Suricata & AWS

Pre and Post Session Mirroring

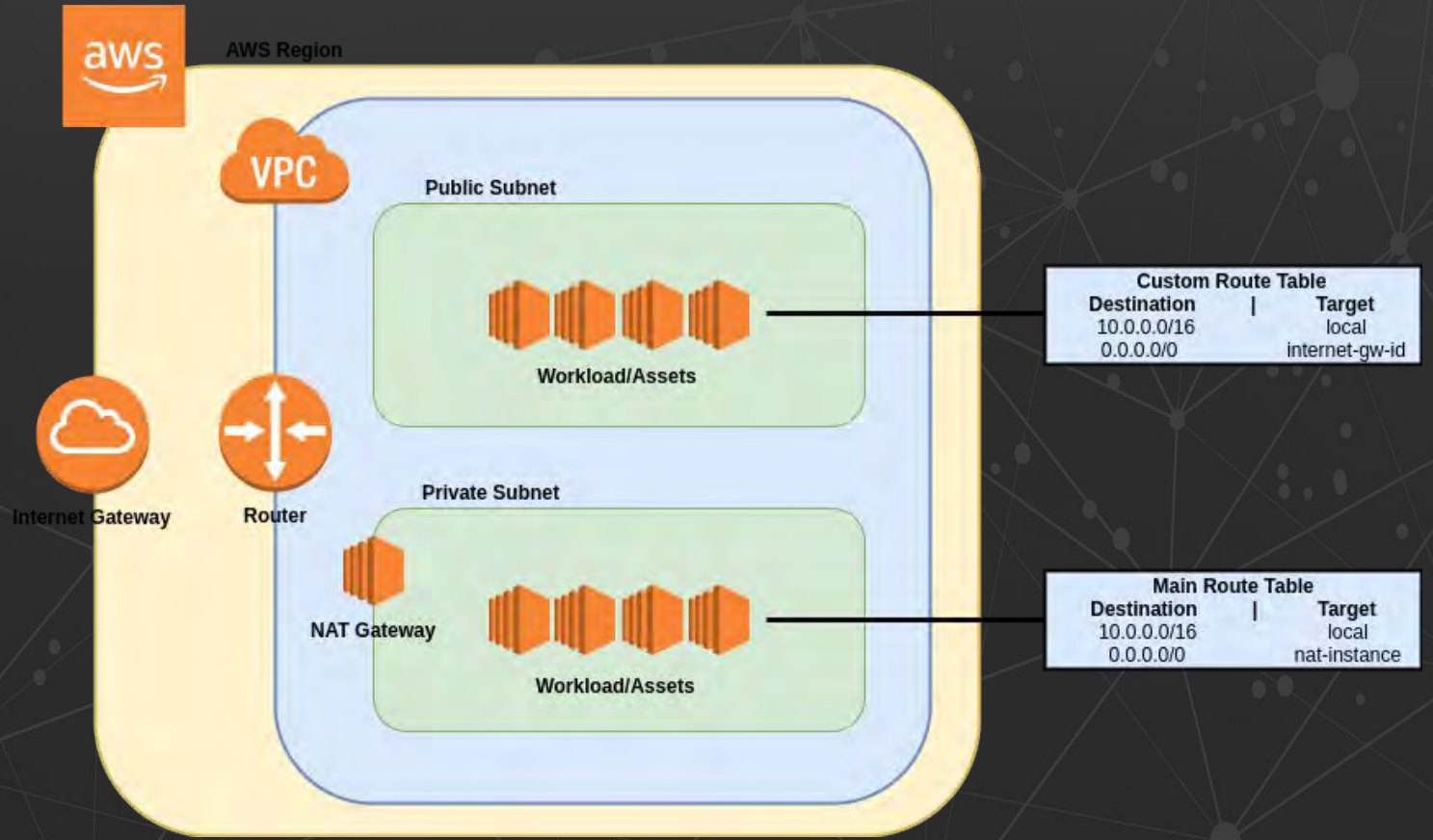
$ cat about.txt

- Overview of Suricata in AWS
- Some lessons learned
- Sharing is caring
- Community feedback

