NetWitness® Platform XDR

Ixia CloudLens Integration Guide

NETWITNESS

Platform XDR

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.

November, 2022

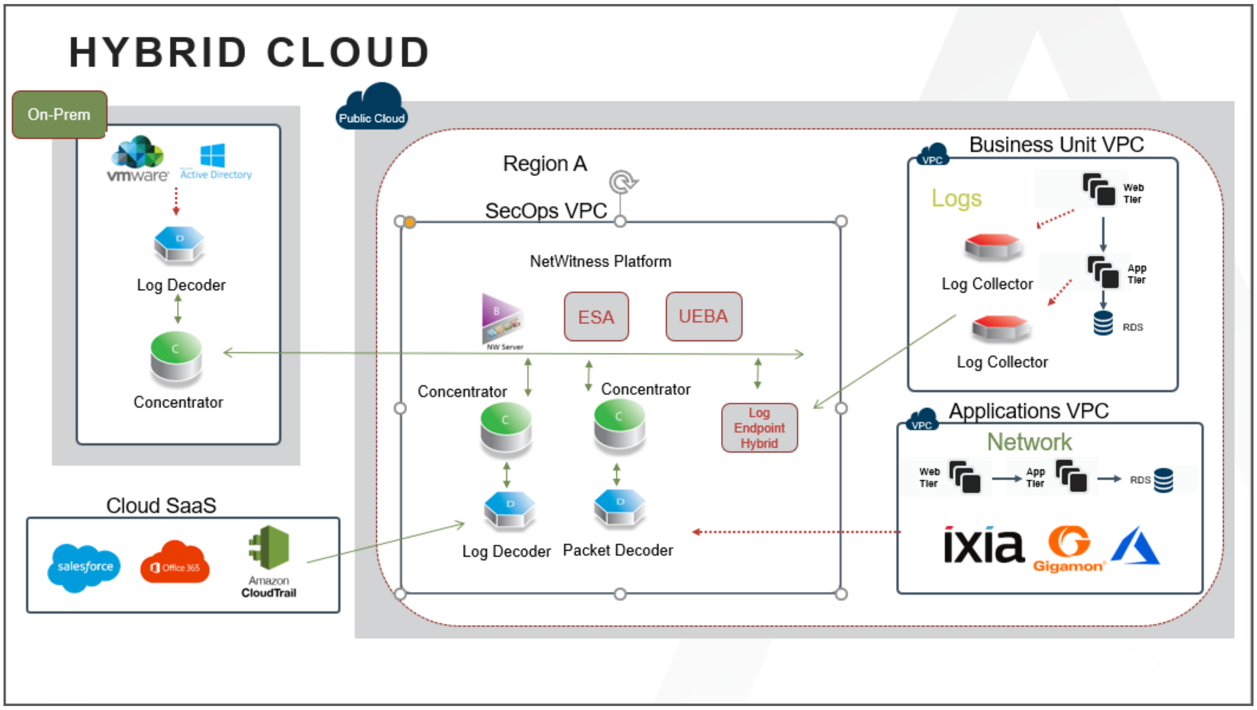
Contents

NetWitness Platform XDR Integration with Keysight Ixia CloudLens

NetWitness Logs and Packets combined with Keysight's network packet brokers and Taps provides pervasive visibility with advanced analytics– including real time behavior analytics - to detect and investigate sophisticated attacks.

The Keysight solution and NetWitness Platform XDR work together to capture and analyze network packet traffic in a scalable solution that can accurately and efficiently monitor networks of any size. Keysight network packet brokers passively direct out-of-band network packet data from multiple access points, such as SPANs, taps, and virtual taps (also sold by Keysight), in the network to NetWitness for capture. Traffic is aggregated from all needed access points in the network to provide comprehensive visibility.

NetWitness Logs and Packets unique architecture captures and enriches data sources with security context in real-time. Additionally, threat intelligence is applied to the enriched data to identify high risk indicators as APT domains, suspicious proxies or malicious networks. This method of processing large data sources in real-time provides analysts with security insight into their entire environment from on-premise to cloud.



