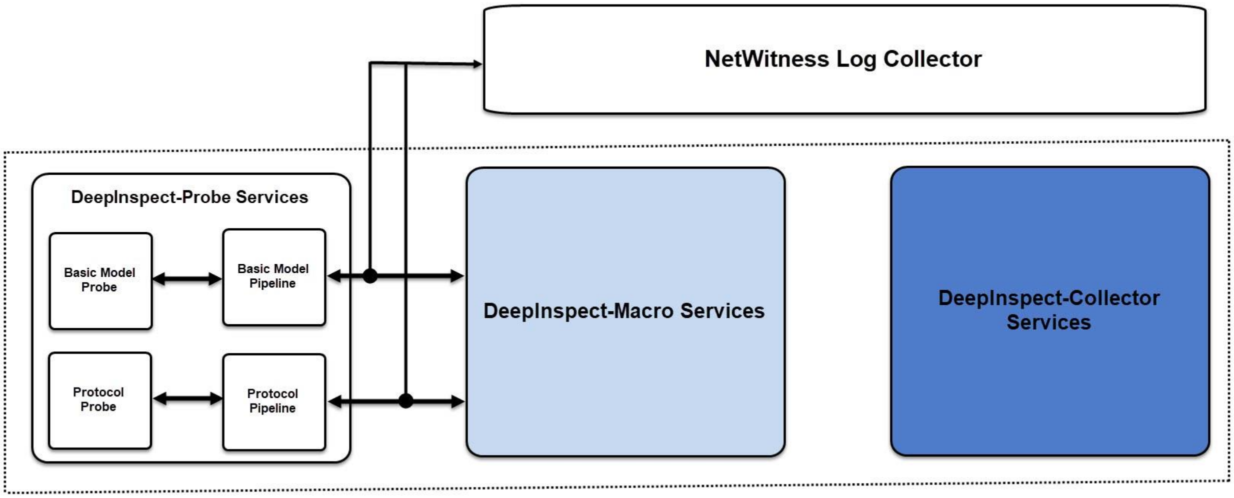
DeepInspect offers various below methods to enhance visibility in NetWitness.

- IT/OT Protocols Metadata forwarding to NetWitness Platform Log Collector
- IT/OT Protocols Raw Traffic forwarding to NetWitness Platform Packet Decoder
- IT/OT Alerting forward to NetWitness Platform Log Collector
- Asset Discovery and Anomaly Detection Feeds to NetWitness Server.

IMPORTANT: Please note that this integration is developed in collaboration with DeepInspect. For any inquiries regarding the configuration process on DeepInspect's end, please contact DeepInspect support.

IT/OT Protocols Metadata forwarding to NetWitness Platform Log Collector

DeepInspect probe pipelines have the ability to forward IT/OT protocols metadata. The platform comes with installation and configuration scripts that allow you to manage the reconfiguration of DeepInspect to forward specific pipelines metadata to NetWitness Log Collector. The NetWitness Log Collector will then manage a syslog server where all DeepInspect metadata will be accepted.



To manage the metadata forwarding, you will be requested to provide below parameter information:

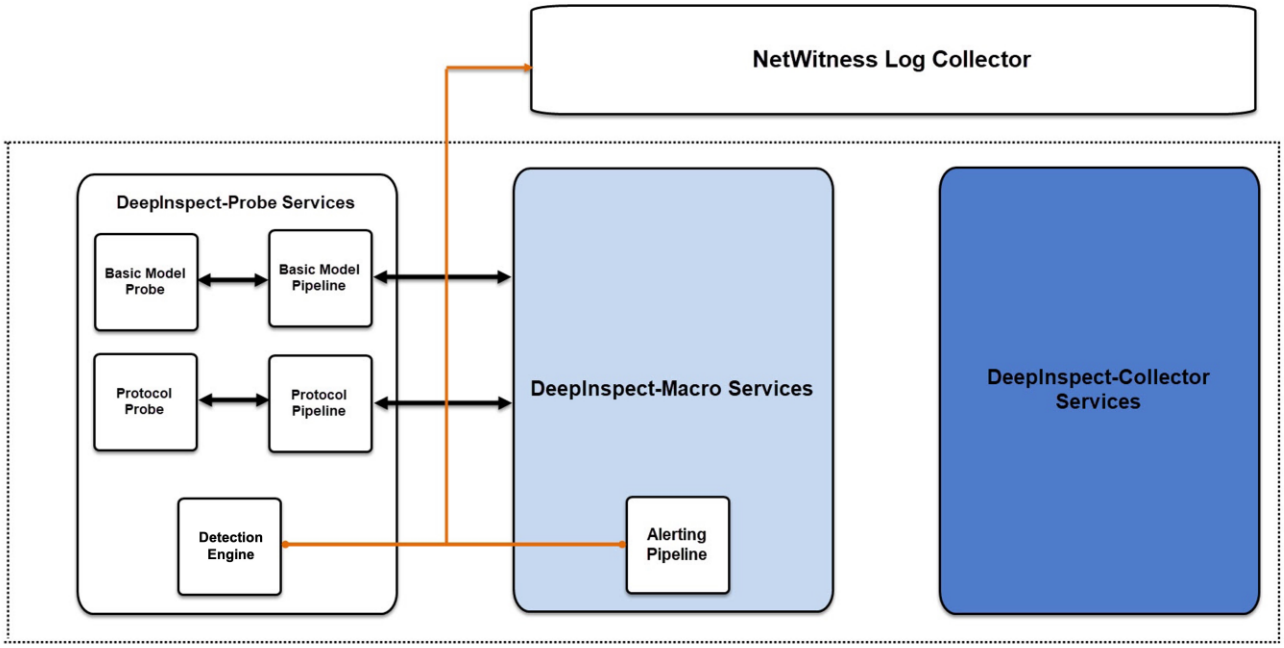
- **NetWitness Log Collector IP** – IP Address of the NetWitness Log Collector syslog server.
- **NetWitness Log Collector Port** – Port of the NetWitness Log Collector syslog server.

To configure metadata forwarding, request DeepInspect support on ***support@deepinspect.it*** for the `DeepInspect_RSAXDR_Integration_Meta` playbook. After receiving the playbook, execute following command on collector:

```
ansible-playbook deepinspect_rsaxdr_integration_meta.yml -k -K -u root
```

IT/OT Alerting forward to NetWitness Platform Log Collector

DeepInspect Detection Engine and macro pipelines have implemented alert forwarding, which allows users to receive anomalies directly from the traffic and outliers identified by models built in the back-end. DeepInspect also contains installation and configuration scripts to manage alert forwarding to NetWitness Log Collector.



To manage the DeepInspect alert forwarding, you will be requested to provide below parameter information:

- **DeepInspect-Macro name** – DeepInspect macro to be forwarded to NetWitness Log Collector.
- **NetWitness Log Collector IP** – IP Address of the NetWitness Log Collector syslog server.
- **NetWitness Log Collector Port** – Port of the NetWitness Log Collector syslog server.

To configure metadata forwarding, request DeepInspect support on ***support@deepinspect.it*** for the `DeepInspect_RSAXDR_Integration_Alerting` playbook. After receiving the playbook, execute following command on collector:

```
ansible-playbook deepinspect_rsaxdr_integration_alerting.yml -k -K -u root
```

Set Up the DeepInspect Event Source in NetWitness Platform

This section provides instructions for configuring the DeepInspect Suite with NetWitness Log Collector. 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 to install the required components.

All DeepInspect 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, DeepInspect Suite is properly configured and secured before deploying to a production environment. For more information, please refer to the DeepInspect Suite documentation or website.

Configure the NetWitness Platform

To set up the DeepInspect event source in NetWitness Platform, perform the following tasks:

- Ensure the Required Parser is Enabled.
- Customize the table-map and index-concentrator Files.