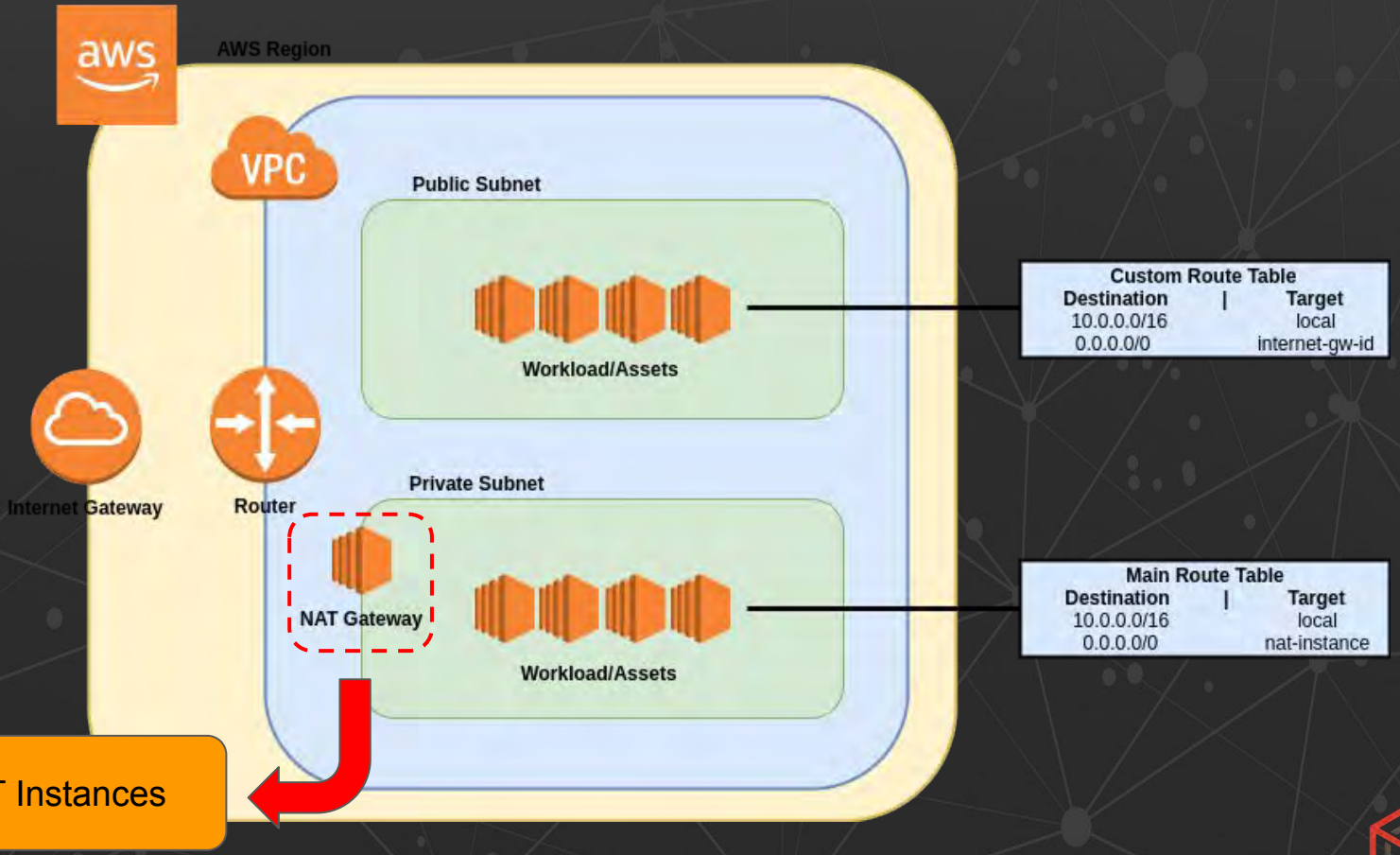
$ cat not-about.txt

- AWS course

$ cat aws-101.txt



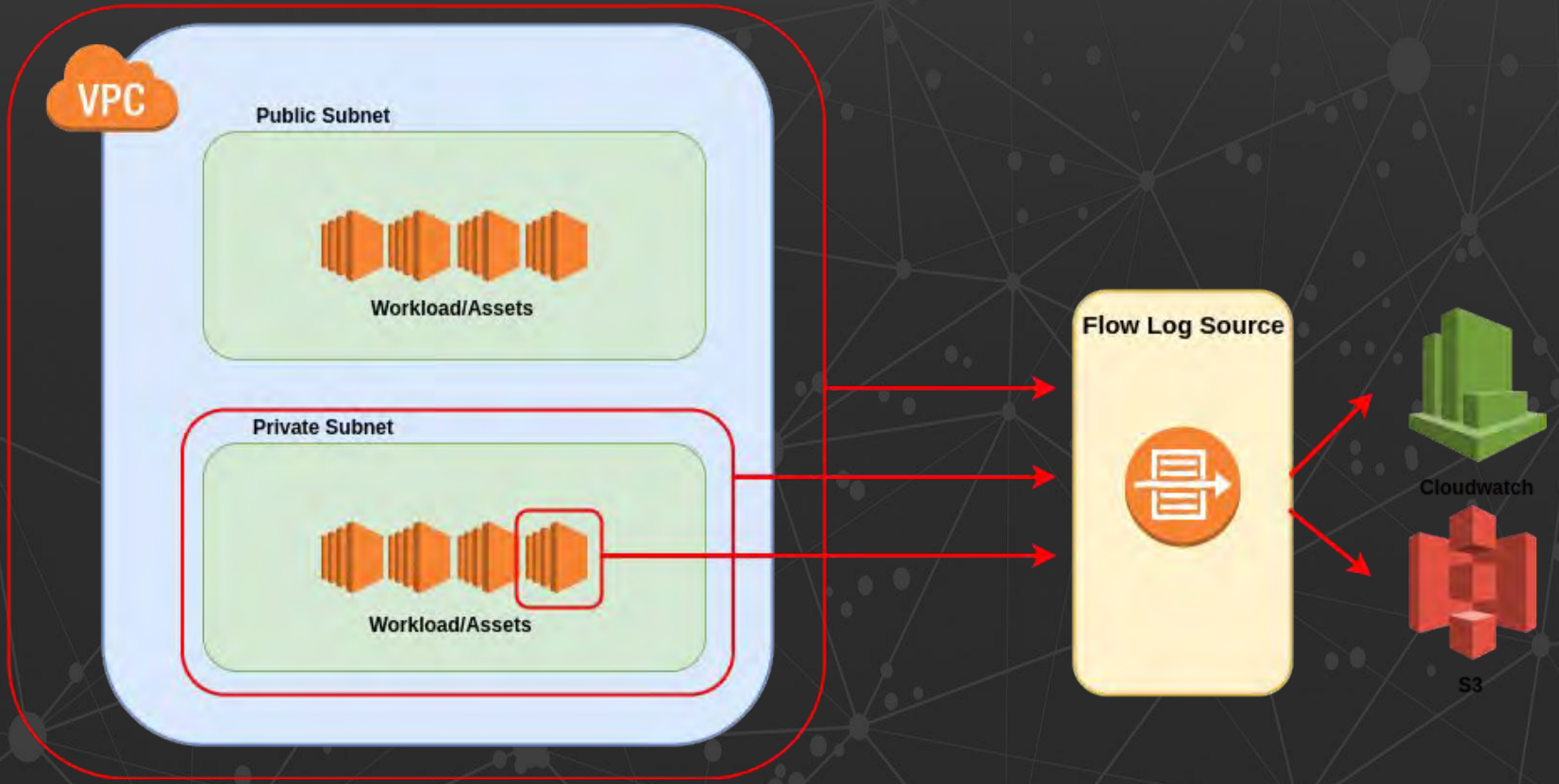
$ eog nsm-aws-pre.png



$ cat nsm-aws-pre.txt

- net.ipv4.ip_forward=1
- Hard to size correctly
- Multi-AZ Deployment *(still, single point of failure)*
- Cost *(instance type & multiple instances)*