Prerequisites

You need to setup the following before you begin the integration process:

For Azure Cloud:

1. Ixia CloudLens Manager instance should be up and running.
2. Authorize Network Security Groups of Cloudlens Manager Instance and client machines (including the decoder machine) to allow traffic from the following ports to the sensor tool.
   - TCP - 22 (SSH) : Connection to the instance / VM.
   - IP Protocol - 47 (GRE) : Required by CloudLens Sensor Tap to send the tapped traffic to the Sensor Tool.
   - UDP Protocol - 19993 (Encrypted Tunnel) : Required by CloudLens Sensor Tap to send the tapped traffic to the Sensor Tool.

For AWS Cloud:

1. Ixia CloudLens Manager instance should be up and running.
2. For Client Machines (as well as Decoder machine) and Cloudlens Manager Instance, the following ports must be opened on AWS Security Group Inbound Rules:
   - TCP - 22 (SSH) : Connection to the instance / VM.
   - IP Protocol - 47 (GRE) : Required by CloudLens Sensor Tap to send the tapped traffic to the Sensor Tool.
   - UDP Protocol - 19993 (Encrypted Tunnel) : Required by CloudLens Sensor Tap to send the tapped traffic to the Sensor Tool.

Integration Steps

Perform the following tasks to integrate the NetWitness Decoder with Ixia CloudLens.

1. Prepare Cloud Environment
2. Create CloudLens Project
3. Install Docker Container on Decoder
4. Install Docker Container on Clients
5. Create Mapping between Netwitness Decoder and Ixia Clients
6. Validate CloudLens Packets Arriving at Decoder
7. Set the Interface in the Packet Decoder

Prepare Cloud Environment

Prepare your cloud environment by performing the following procedures:

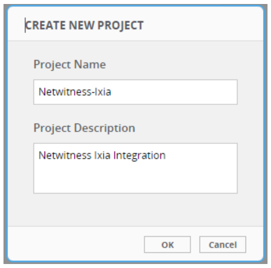
1. Deploy a NetWitness Decoder instance in your cloud environment. For more information, see Azure Installation Guide or AWS Installation Guide depending on your cloud environment.
2. Deploy client machines from which you want to route the traffic to NetWitness Decoder.

Create CloudLens Project

1. Login to **Ixia Cloudlens Manager** and go to the **Configure** Page.
2. Click + (add) to create a new project.
3. In the **CREATE NEW PROJECT** view,
   - Enter the Project Name

     For Example: **Netwitness-Ixia**.
   - Enter the Project Description

     For Example: **Netwitness Ixia Integration**.