Ensure the Required Parser is Enabled

If you do not see your parser in the list while performing this procedure, you need to download it in NetWitness Platform Live.

To enable the required parser:

1. In the **NetWitness** menu, select (Admin) > **Services**.
2. In the **Services** grid, select a Log Decoder, and from the **Actions** () menu, choose **View** > **Config**.
3. In the **Service Parsers Configuration** panel, search for your event source, **deepinspect** and ensure that the **Config Value** field for your event source (**deepinspect**) is selected.

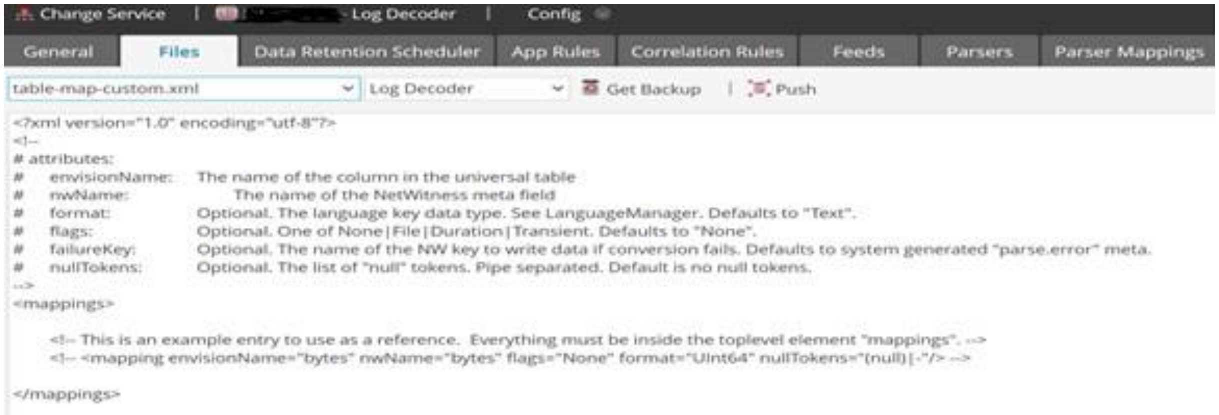
   The new device is listed under the Log Decoder(s) General Tab within the **Service Parsers Configuration**.

   Note: The required parser is deepinspect.

Customize the table-map and index-concentrator Files

the **table-map-custom.xml** and **index-concentrator-custom.xml** files should be edited to add custom meta keys and indexing the same. After deploying the parser from LIVE, log into NetWitness Platform to perform the following actions:

1. In the **NetWitness** menu, select ⚒ (Admin) > **Services**.
2. In the **Services** grid, select a Log Decoder, and from the **Actions** (⚙) menu, choose **View** > **Config**.
3. Open the **table-map-custom.xml**.

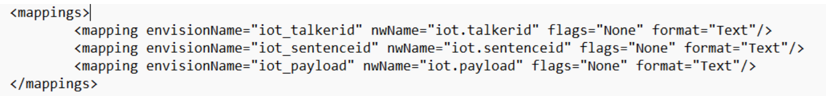


```
<?xml version="1.0" encoding="utf-8"?>
<!--
# attributes:
#     envisionName:    The name of the column in the universal table
#     nwName:               The name of the NetWitness meta field
#     format:          Optional. The language key data type. See LanguageManager. Defaults to "Text".
#     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>

    <!-- This is an example entry to use as a reference.  Everything must be inside the toplevel element "mappings". -->
    <!-- <mapping envisionName="bytes" nwName="bytes" flags="None" format="UInt64" nullTokens="(null)|-"/> -->

</mappings>
```