- Limited visibility *(no lateral)*

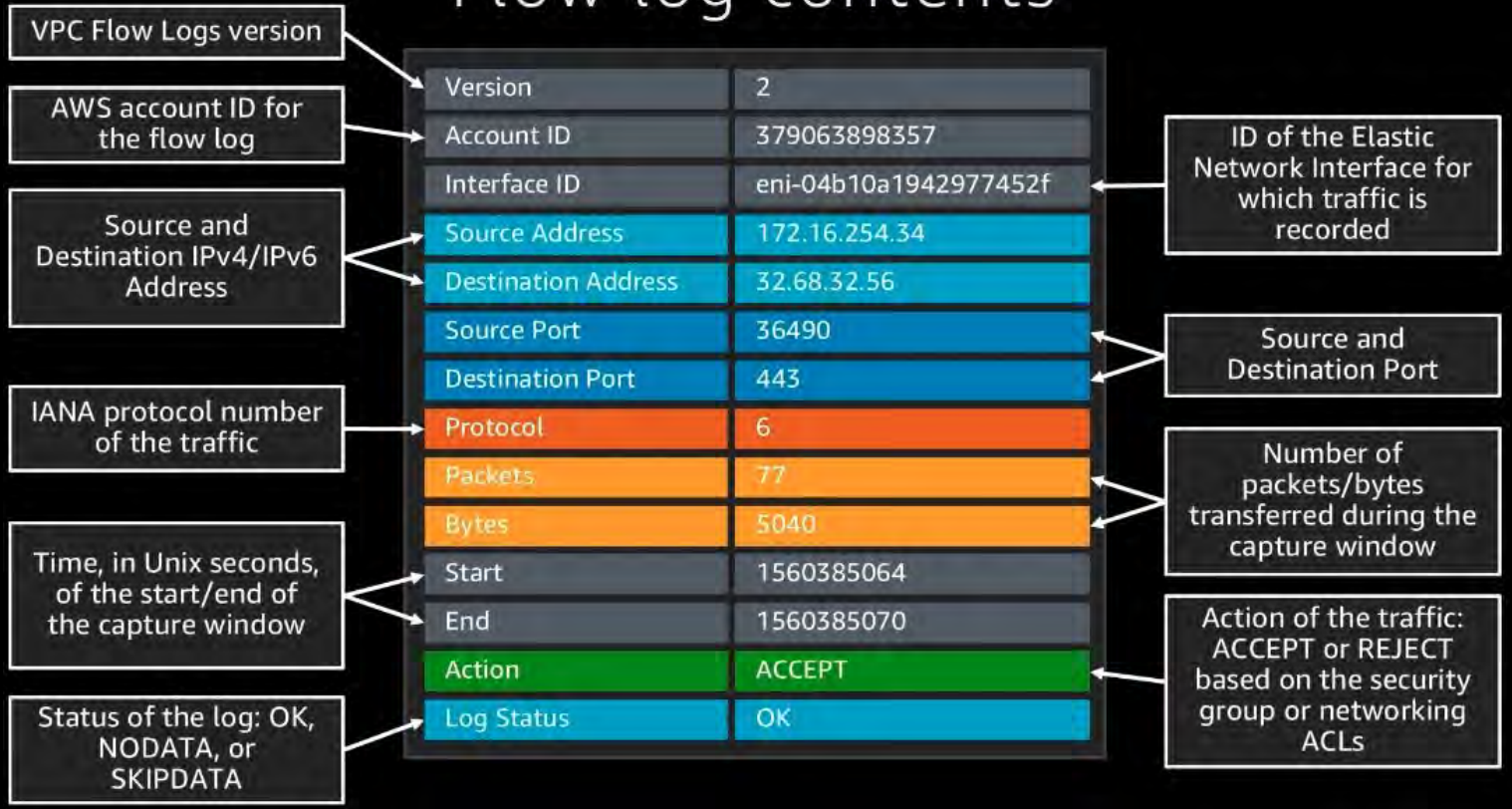
$ cat nsm-aws-pre-flowlogs.txt | more



Reference: AWS re:Inforce 2019: SEP209 (Youtube)

--MORE--

Flow log contents



SURICON

Source: AWS re:Inforce 2019: SEP209 (Youtube)

--MORE--

- Used as a building block
- Excellent tool for troubleshooting
- Security Groups & Network ACL’s