CREATE NEW PROJECT

Project Name

Netwitness-Ixia

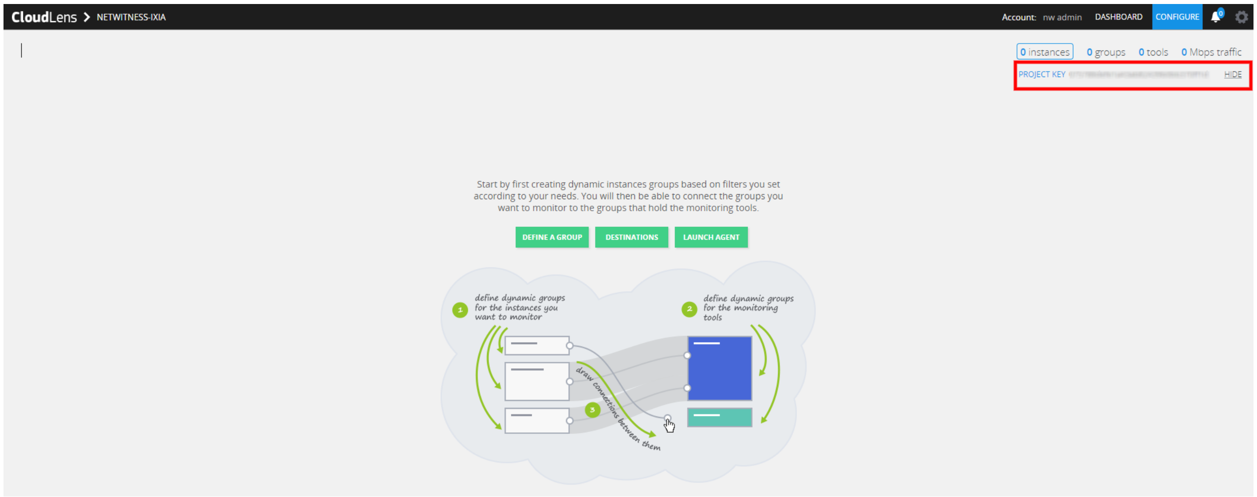
Project Description

Netwitness Ixia Integration

4. Click **OK**.

5. Click **SHOW PROJECT KEY** to get the API Key for the project.

   The key is required to configure the **Host and Tool agents**.



CloudLens > NETWITNESS-IXIA
Account: nw admin
DASHBOARD
CONFIGURE
0 instances
0 groups
0 tools
0 Mbps traffic
PROJECT KEY [redacted]
HIDE

Start by first creating dynamic instances groups based on filters you set according to your needs. You will then be able to connect the groups you want to monitor to the groups that hold the monitoring tools.

DEFINE A GROUP
DESTINATIONS
LAUNCH AGENT
1 define dynamic groups for the instances you want to monitor
2 define dynamic groups for the monitoring tools
3 draw connections between them

Install Docker Container on Decoder

1. SSH to Network Decoder.

2. Setup the docker. For more information on how to setup the docker, see https://docs.docker.com/engine/install/centos/.

3. Run the following commands to setup Docker insecure-registry parameter and pull the sensor image from CloudLens:

```
echo "{\"insecure-registries\":[\"<CloudLens_IP_here>\"]}" | sudo tee
/etc/docker/daemon.json
```

```
sudo systemctl enable docker.service
```

```
sudo service docker restart
```

4. Pull the CloudLens agent docker image. Run the following command:

```
sudo docker pull <CloudLens_IP_here>/sensor
```

5. Start the CloudLens agent with **ProjectKeyFromIxiaProjectPortal** retrieved from Create CloudLens Project and CloudLens Manager IP. Run the following command:

```
sudo docker run -v /lib/modules:/lib/modules -v /var/log:/var/log/cloudlens
-v /:/host -v /var/run/docker.sock:/var/run/docker.sock --cap-add SYS_
MODULE --cap-add SYS_RESOURCE --cap-add NET_RAW --cap-add NET_ADMIN --name
cloudlens-agent -d --restart=on-failure --net=host --log-opt max-size=50m -
-log-opt max-file=3 <CloudLens_IP_here>/sensor --accept_eula yes --project_
key ProjectKeyFromIxiaProjectPortal --server <CloudLens_IP_here> --ssl_
verify no
```

Install Docker Container on Clients

1. SSH to client VM with root privileges.
2. Setup the docker for the OS / Distributions. For more information, see https://docs.docker.com/engine/install/.
3. Run the following commands to setup Docker insecure-registry parameter and pull the sensor image from CloudLens:

   ```
   echo "{\"insecure-registries\":[\"<CloudLens_IP_here>\"]}" | sudo tee
   /etc/docker/daemon.json
   ```

   ```
   sudo systemctl enable docker.service
   ```

   ```
   sudo service docker restart
   ```

4. Pull the CloudLens agent docker image. Run the following command.

   ```
   sudo docker pull <CloudLens_IP_here>/sensor
   ```

5. Start the CloudLens agent with **ProjectKeyFromIxiaProjectPortal** retrieved from Create CloudLens Project and CloudLens Manager IP. Run the following command.

