Traffic Monitoring and Packet Analysis

**SPL-TF-100-SITMPA-1 - Version 1.0.4**

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

Security engineers can use Amazon VPC Traffic Mirroring to send a complete copy of traffic to a target of their choice. This is a very important capability to allow for in-depth traffic monitoring, analysis, and threat detection. In this lab, you learn how to use VPC Traffic Mirroring to capture the traffic of your interest for monitoring purposes.

OBJECTIVES

By the end of this lab, you should be able to do the following:

* Identify the elastic network interfaces (ENI) to be used for Traffic Mirroring.
* Configure a traffic mirror target.
* Configure a traffic mirror filter to select traffic of interest.
* Create a traffic mirror session.
* Verify that selected traffic is being sent to the mirror target.
* Modify the traffic mirror filter to capture different traffic.
* Send the captured packets to a file for detailed analysis.

TECHNICAL KNOWLEDGE PREREQUISITES

This lab requires that you have a basic knowledge of:

* AWS services as defined in the AWS Cloud Practitioner Essentials course.
* Networking concepts such as IP Addressing and CIDR notation.
* Navigating through the AWS Management Console.
* Running commands in a Linux command line interface (CLI).

DURATION

This lab requires approximately *60* minutes to complete.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Learn more:** Where to find more information.
* **Note:** A hint, tip, or important guidance.
* **Command:** A command that you must run.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Hint:** A hint to a question or challenge.
* **Task complete:** A conclusion or summary point in the lab.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

**Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**WARNING:** **Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

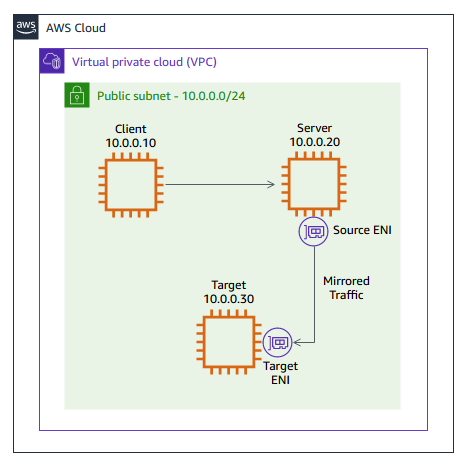
**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

LAB ENVIRONMENT

The following diagram shows the basic architecture of the lab environment:



*Image description: The following list details the major resources in the lab:*

* *An Amazon Virtual Private Cloud (Amazon VPC) with one public subnet.*
* *An Amazon Elastic Compute Cloud (Amazon EC2) instance acting as a client sending the traffic to mirror. This instance has an IP address of 10.0.0.10.*
* *An EC2 instance acting as a server receiving the traffic to mirror. This instance has an IP address of 10.0.0.20.*
* *An EC2 instance acting as a target to send mirrored traffic to. This instance has an IP address of 10.0.0.30.*
* *Each instance has a security group that allows the required traffic for that instance based on the lab scenario.*

**VPC Traffic Mirroring concepts**

The following are the key concepts for Traffic Mirroring:

* Source — The network interface to monitor.
* Target — The destination for mirrored traffic.
* Filter — A set of rules that defines the traffic that is copied in a traffic mirror session.
* Session — An entity that describes Traffic Mirroring from a source to a target using filters.

**Learn more:** Refer to *What is Traffic Mirroring?* in the **Additional resources** section for more information.

Consider the lab scenario where you have client-server sessions running and you need to capture selected inbound/outbound traffic on the *Server* instance. This is how the Traffic Mirroring concepts are applied in your scenario:

* Source: This is the network interface of the *Server* instance.
* Target: This is network interface of the *Target* instance.
* Filter: You can configure multiple filter rules to define which inbound/outbound traffic you want to capture. In the lab, you want to capture ICMP and HTTP traffic.
* Session: This is the session that describes a source, target, and filter to capture the traffic.

AWS SERVICES NOT USED IN THIS LAB

AWS service capabilities used in this lab are limited to what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

**Task 1: Identify the network interfaces to be used for Traffic Mirroring**

In this task, you view and record the network interfaces IDs for two EC2 instances that serve as the source and target for Traffic Mirroring.

1. At the top of the AWS Management Console, in the search bar, search for and choose

EC2

.