$ cat nsm-aws-pre-alternatives.txt

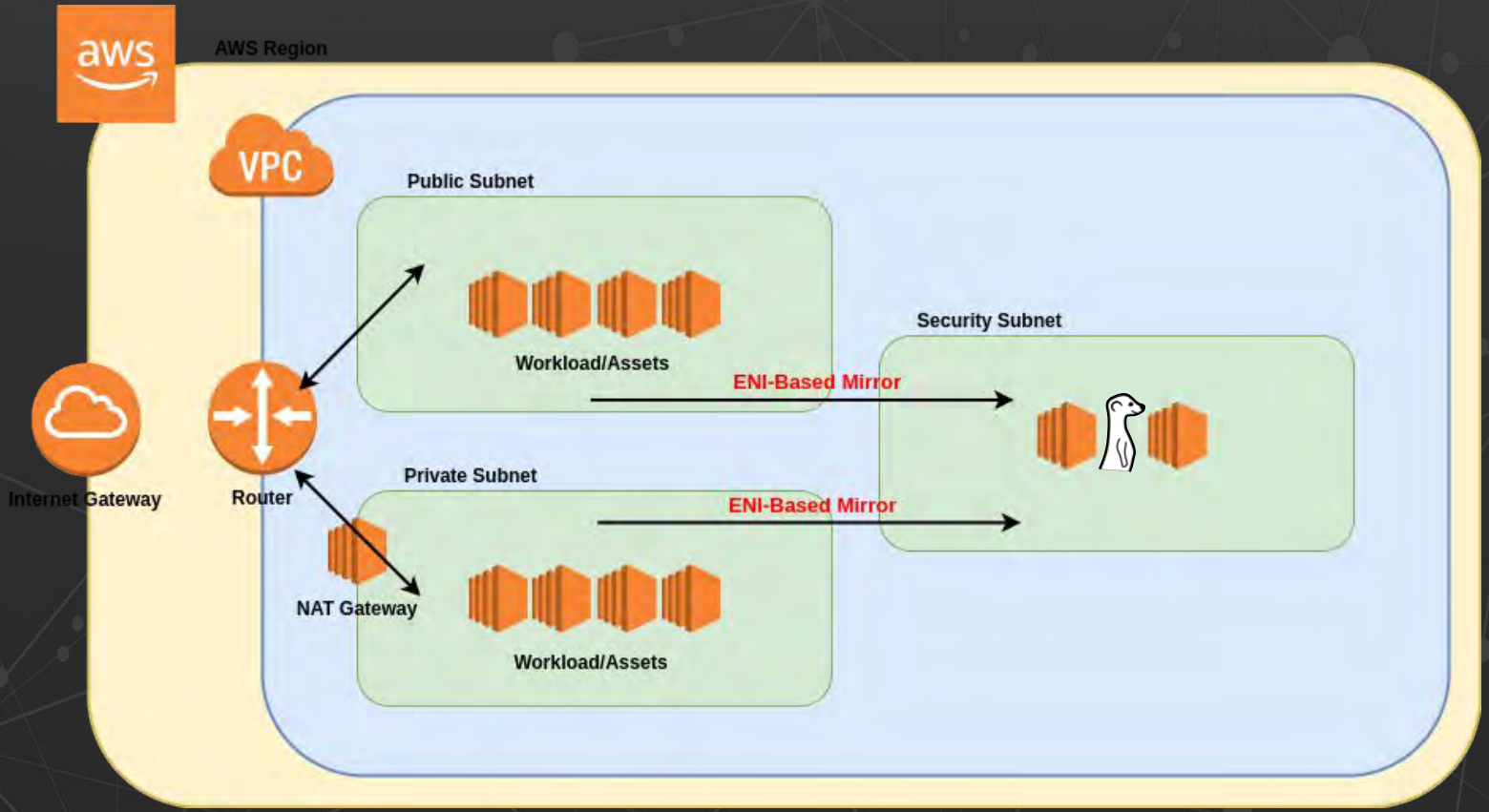
- Agents
- Traffic duplication at OS level
- Next-gen *<buzzword>* mirroring tech
- COST!

$ cat quote-nsm-amazon.txt

“Our number one tenet is to not cause harm or an availability impact; none of the cloud visibility solutions previously available allowed us to be non intrusive...until now.”

Dave Burke, Principal Security Engineer, Amazon.com

$ cat nsm-aws-mirror.txt | more



--MORE--

- No longer inline
- No more traffic duplication at OS
- No agents/maintenance
- Capture at the **E**lastic **N**etwork **I**nterface level
- LATERAL MOVEMENT!
- Cost
- Visibility into often missed log-centric tools
- *_insert_reason_why_we_love_NSM*

$ cat nsm-aws-anatomy-mirror.txt | more

TARGET

FILTER

SESSIONS

SURICON

Icons from **ultimatearm** & **Nikita Golubev** @ flaticon.com

--MORE--

- Elastic Network Interface
- Not everything with an ENI, though
- EC2 and Network Load Balancer
- No 1:1; Target can be used by several Sessions
- UDP 4789 (VXLAN) in SG

SURICON

--MORE--

FILTER

- Inbound or Outbound
- Protocol-based (TCP/UDP) filtering
- Source & Destination
- CIDRs supported
- Port (for both SRC and DEST)

SURICON

--MORE--

SESSIONS

- Up to 3 sessions per source *(ENI)*
- Lower session has priority (packets are mirrored only once)
  - #1 - HTTP -> Sensor01
  - #2 - HTTPS -> Sensor02
  - #3 - ALL -> Sensor03