   ```
   sudo docker run -v /lib/modules:/lib/modules -v /var/log:/var/log/cloudlens
   -v /:/host -v /var/run/docker.sock:/var/run/docker.sock --cap-add SYS_
   MODULE --cap-add SYS_RESOURCE --cap-add NET_RAW --cap-add NET_ADMIN --name
   cloudlens-agent -d --restart=on-failure --net=host --log-opt max-size=50m -
   -log-opt max-file=3 <CloudLens_IP_here>/sensor --accept_eula yes --project_
   key ProjectKeyFromIxiaProjectPortal --server <CloudLens_IP_here> --ssl_
   verify no
   ```

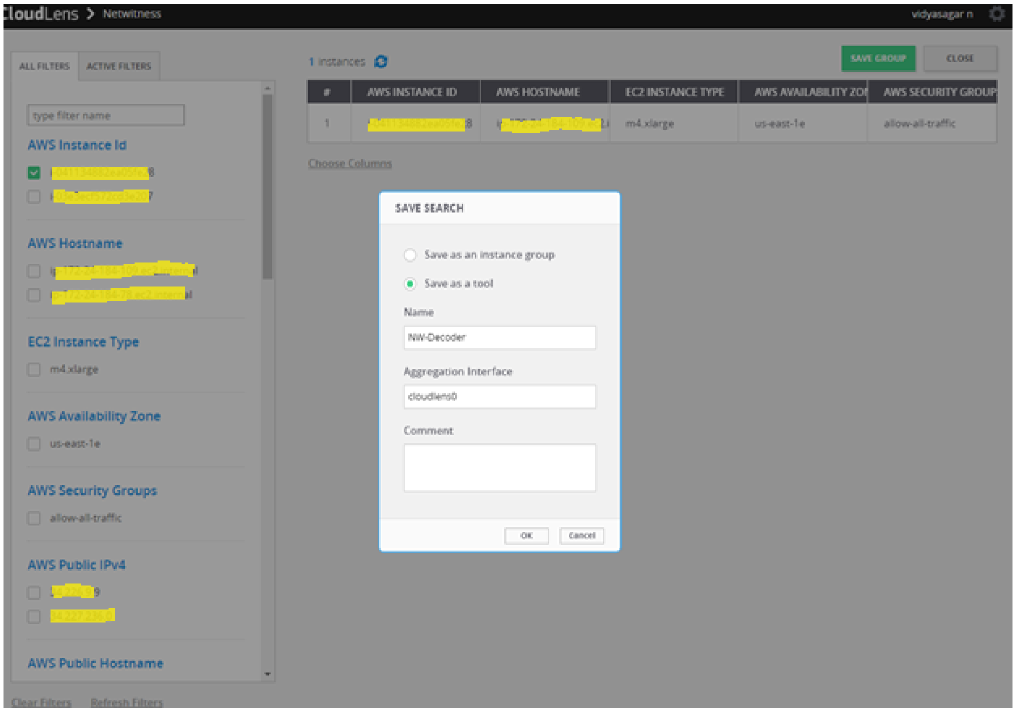
Create Mapping between Netwitness Decoder and Ixia Clients

Map the Network Decoder to the client machines to route the traffic to the Network Decoder. Do the following:

1. Go to the **CloudLens Manager** UI.
2. Click on your project and open it.
3. Click **Define Group** or the Instances count.

   You should see two instances listed, one for your decoder and the other for the client machines.
4. Apply filter for the decoder instance and click **Save Search**.
5. Select **Save as a tool**.
6. Specify a name for the tool and the **Aggregation Interface**.

   > **Note:** Use a meaningful name for the Aggregation Interface (for example **cloudlens0**. This is a virtual interface that appears in the OS where your Tool is installed. You need to instruct your tool to ‘listen’ to that interface in a subsequent step.



