Dom Moore
Traffic Mirroring

Overview

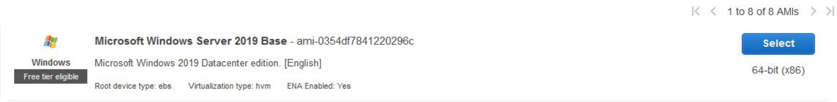
Apply network packet analysis concepts to a cloud environment.

Part 1: Staging the Security Onion AMI

Step 1- Deploy an AMI that uses an instance type built on the Nitro System.

Step 2- Deploy Security Onion

Deploy instance



1 to 8 of 8 AMIs

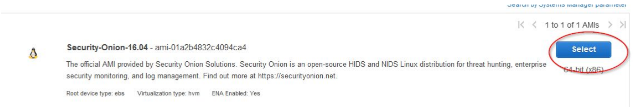
Microsoft Windows Server 2019 Base - ami-0354df7841220296c

Microsoft Windows 2019 Datacenter edition. [English]

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

64-bit (x86)

Deploy Security Onion



Search by Systems Manager parameter

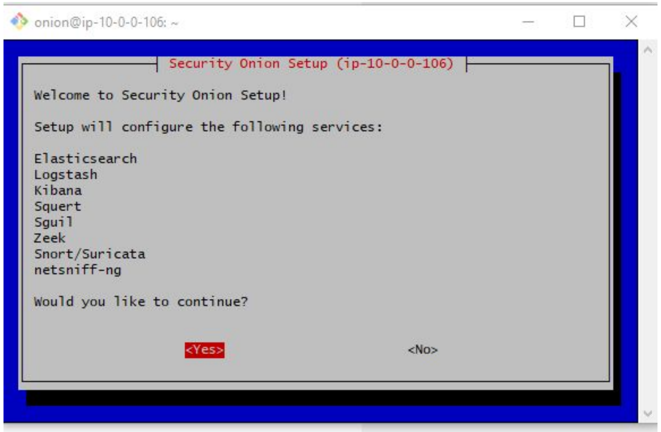
1 to 1 of 1 AMIs

Security-Onion-16.04 - ami-01a2b4832c4094ca4

The official AMI provided by Security Onion Solutions. Security Onion is an open-source HIDS and NIDS Linux distribution for threat hunting, enterprise security monitoring, and log management. Find out more at https://securityonion.net.

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

64-bit (x86)



onion@ip-10-0-0-106: ~

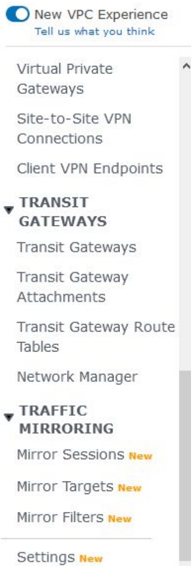
```
Security Onion Setup (ip-10-0-0-106)

Welcome to Security Onion Setup!

Setup will configure the following services:

Elasticsearch
Logstash
Kibana
Squert
Sguil
Zeek
Snort/Suricata
netsniff-ng

Would you like to continue?

                 <Yes>                    <No>
```

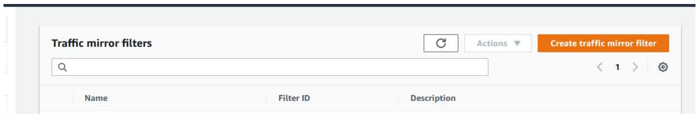
Part 2: Traffic Mirroring Implementation

- Create a mirror target.
- Create a mirror filter.
- Create a mirror session.

Target

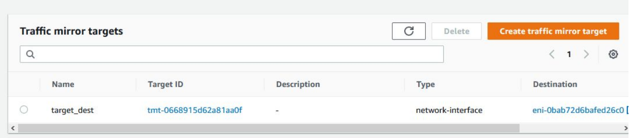
Filter





Traffic mirror filters

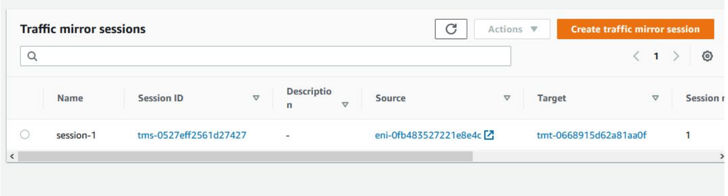
Name	Filter ID	Description



Traffic mirror targets

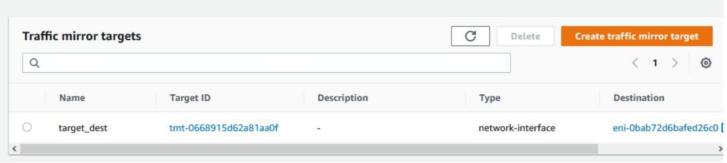
	Name	Target ID	Description	Type	Destination
○	target_dest	tmt-0668915d62a81aa0f	-	network-interface	eni-0bab72d6bafed26c0

session



Traffic mirror sessions

	Name	Session ID	Description	Source	Target	Session r
○	session-1	tms-0527eff2561d27427	-	eni-0fb483527221e8e4c	tmt-0668915d62a81aa0f	1