$ cat nsm-aws-first-mirror.txt | more

- Launch your instance

c5n.large	2	5.25	EBS only	Yes	Up to 25 Gigabit
c5n.xlarge	4	10.5	EBS only	Yes	Up to 25 Gigabit
c5n.2xlarge	8	21	EBS only	Yes	Up to 25 Gigabit
c5n.4xlarge	16	42	EBS only	Yes	Up to 25 Gigabit
c5n.9xlarge	36	96	EBS only	Yes	50 Gigabit
c5n.18xlarge	72	192	EBS only	Yes	100 Gigabit

SURICON

Icons from **ultimatearm** & **Nikita Golubev** @ flaticon.com

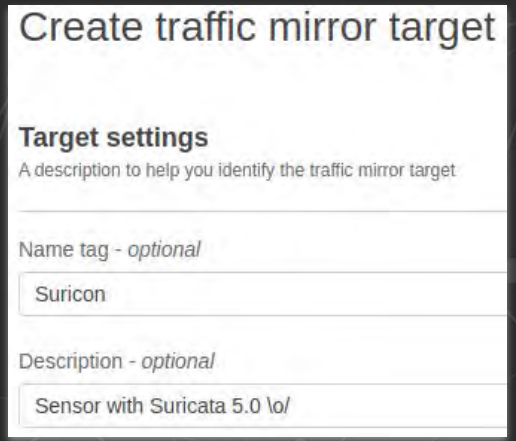
--MORE--

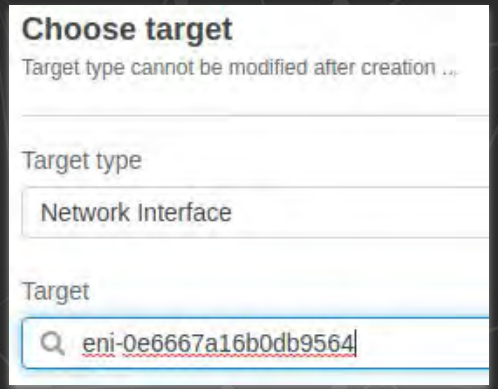
- Launch your instance
- Name your interfaces

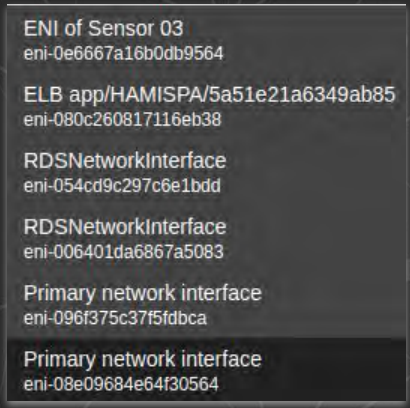
Icons from **ultimatearm** & **Nikita Golubev** @ flaticon.com

--MORE--

- Launch your instance
- Name your interfaces
- Create your target



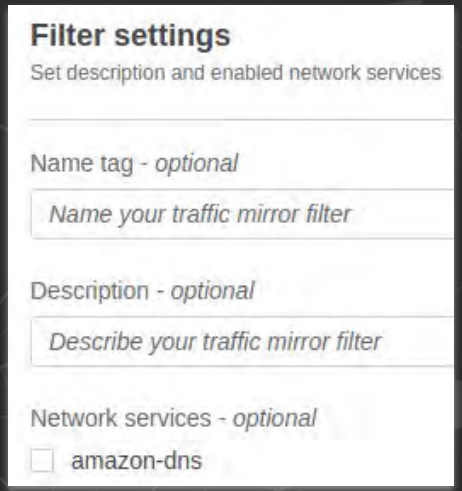


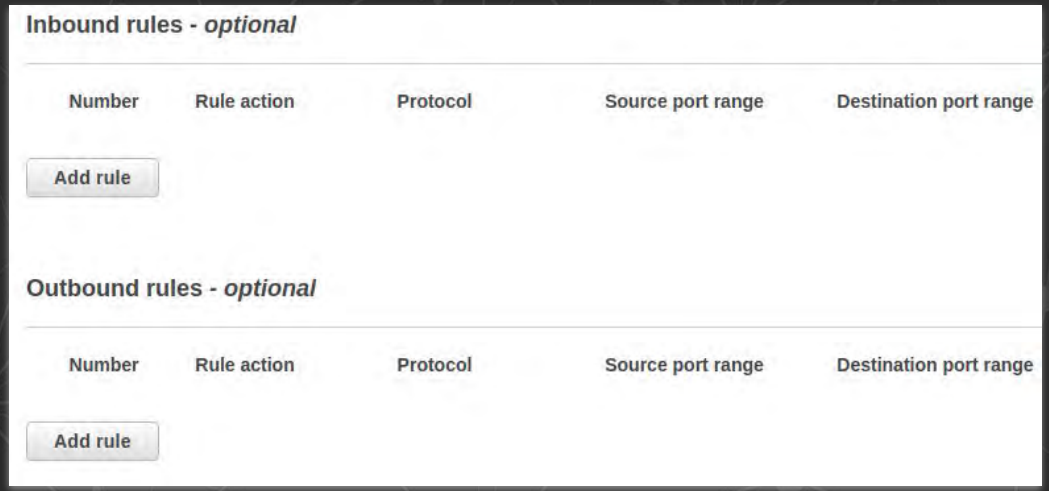


SURICON

--MORE--

- Launch your instance
- Name your interfaces
- Create your target
- Create your filters



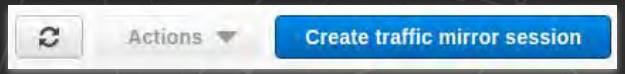


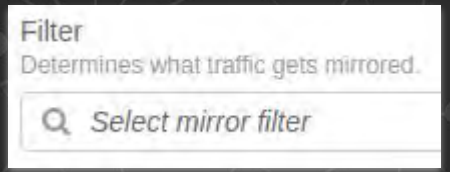
SURICON

Icons from **ultimatearm** & **Nikita Golubev** @ flaticon.com

--MORE--

- Launch your instance
- Name your interfaces
- Create your target
- Create your filters
- Create your session





SURICON

Icons from **ultimatearm** & **Nikita Golubev** @ flaticon.com

Export to PDF

https://youtu.be/jy8wH-YKiF0

$ cat pre-toolkit-intro.txt

Can we make it easier?