#	AWS INSTANCE ID	AWS HOSTNAME	EC2 INSTANCE TYPE	AWS AVAILABILITY ZO	AWS SECURITY GROUP
1	[redacted]	[redacted]	m4.xlarge	us-east-1e	allow-all-traffic

type filter name

AWS Instance Id
☑ [redacted]
☐ [redacted]

AWS Hostname
☐ [redacted]
☐ [redacted]

EC2 Instance Type
☐ m4.xlarge

AWS Availability Zone
☐ us-east-1e

AWS Security Groups
☐ allow-all-traffic

AWS Public IPv4
☐ [redacted]
☐ [redacted]

AWS Public Hostname

SAVE SEARCH

○ Save as an instance group
● Save as a tool

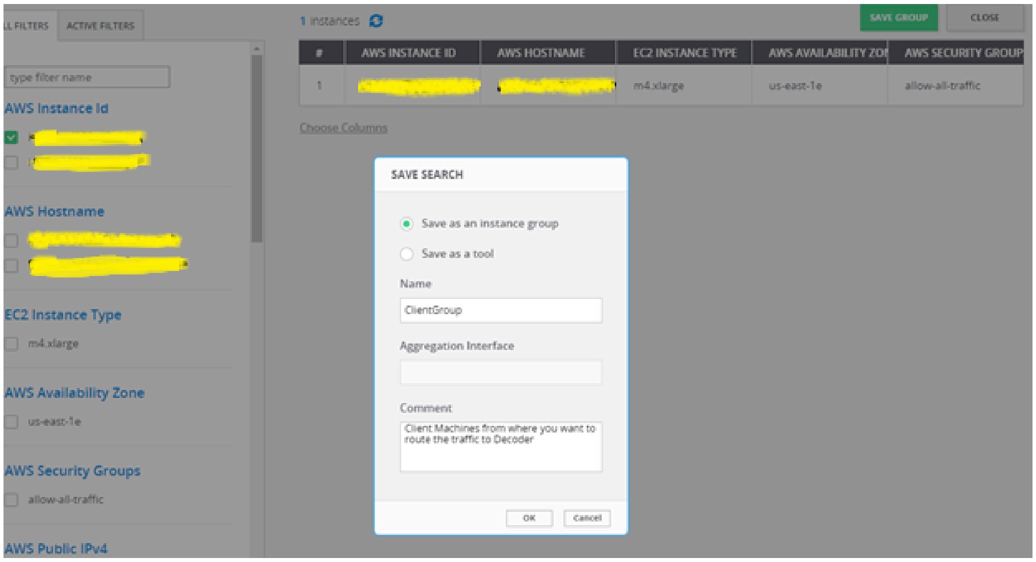
Name: NW-Decoder

Aggregation Interface: cloudlens0

Comment:

OK Cancel

7. Apply filter for the client host instance from the list and click **Save Search**.



#	AWS INSTANCE ID	AWS HOSTNAME	EC2 INSTANCE TYPE	AWS AVAILABILITY ZO	AWS SECURITY GROUP
1	[redacted]	[redacted]	m4.xlarge	us-east-1e	allow-all-traffic

type filter name

AWS Instance Id
☑ [redacted]
☐ [redacted]

AWS Hostname
☐ [redacted]
☐ [redacted]

EC2 Instance Type
☐ m4.xlarge

AWS Availability Zone
☐ us-east-1e

AWS Security Groups
☐ allow-all-traffic

AWS Public IPv4

SAVE SEARCH

● Save as an instance group
○ Save as a tool

Name: ClientGroup

Aggregation Interface:

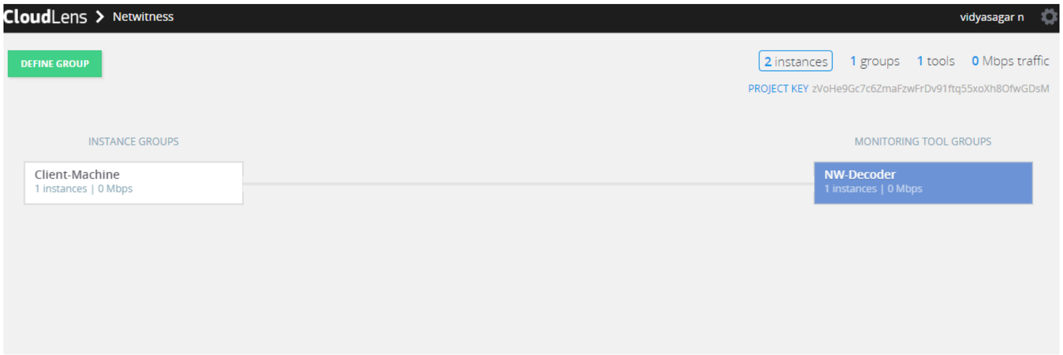
Comment: Client Machines from where you want to route the traffic to Decoder

OK Cancel

8. Navigate back to the top-level view of the project.

   Your client machine instance and Decoder instance are now displayed.