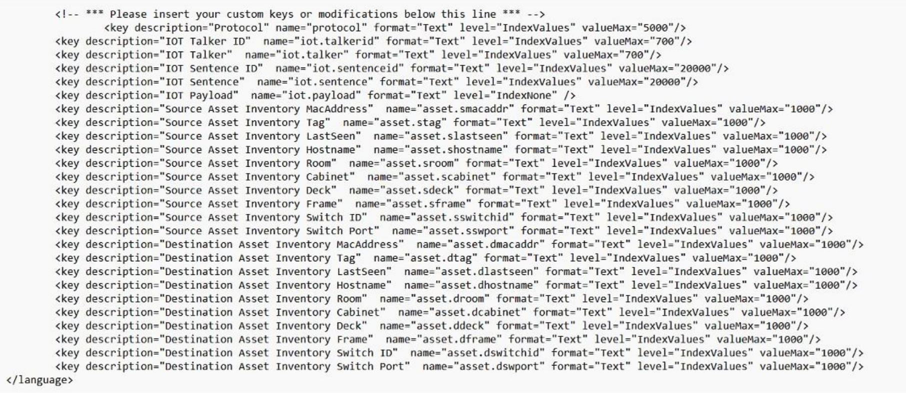
Add the required custom meta keys inside the <mappings> tag like below:



```
<mappings>
        <mapping envisionName="iot_talkerid" nwName="iot.talkerid" flags="None" format="Text"/>
        <mapping envisionName="iot_sentenceid" nwName="iot.sentenceid" flags="None" format="Text"/>
        <mapping envisionName="iot_payload" nwName="iot.payload" flags="None" format="Text"/>
</mappings>
```

4. Open the **index-concentrator-custom.xml**.

Add the required custom keys to be indexed after the line "**<!-- *** Please insert your custom keys or modifications below this line *** -->**" in the xml.



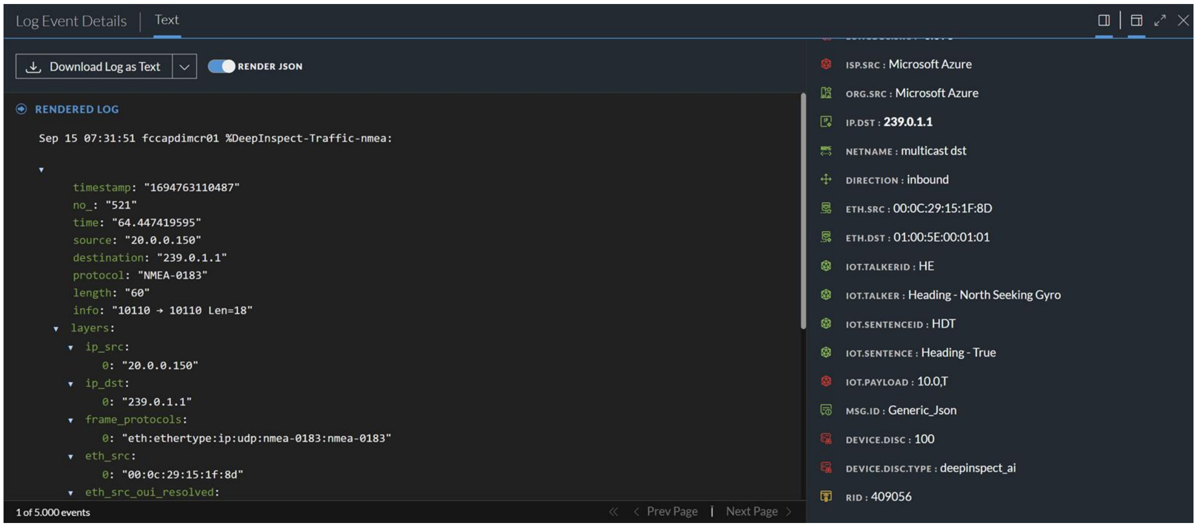
```
        <!-- *** Please insert your custom keys or modifications below this line *** -->
                <key description="Protocol" name="protocol" format="Text" level="IndexValues" valueMax="5000"/>
        <key description="IOT Talker ID"  name="iot.talkerid" format="Text" level="IndexValues" valueMax="700"/>
        <key description="IOT Talker"  name="iot.talker" format="Text" level="IndexValues" valueMax="700"/>
        <key description="IOT Sentence ID"  name="iot.sentenceid" format="Text" level="IndexValues" valueMax="20000"/>
        <key description="IOT Sentence"  name="iot.sentence" format="Text" level="IndexValues" valueMax="20000"/>
        <key description="IOT Payload"  name="iot.payload" format="Text" level="IndexNone" />
        <key description="Source Asset Inventory MacAddress"  name="asset.smacaddr" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Source Asset Inventory Tag"  name="asset.stag" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Source Asset Inventory LastSeen"  name="asset.slastseen" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Source Asset Inventory Hostname"  name="asset.shostname" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Source Asset Inventory Room"  name="asset.sroom" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Source Asset Inventory Cabinet"  name="asset.scabinet" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Source Asset Inventory Deck"  name="asset.sdeck" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Source Asset Inventory Frame"  name="asset.sframe" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Source Asset Inventory Switch ID"  name="asset.sswitchid" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Source Asset Inventory Switch Port"  name="asset.sswport" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Destination Asset Inventory MacAddress"  name="asset.dmacaddr" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Destination Asset Inventory Tag"  name="asset.dtag" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Destination Asset Inventory LastSeen"  name="asset.dlastseen" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Destination Asset Inventory Hostname"  name="asset.dhostname" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Destination Asset Inventory Room"  name="asset.droom" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Destination Asset Inventory Cabinet"  name="asset.dcabinet" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Destination Asset Inventory Deck"  name="asset.ddeck" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Destination Asset Inventory Frame"  name="asset.dframe" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Destination Asset Inventory Switch ID"  name="asset.dswitchid" format="Text" level="IndexValues" valueMax="1000"/>
        <key description="Destination Asset Inventory Switch Port"  name="asset.dswport" format="Text" level="IndexValues" valueMax="1000"/>
</language>
```