$ cat toolkit-intro.txt

Mirror Toolkit

A set of tools to ease the creation of traffic mirror sessions, increase automation and facilitate maintenance.

SURICON

$ cat toolkit-automirror.txt

AutoMirror

- Fully automate session creation
- Automate time consuming tasks *(double-check identifiers)*
- Allow configuration via standard AWS methods *(Tags)*
- Set and forget

$ cat toolkit-automirror-demo.txt

AutoMirror DEMO

Plan B

SURICON

$ cat toolkit-automirror-demo.txt

Video of a similar demo

https://youtu.be/lZn4KDexC-4

SURICON

$ cat toolkit-config.txt

NSM Compliance

- Custom rule for AWS Config
- Automate technical state compliance
- Good fit for AutoMirror
- Can be used separately

SURICON

$ eog toolkit-config-demo.png

3CS-MirrorCompliance

Description	Technical state compliance for mirror sessions
Trigger type	Configuration changes
Scope of changes	Tags
Resources with tags	Mirror: True
Auto remediation	Off
Config rule ARN	arn:aws:config:eu-central-1:302806946273:config-rule/config-rule-i2zrbq
Parameters	*null*
Overall rule status	Last successful invocation on October 30, 2019 at 12:30:20 PM Last successful evaluation on October 30, 2019 at 12:28:32 PM

Choose resources in scope

Resources in scope represent those resources where this rule is being applied to and their compliance status.

Compliance status

Noncompliant

Resource ID	Resource type	Resource compliance status
i-067f76d952def8eed	EC2 Instance	Noncompliant

SURICON

$ cat toolkit-release.txt

AWS Mirror Toolkit

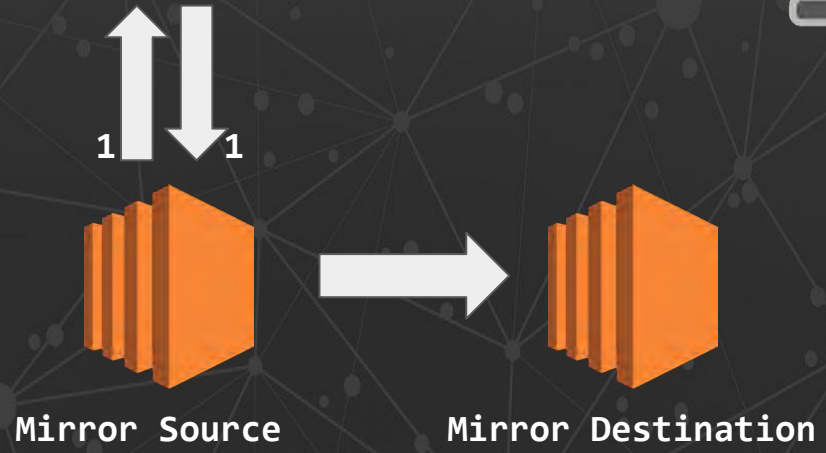
github.com/3CORESec/AWS-Mirror-Toolkit

SURICON

$ cat performance-considerations.txt

Traffic counts towards mirror source capacity.