9. Drag a connection between the client machine instance and Decoder instance to allow the flow of packets.



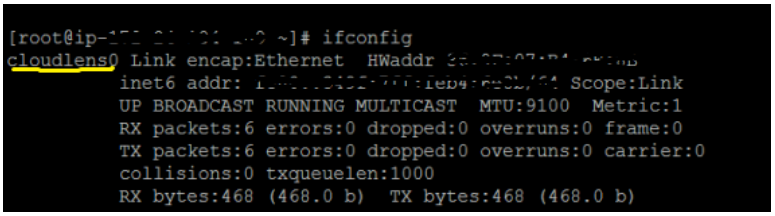
Validate CloudLens Packets Arriving at Decoder

Complete the following steps to validate that the packets are actually arriving at the Network Decoder.

1. SSH to the Network Decoder.
2. Run the following command.

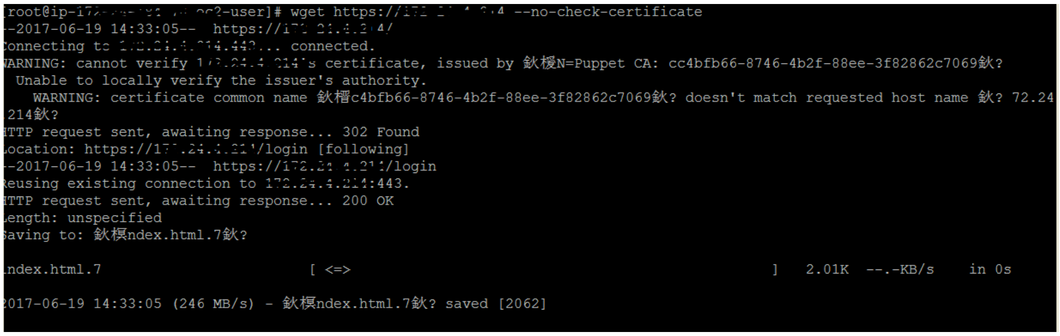
   `ifconfig`

   The new aggregation interface you created is displayed.



```
[root@ip-[redacted] ~]# ifconfig
cloudlens0 Link encap:Ethernet  HWaddr [redacted]
          inet6 addr: [redacted] Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:9100  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:468 (468.0 b)  TX bytes:468 (468.0 b)
```

3. Generate traffic from the client machine CLI (for example: `wget http://www.google.com/`).



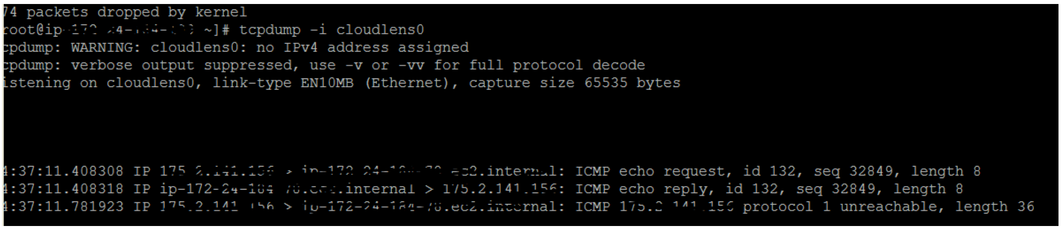
```
[root@ip-[redacted] ec2-user]# wget https://[redacted] --no-check-certificate
--2017-06-19 14:33:05--  https://[redacted]/
Connecting to [redacted]... connected.
WARNING: cannot verify [redacted]'s certificate, issued by 鈥楥N=Puppet CA: cc4bfb66-8746-4b2f-88ee-3f82862c7069鈥?
  Unable to locally verify the issuer's authority.
    WARNING: certificate common name 鈥榗4bfb66-8746-4b2f-88ee-3f82862c7069鈥? doesn't match requested host name 鈥? 72.24
.214鈥?
HTTP request sent, awaiting response... 302 Found
Location: https://[redacted]/login [following]
--2017-06-19 14:33:05--  https://[redacted]/login
Reusing existing connection to [redacted]:443.
HTTP request sent, awaiting response... 200 OK
Length: unspecified
Saving to: 鈥榠ndex.html.7鈥?

index.html.7                  [ <=>                                           ]   2.01K  --.-KB/s    in 0s

2017-06-19 14:33:05 (246 MB/s) - 鈥榠ndex.html.7鈥? saved [2062]
```