5. Navigate to (Admin) > **Services** and select the **Log Decoder(s)** then from the **Actions** ( ) menu, choose **Restart**.

6. Navigate to (Admin) > **Services** and select the **Concentrator(s)** then from the **Actions** ( ) menu, choose **Restart**.

7. Navigate to (Admin) > **Services** and select the **Log Decoder(s)** then from the **Actions** ( ) menu, choose **View** > **Config**.

8. The Log Decoder is now ready to parse events for this device. Below is an example of the NetWitness metadata collected.



```
Sep 15 07:31:51 fccapdimcr01 %DeepInspect-Traffic-nmea:

▾
        timestamp: "1694763110487"
        no_: "521"
        time: "64.447419595"
        source: "20.0.0.150"
        destination: "239.0.1.1"
        protocol: "NMEA-0183"
        length: "60"
        info: "10110 → 10110 Len=18"
    ▾ layers:
      ▾ ip_src:
          0: "20.0.0.150"
      ▾ ip_dst:
          0: "239.0.1.1"
      ▾ frame_protocols:
          0: "eth:ethertype:ip:udp:nmea-0183:nmea-0183"
      ▾ eth_src:
          0: "00:0c:29:15:1f:8d"
      ▾ eth_src_oui_resolved:
```

ISP.SRC : Microsoft Azure
ORG.SRC : Microsoft Azure
IP.DST : **239.0.1.1**
NETNAME : multicast dst
DIRECTION : inbound
ETH.SRC : 00:0C:29:15:1F:8D
ETH.DST : 01:00:5E:00:01:01
IOT.TALKERID : HE
IOT.TALKER : Heading - North Seeking Gyro
IOT.SENTENCEID : HDT
IOT.SENTENCE : Heading - True
IOT.PAYLOAD : 10.0,T
MSG.ID : Generic_Json
DEVICE.DISC : 100
DEVICE.DISC.TYPE : deepinspect_ai
RID : 409056