Production traffic
>
Mirrored Traffic



4GB of traffic for source
2GB of traffic for destination

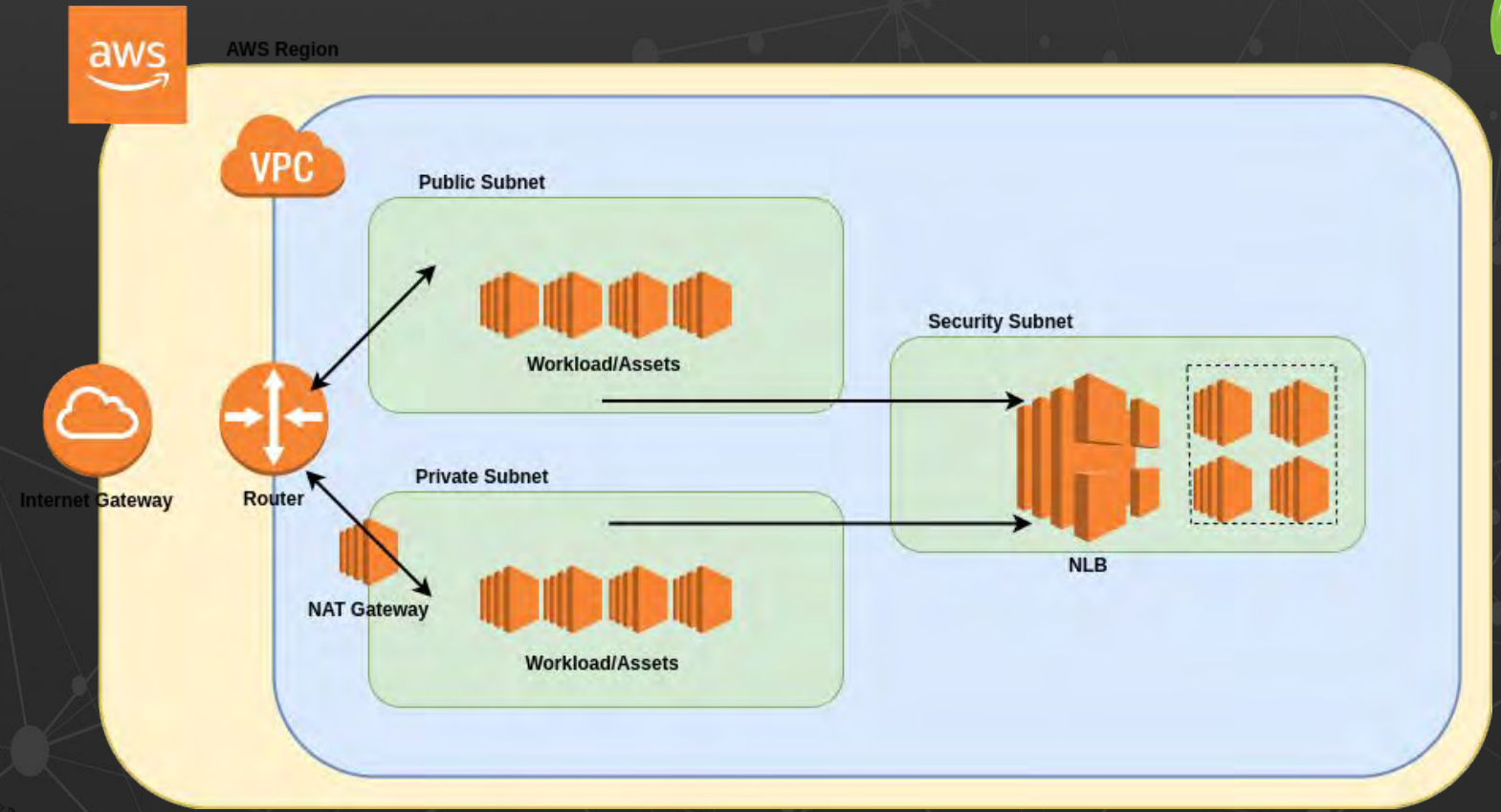
SURICON

$ cat nsm-aws-hpc1.txt

- **Enhanced Networking on Linux**
- Powered by Single Root I/O Virtualization (SR-IOV) for lower CPU utilization
- Higher bandwidth, PPS performance and lower inter-instance latency
- Available on Elastic Network Adapters (up to 100 Gbps)
- Example: EC2 C5n - Network Optimized
- Make use of Placement Groups: Cluster

Source: https://docs.aws.amazon.com/en_pv/AWSEC2/latest/UserGuide/enhanced-networking.html

$ eog nsm-aws-hpc2.png



Source: https://docs.aws.amazon.com/en_pv/AWSEC2/latest/UserGuide/enhanced-networking.html

$ info nsm-aws-hpc2.png

- Traffic destination: Network Load Balancer
- Flow hashing applied to traffic mirror
  - Protocol (UDP); Source IP; Source Port; Destination IP; Destination Port
- Behind NLB: EC2 C5n instances on ASG
- ASG launches instances with custom AMI
- Health check done to TCP port

$ eog nsm-deployment-types.png

- Hub and spoke model
- Replacement of VPC Peering
- Centrally managed routing/policies
- 50 Gbps

$ cat pre-guardduty.txt

Is there a place for NSM in cloud environments?

$ cat guardduty.txt | more

AWS GuardDuty is a managed service that continuously monitors malicious and unauthorized behaviour to protect AWS accounts, relying on CloudTrail, VPC Flow Logs and DNS logs.

SURICON

--MORE--

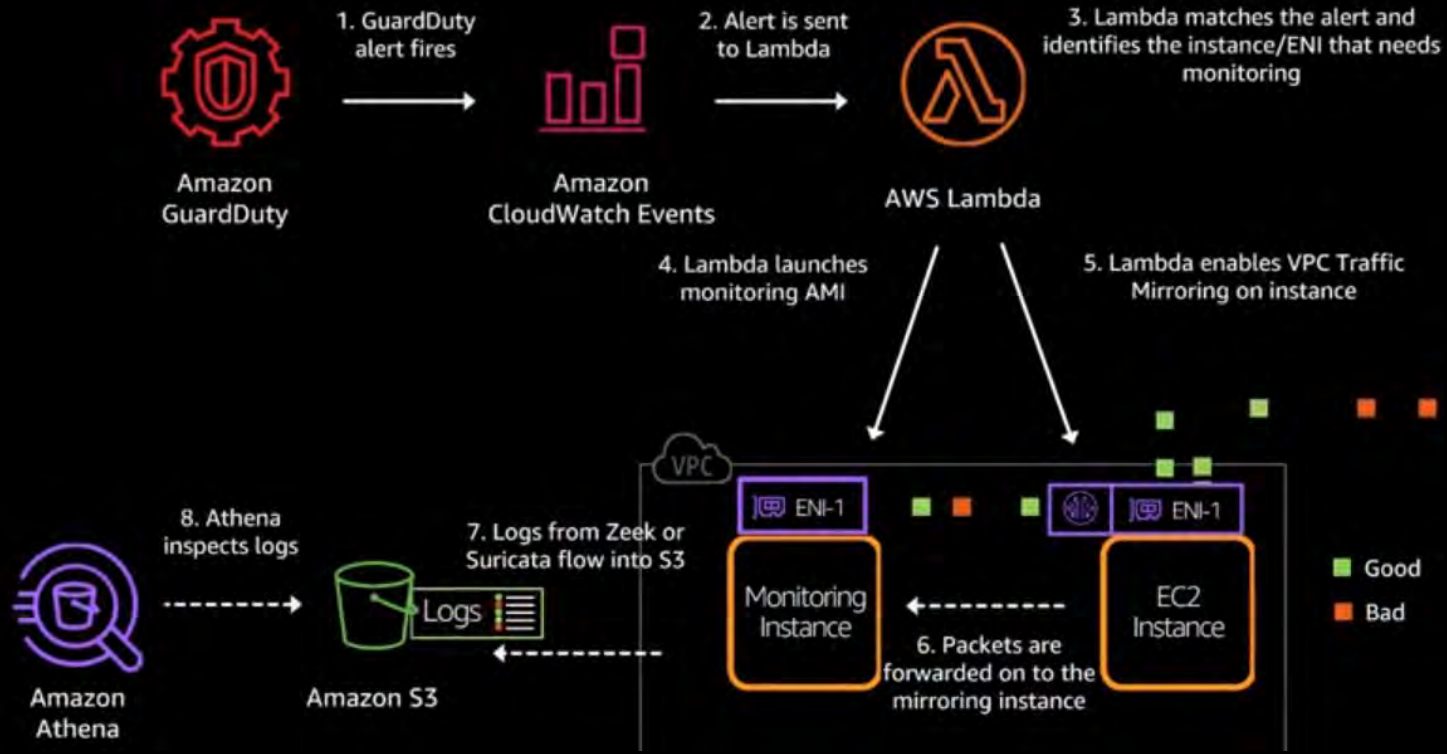
- Application & Network
- Machine Learning
- 1-click enabled
- Lambda execution for remediation actions