1. In the navigation pane at the left of the page, under **Instances**, choose **Instances**.
2. Choose the link for the Instance ID that is named **Server**. This opens the instance summary page for the *Server* EC2 instance.
3. Scroll down in the **Instance summary** page. Locate and choose the **Networking** tab.
4. Scroll down and locate the **Network Interfaces** section. Copy the **Interface ID** and record it in your preferred text editor as the *Source ENI*. You need this ID in future tasks. This interface acts as the *source* of the mirroring traffic.
5. Repeat the previous steps to locate, view, and record the **Interface ID** for the EC2 Instance named *Target*. Record the ID as the *Target ENI*. This interface acts as the *target* of the mirroring traffic.

**Task complete:** You have successfully viewed and recorded the network interfaces IDs for two EC2 instances that serve as the source and target for Traffic Mirroring.

**Task 2: Configure a traffic mirror target**

In this task, you configure a traffic mirror target that receives the captured traffic.

**Note:** For this task, you need to reference the Interface ID recorded for the *Target ENI*. This is where copies of the mirrored traffic are sent.

1. At the top of the AWS Management Console, in the search bar, search for and choose

VPC

.

1. In the navigation pane at the left of the page, under **Traffic Mirroring**, choose **Mirror targets**.
2. Choose **Create traffic mirror target**.
3. In the **Create traffic mirror target** page, in the **Target settings** section:

* For **Name tag - optional**, enter

MyTarget

* For **Description - optional**, enter

This is the target that receives the mirrored traffic

1. In the **Choose target** section:

* For **Target type**, select **Network Interface**
* For **Target**, select the Interface ID that matches the *Target ENI* that you recorded earlier.

1. Choose **Create**

You should now have a traffic mirror target successfully created and listed on the **Traffic mirror targets** page.

**Task complete:** You have successfully configured a traffic mirror target that receives the captured traffic.

**Task 3: Configure a traffic mirror filter**

In this task, you configure a traffic mirror filter to select which traffic is copied and mirrored to the traffic mirror target.

In a traffic mirror filter, you can configure *Inbound* and/or *outbound* rules to define which traffic (protocol, port, source/destination ports, and source/destination IP addresses) you want capture or ignore in the traffic mirror session.

In this filter, you configure a rule to capture any ICMP traffic inbound to the *Server* instance which has an IP of 10.0.0.20 from the subnet IP range of 10.0.0.0/24.

1. In the navigation pane at the left of the page, under **Traffic Mirroring**, choose **Mirror filters**.
2. Choose **Create traffic mirror filter**.
3. In the **Create traffic mirror filter** page, in the **Filter settings** section:

* For **Name tag - optional**, enter

MyFilter

* For **Description - optional**, enter

Inbound traffic from subnet to server

1. In the **Inbound rules - optional** section, choose **Add rule** then, configure the following:

* For **Number**, enter

100

* For **Rule action**, select **accept**
* For **Protocol**, select **ICMP(1)**
* For **Source CIDR block**, enter

10.0.0.0/24

* For **Destination CIDR block**, enter

10.0.0.20/32

* For **Description**, enter

Inbound ICMP from subnet to server

1. Choose **Create**.
2. After your traffic mirror filter and rule are created, choose **Close**.

**Task complete:** You have successfully configured a traffic mirror filter with a rule to capture selected traffic.

**Task 4: Configure a traffic mirror session**

In this task, you create a traffic mirror session. Traffic mirror sessions send mirrored packets from the source to a target so that you can monitor and analyze traffic.

**Note:** For this task, you need to reference the Interface ID recorded for the *Source ENI*. This is where traffic is copied from for the traffic mirror session.

1. In the navigation pane at the left of the page, under **Traffic Mirroring**, choose **Mirror sessions**.
2. Choose **Create traffic mirror session**.
3. In the **Create traffic mirror page** page, in the **Session settings** section:

* For **Name tag - optional**, enter

ServerTraffic

* For **Description - optional**, enter

Mirroring server traffic

* For **Mirror source**, select the ID that matches the *Source ENI* that you recorded earlier
* For **Mirror target**, select **MyTarget**

1. In the **Additional settings** section, configure the following:

* For **Session number**, enter

1

* For **VNI - optional**, enter

100

* For **Filter**, select **MyFilter**

1. Choose **Create**.

**Task complete:** You have successfully created a traffic mirror session that sends copies of mirrored packets from the *server* instance to the *target* instance. The session only captures traffic that matches the mirror filter.