Appendix

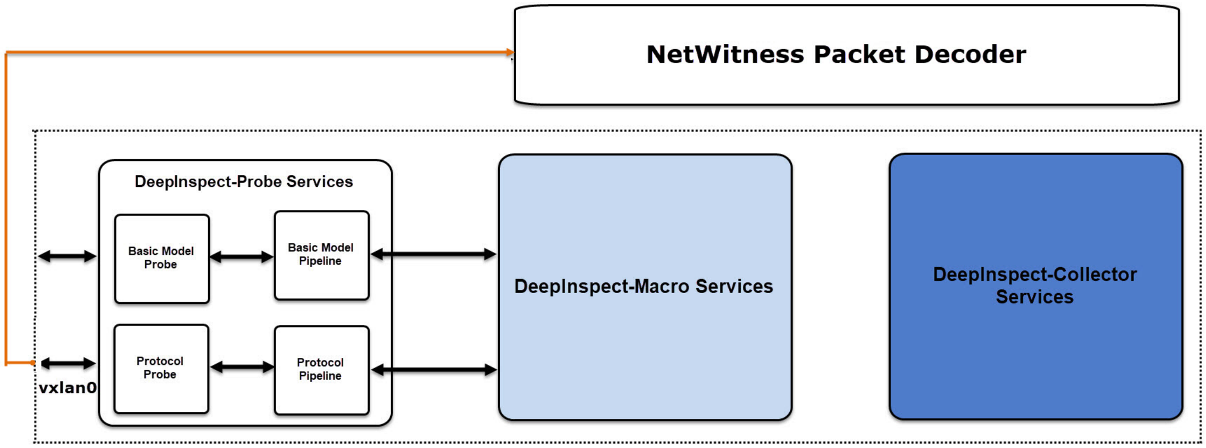
DeepInspect has the ability to create dynamic asset inventory from network traffic collected by probe services and forward IT/OT protocols Raw Traffic. The below sections provide necessary instructions.

IMPORTANT: Please note that this integration is developed in collaboration with DeepInspect. NetWitness recommends you to contact DeepInspect to configure the IT/OT protocols Raw Traffic forwarding or Asset Discovery and Anomaly Detection Feeds.

IT/OT Protocols Raw Traffic forwarding to NetWitness

Platform Packet Decoder

DeepInspect probe pipelines have the ability to forward IT/OT protocols Raw Traffic. DeepInpsect contains installation and configuration scripts to manage the configuration of a VxLAN tunnel for forwarding raw data to NetWitness Packet Decoder.


NetWitness Packet Decoder
DeepInspect-Probe Services
Basic Model Probe
Basic Model Pipeline
Protocol Probe
Protocol Pipeline
vxlan0
DeepInspect-Macro Services
DeepInspect-Collector Services

To manage the raw traffic forwarding, you will be requested to provide below parameter information:

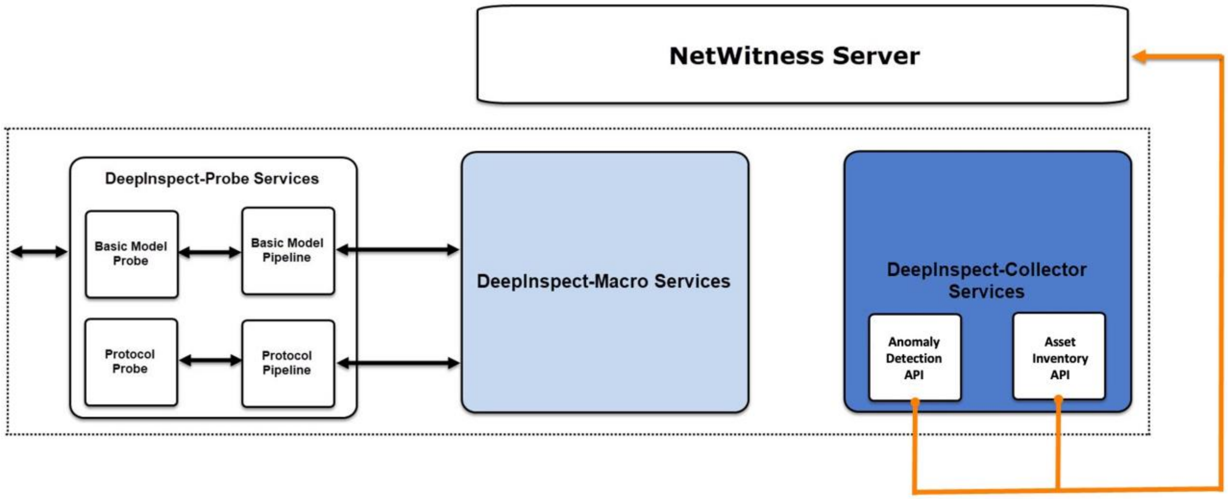
- **Probe Names** – List of DeepInspect probes to be forwarded to NetWitness Platform Log Collector.
- **NetWitness Packet Decoder VxLAN IP Address** – IP address of the VxLAN interface established on the NetWitness Packet Decoder.
- **NetWitness Packet Decoder APIs URL** – URL of NetWitness APIs to retrieve Network rules to filter raw traffic forwarded to NetWitness.
- **NetWitness Packet Decoder APIs Credentials** – Credentials of NetWitness APIs to retrieve Network rules to filter raw traffic forwarded to NetWitness.

To configure metadata forwarding, request DeepInspect support on ***support@deepinspect.it*** for the `DeepInspect_RSAXDR_Integration_Raw` playbook. After receiving the playbook, execute following command on collector:

```
ansible-playbook deepinspect_rsaxdr_integration_raw.yml -k -K -u root
```

Asset Discovery and Anomaly Detection Feeds to NetWitness Server

DeepInspect builds a dynamic asset inventory from network traffic collected by probe services. The solution also enriches asset inventory with anomaly detection.
DeepInpsect contains installation and configuration scripts to manage enrichment requests from the NetWitness Server.



To configure asset discovery and anomaly detection, request DeepInspect support on ***support@deepinspect.it*** for the `DeepInspect_RSAXDR_Integration_Enrichment` playbook. After receiving the playbook, execute following command on collector:

```
ansible-playbook deepinspect_rsaxdr_integration_enrichment.yml -k -K -u root
```

Getting Help with NetWitness Platform

Self-Help Resources

There are several options that provide you with help as you need it for installing and using NetWitness:

- See the documentation for all aspects of NetWitness here: https://community.netwitness.com/t5/netwitness-platform/ct-p/netwitness-documentation.
- Use the **Search** and **Create a Post** fields in NetWitness Community portal to find specific information here: https://community.netwitness.com/t5/netwitness-discussions/bd-p/netwitness-discussions.
- See the NetWitness Knowledge Base: https://community.netwitness.com/t5/netwitness-knowledge-base/tkb-p/netwitness-knowledge-base.
- See the documentation for Logstash JDBC input plugin here: https://www.elastic.co/guide/en/logstash/current/plugins-inputs-jdbc.html.
- See Troubleshooting section in the guides.
- See also NetWitness® Platform Blog Posts.
- If you need further assistance, Contact NetWitness Support.

Contact NetWitness Support

When you contact NetWitness Support, please provide the following information:

- The version number of the NetWitness Platform or application you are using.
- Logs information, even source version, and collection method.
- If you have problem with an event source, enable **Debug** parameter (set this parameter to **On** or **Verbose**) and collect the debug logs to share with the NetWitness Support team.

Use the following contact information if you have any questions or need assistance.

NetWitness Community Portal	https://community.netwitness.com In the main menu, click **Support** > **Case Portal** > **View My Cases**.
International Contacts (How to Contact NetWitness Support)	https://community.netwitness.com/t5/support/ct-p/support
Community	https://community.netwitness.com/t5/netwitness-discussions/bd-p/netwitness-discussions

Feedback on Product Documentation

You can send an email to feedbacknwdocs@netwitness.com to provide feedback on NetWitness Platform documentation.