“Threat intelligence coupled with machine learning and behavior models help you detect activity such as crypto-currency mining, credential compromise behavior, communication with known command-and-control servers, or API calls from known malicious IPs.”

SURICON

Source: https://aws.amazon.com/guardduty/

$ eog suricata-at-amazon-retail.png



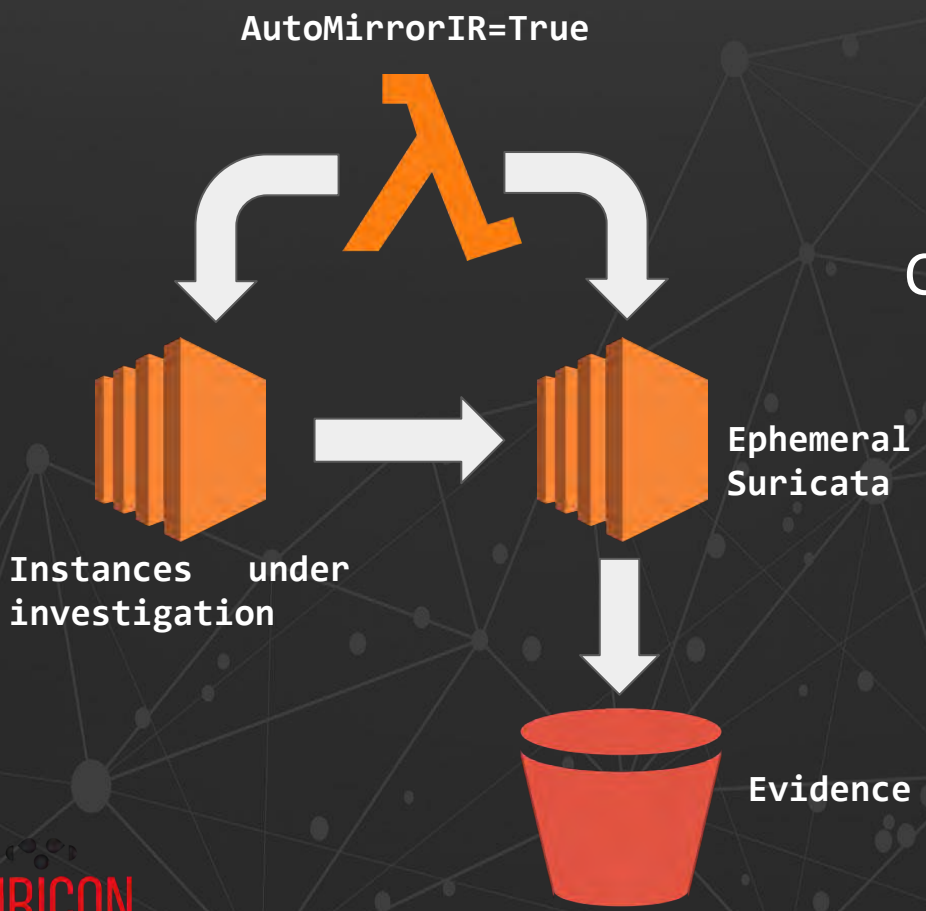
Source: AWS re:Inforce 2019: SEP209 (Youtube)

$ cat nsm-ir.txt

Example: AutoMirror in IR

SURICON

$ cat automirror-ir.txt



Coming to the toolkit ...

Soon!

ish

SURICON

$ cat nsm-resilience.txt

In an environment with properly configured IAM policies and groups, tampering with traffic collection is not possible, making it resilient against manipulation and tampering.

$ cat closing-remarks.txt

- New way of looking at cloud-based NSM
- Interesting challenges and opportunities
- Serverless visibility?
- HPC NSM (Suricon 2020?)
- New security & networking challenges

$ cat questions.txt

Questions?

 @0xTF

SURICON