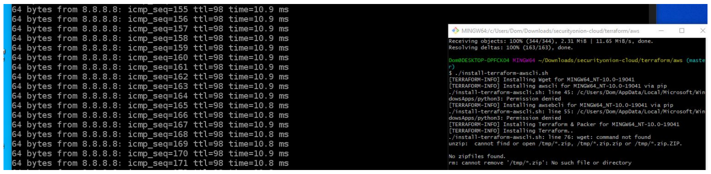
Traffic mirror targets

	Name	Target ID	Description	Type	Destination
○	target_dest	tmt-0668915d62a81aa0f	-	network-interface	eni-0bab72d6bafed26c0

Part 3: Verify Traffic Mirroring

- Send an ICMP packet to scanme.nmap.org and capture the packet. Take a screenshot of your ICMP packet and paste into your deliverable
- Capture and record an HTTP GET request to an HTTP-only website to capture the HTML text and tag contents of the page. Take a screenshot of your HTTP OK packet alongside your browser's inspection dump and paste it into your deliverable.
- Capture and record an FTP authentication request. Take a screenshot of your FTP traffic and paste it to your deliverable.

Ping

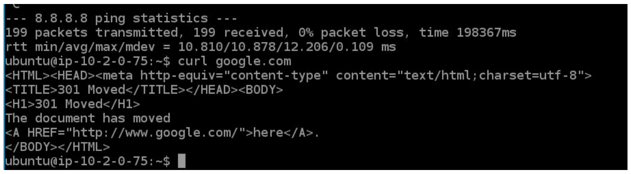


```
64 bytes from 8.8.8.8: icmp_seq=155 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=156 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=157 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=158 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=159 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=160 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=161 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=162 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=163 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=164 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=165 ttl=98 time=10.8 ms
64 bytes from 8.8.8.8: icmp_seq=166 ttl=98 time=10.8 ms
64 bytes from 8.8.8.8: icmp_seq=167 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=168 ttl=98 time=10.8 ms
64 bytes from 8.8.8.8: icmp_seq=169 ttl=98 time=10.8 ms
64 bytes from 8.8.8.8: icmp_seq=170 ttl=98 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=171 ttl=98 time=10.8 ms
```

MINGW64:/c/Users/Dom/Downloads/securityonion-cloud/terraform/aws

```
Receiving objects: 100% (344/344), 2.31 MiB | 11.65 MiB/s, done.
Resolving deltas: 100% (163/163), done.

Dom@DESKTOP-DPFCK04 MINGW64 ~/Downloads/securityonion-cloud/terraform/aws (maste
r)
$ ./install-terraform-awscli.sh
[TERRAFORM-INFO] Installing Wget for MINGW64_NT-10.0-19041
[TERRAFORM-INFO] Installing awscli for MINGW64_NT-10.0-19041 via pip
./install-terraform-awscli.sh: line 45: /c/Users/Dom/AppData/Local/Microsoft/Win
dowsApps/python3: Permission denied
[TERRAFORM-INFO] Installing awsebcli for MINGW64_NT-10.0-19041 via pip
./install-terraform-awscli.sh: line 55: /c/Users/Dom/AppData/Local/Microsoft/Win
dowsApps/python3: Permission denied
[TERRAFORM-INFO] Installing Terraform & Packer for MINGW64_NT-10.0-19041
[TERRAFORM-INFO] Installing Terraform..
./install-terraform-awscli.sh: line 76: wget: command not found
unzip:  cannot find or open /tmp/*.zip, /tmp/*.zip.zip or /tmp/*.zip.ZIP.

No zipfiles found.
rm: cannot remove '/tmp/*.zip': No such file or directory
```

Http



```
--- 8.8.8.8 ping statistics ---
199 packets transmitted, 199 received, 0% packet loss, time 198367ms
rtt min/avg/max/mdev = 10.810/10.878/12.206/0.109 ms
ubuntu@ip-10-2-0-75:~$ curl google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
ubuntu@ip-10-2-0-75:~$
```

Part 4: Reporting

- Why might an organization implement traffic mirroring in its AWS cloud?
  - To detect patterns on the network and look for abnormal instances of traffic
- Compare and contrast Wireshark and Security Onion.
  - The overall sanctions are similar as it pertains to setting up the rules and aws, but the security onion interfaces take some adjusting
- How does capturing network traffic in the cloud differ from on-prem LAN?
  - The overall fundamentals are the same you have to acknowledge that you're on a public platform with an encrypted vpc but you are still on the internet
- What feature used today should be reconfigured if you end up capturing too much-unwanted traffic from the source network interface?
  - You could filter the rules to be more specific, I did not accept ssh filters to avoid unwanted traffic
- What other tools are present on Security Onion that might help us observe and record what is happening in our cloud?
  - You can log your results and create files to be able to view traffic for better analysis.