**Task 5: Capture and verify mirrored traffic at the target host**

In this task, you capture and verify the mirrored traffic at the *Target* instance.

You need to connect to the terminal of both *Client* and *Target* instances. You use the *Client* instance terminal to initiate the traffic from the client to the server. While you use the *Target* instance terminal to capture and inspect the traffic. You use AWS System Manager Session Manager to connect to the instances.

1. To connect to the **Client** instance terminal, from the list to the left of these instructions, copy the value of **ClientSSMSessionURL** and open it in a new browser tab. This opens a terminal session to the instance with the **Client$** shell prompt.
2. To connect to the **Target** instance terminal, from the list to the left of these instructions, copy the value of **TargetSSMSessionURL** and open it in a new browser tab. This opens a terminal session to the instance with the **Target$** shell prompt.

**Hint:** For the remainder of the lab, you use the terminal to run AWS CLI commands on both *Client* and *Target* instances. Make sure that you keep the browser tabs for each terminal session open. Also, before you run a command, double-check that you are using the correct terminal (whether for the *Client* or *Target* instance).

For this lab, you use tcpdump utility on the **Target** host to capture traffic. Tcpdump is a command line utility that allows you to capture and analyze network traffic moving through a system.

**Learn more:** Refer to *TCPDUMP & LiBCAP* in the **Additional resources** section for more information.

You want to start a persistent traffic capture on the **Target** instance for traffic sent over UDP port 4789. This is the port that VPC Traffic Mirroring uses to encapsulate the captured packets.

**Note:** The security group of the **Target** instance must allow inbound traffic on UDP port 4789. Otherwise, the tcpdump cannot capture the traffic sent by the traffic mirror session.

1. **Command:** To start a persistent traffic capture, run the following command on the **Target** instance:

sudo tcpdump -nni eth0 -vvv udp dst port 4789

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

Target$

Your **Target** instance is ready to listen to captured traffic. Now, you need to start generating traffic that matches the mirror filter you configured earlier.

1. **Command:** To initiate ICMP traffic, run the following command on the **Client** instance (This command sends a single ICMP packet from the **Client** to the **Server**:

ping 10.0.0.20 -c 1

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PING 10.0.0.20 (10.0.0.20) 56(84) bytes of data.

64 bytes from 10.0.0.20: icmp\_seq=1 ttl=255 time=1.80 ms

--- 10.0.0.20 ping statistics ---

1 packets transmitted, 1 received, 0% packet loss, time 0ms

rtt min/avg/max/mdev = 1.804/1.804/1.804/0.000 ms

Client$

1. Choose the web browser tab where you have the **Target** instance prompt open. In this tab, the open tcpdump session should display the packet that was captured.

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

22:55:33.523992 IP (tos 0x0, ttl 255, id 0, offset 0, flags [none], proto UDP (17), length 134)

10.0.0.20.65436 > 10.0.0.30.4789: [no cksum] VXLAN, flags [I] (0x08), vni 100

IP (tos 0x0, ttl 255, id 31712, offset 0, flags [DF], proto ICMP (1), length 84)

10.0.0.10 > 10.0.0.20: ICMP echo request, id 27541, seq 1, length 64

The first two lines of the output has information on the captured traffic. It shows that the captured traffic was sent from the mirror source of 10.0.0.20 (Server) to the mirror target of 10.0.0.30 (Target) using port 4789. It also shows the VNI ID of 100 which you configured in the mirror session.

The second two lines shows the ICMP packet that was sent from 10.0.0.10 (Client) to 10.0.0.20 (Server) using ICMP echo request.

**Task complete:** You have successfully started a capture session, initiated traffic, and verified capturing the traffic on the **Target** instance.

**Task 6: Modify the traffic mirror filter to capture different traffic**

In this task, you modify the traffic mirror filter to capture HTTP traffic from the **Client** to the **Server**.

1. **Command:** To initiate HTTP traffic from **Client** to **Server** instances, run the following command on the **Client** instance:

curl http://10.0.0.20

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Test Website - Server EC2 Instance

Client$

1. Choose the web browser tab where you have the **Target** instance prompt open. You cannot see any packets captured as the filter you created is only capturing ICMP traffic to the server.

Now modify the filter to allow HTTP capture.