4. SSH to the Network Decoder and go to your Network Decoder instance CLI.
5. Run the following command to look for suitable results in the tcpdump.

```
tcpdump -I Cloudlens0
```



```
/4 packets dropped by kernel
root@ip-[redacted] ~]# tcpdump -i cloudlens0
cpdump: WARNING: cloudlens0: no IPv4 address assigned
cpdump: verbose output suppressed, use -v or -vv for full protocol decode
istening on cloudlens0, link-type EN10MB (Ethernet), capture size 65535 bytes



4:37:11.408308 IP 175[redacted] > ip-172[redacted].internal: ICMP echo request, id 132, seq 32849, length 8
4:37:11.408318 IP ip-172-24-[redacted].internal > 175.2.141.156: ICMP echo reply, id 132, seq 32849, length 8
4:37:11.781923 IP 175.2.141[redacted] > ip-172-24-[redacted].ec2.internal: ICMP 175.2[redacted] protocol 1 unreachable, length 36
```

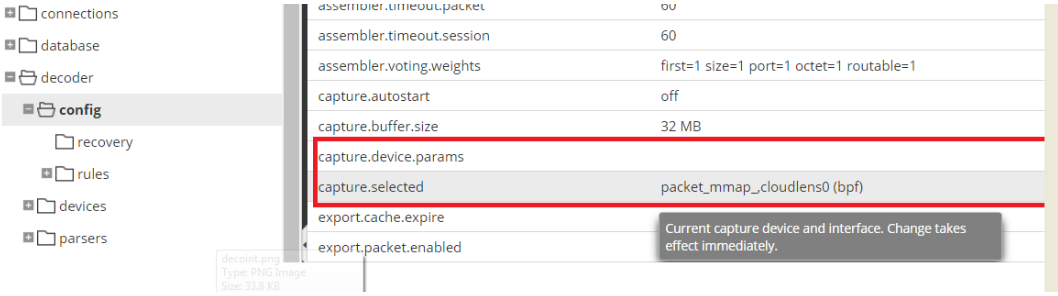
Set the Interface in the Packet Decoder

Complete the following steps in the Network Decoder to set the interface for the Ixia integration.

1. SSH to the Network Decoder.
2. Run the following command to restart the decoder service:

   ```
   $ sudo restart nwdecoder
   ```

   The Network Decoder is now set to capture the network traffic.
3. Log in to NetWitness and click **(Admin) > Services**.
4. Select a Decoder service and click > **View > Explore**.
5. Expand the **decoder** node and click **config** to view the configuration settings.
6. Set the **capture.selected** parameter to the following value.
   packet_mmap_,cloudlens0(bpf)



assembler.timeout.packet	60
assembler.timeout.session	60
assembler.voting.weights	first=1 size=1 port=1 octet=1 routable=1
capture.autostart	off
capture.buffer.size	32 MB
capture.device.params	
capture.selected	packet_mmap_,cloudlens0 (bpf)
export.cache.expire	
export.packet.enabled	

Current capture device and interface. Change takes effect immediately.

7. Restart the Decoder service after you set the **capture.selected** parameter.

Getting Help with NetWitness Platform XDR

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 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 XDR 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 nwdocsfeedback@netwitness.com to provide feedback on NetWitness Platform documentation.