1. At the top of the AWS Management Console, in the search bar, search for and choose

VPC

.

1. In the navigation pane at the left of the page, under **Traffic mirroring**, choose **Mirror filters**.
2. Choose the link under the **Filter ID** column.
3. In the **MyFilter** details page, choose the **Add inbound rule**.
4. In the **Add traffic mirror filter rule**, configure the following:

* For **Number**, enter

200

* For **Description**, enter

Inbound HTTP from subnet to server

* For **Rule action**, select **accept**
* For **Protocol**, select **TCP(6)**
* For **Destination port range - optional**, enter

80

* For **Source CIDR block**, enter

10.0.0.0/24

* For **Destination CIDR block**, enter

10.0.0.20/32

1. Choose **Add rule**.

Your filter is now updated to have a second rule for HTTP.

1. Choose the web browser tab where you have the **Client** instance prompt open to initiate HTTP traffic again.
2. **Command:** To initiate HTTP traffic from **Client** to **Server** instances, run the following command on the **Client** instance:

curl http://10.0.0.20

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Test Website - Server EC2 Instance

Client$

1. Choose the web browser tab where you have the **Target** instance prompt open. You should be able to see the HTTP traffic captured.
2. While on the **Target** instance prompt, press CTRL+C to stop the current tcpdump session.

**Task complete:** You have successfully modified the traffic mirror filter by adding an additional rule to capture HTTP traffic.

**Task 7: Send the captured traffic to a file**

In this task, you capture traffic with tcpdump while sending the captured packets to a file rather than viewing them directly on the screen. This allows you to examine the file later and perform deep analysis using different tools.

The packets are saved into a file that has “pcap” format. This is a standard format used by network monitoring tools and sniffers.

1. Choose the web browser tab where you have the **Target** instance prompt open.
2. **Command:** To start a new tcpdump session that sends captured packets to a file, run the following command on the **Target** instance:

sudo tcpdump -nni eth0 -vvv udp dst port 4789 -w capture.pcap

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes

Got 0

1. Choose the web browser tab where you have the **Client** instance prompt open to initiate HTTP traffic again.
2. **Command:** To initiate ICMP traffic from **Client** to **Server** instances, run the following command on the **Client** instance:

ping 10.0.0.20 -c 1

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PING 10.0.0.20 (10.0.0.20) 56(84) bytes of data.

64 bytes from 10.0.0.20: icmp\_seq=1 ttl=255 time=0.250 ms

--- 10.0.0.20 ping statistics ---

1 packets transmitted, 1 received, 0% packet loss, time 0ms

rtt min/avg/max/mdev = 0.250/0.250/0.250/0.000 ms

Client$

1. Choose the web browser tab where you have the **Target** instance prompt open. Notice that the prompts displays “Got 1” to indicate that it captured one packet.
2. Press CTRL+C to stop the current tcpdump session.
3. **Command:** To view the captured file contents, run the following command on the **Target** instance:

tcpdump -r capture.pcap

**Expected output:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* This is OUTPUT ONLY. \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

reading from file cap1.pcap, link-type EN10MB (Ethernet)

00:29:39.752434 IP ip-10-0-0-20.us-west-2.compute.internal.65431 > ip-10-0-0-30.us-west-2.compute.internal.4789: VXLAN, flags [I] (0x08), vni 100

IP ip-10-0-0-10.us-west-2.compute.internal > ip-10-0-0-20.us-west-2.compute.internal: ICMP echo request, id 8695, seq 1, length 64

Target$

You can use a tool like Wireshark to perform a deep packet analysis on the pcap file.

**Task complete:** You have successfully captured traffic with tcpdump and sent the captured packets to a file.

**Conclusion**

You have successfully done the following:

* Identified the elastic network interfaces (ENI) to be used for Traffic Mirroring.
* Configured a traffic mirror target.
* Configured a traffic mirror filter to select traffic of interest.
* Configured a traffic mirror session.
* Verified that selected traffic is being sent to the traffic mirror target.
* Modified the traffic mirror filter to capture different traffic.
* Sent the captured packets to a file for detailed analysis.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

* [What is Traffic Mirroring?](https://docs.aws.amazon.com/vpc/latest/mirroring/what-is-traffic-mirroring.html).
* [TCPDUMP & LiBCAP](https://www.tcpdump.org/).

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).