Global Information Assurance Certification Paper

Packet Capture on AWS

GIAC GCCC Gold Certification

Author: Teri Radichel, teri@radicalsoftware.com
Advisor: Dave Hoelzer
Accepted: August 10th 2017

Abstract

Companies using AWS (Amazon Web Services) will find that traditional means of full packet capture using span ports is not possible. As defined in the AWS Service Level Agreement, Amazon runs certain aspects of the cloud platform and does not give customers access to physical networking hardware. Although access to physical network equipment is limited, packet capture is still possible on AWS but needs to be architected in a different way. Instead of using span ports, security professionals can leverage the software that runs on top of the cloud platform. The tools and services provided by AWS may facilitate more automated, cost-effective, scalable packet capture solutions for some companies when compared to traditional data center approaches.

1. Introduction

As companies move to AWS, Amazon's cloud platform, security teams face challenges trying to secure an environment which has different levels of access compared to a traditional data center. Some new tools and services offer innovative ways to secure networks and applications. Other controls limit access to network controls security teams are familiar with using to secure environments.

This paper builds on the concepts explained in my previous SANS Gold Paper, "Balancing Security and Innovation with Event Driven Automation" (Radichel, 2016). In that paper, I demonstrated automated intrusion detection and response on AWS. AWS Flow Logs captured network traffic that had a destination port of 0. It is highly likely that traffic sent to port 0 was an error or nefarious because in most cases no one would purposefully send traffic to this port (Hall, 2009). AWS Flow Logs only provides netflow data which shows the source destination and ports but does not give enough information to determine if that hypothesis is true or false. To figure out if the packets are malicious, a security professional would want to look at the full packet with the payload, options and other information not included in the AWS Flow Log data. Security teams need to implement a different kind of logging solution to capture full packet data for this type of deeper analysis.

Since network hardware on AWS is not accessible due to the Shared Security Responsibility Model explained in the Amazon Web Services: Overview of Security Processes white paper (Amazon Web Services, 2017k), security teams will need to use a software solution that works on the AWS Cloud Platform for packet capture. The network architecture must ensure all packets flow through the packet capture solution. The implementation will need to secure the logs and the packet capture appliance. The solution should also consider the cost (see the Appendix), scalability, long term maintenance, and ability to respond quickly to events.

Teri Radichel, teri@radicalsoftware.com

2. Packet Capture Options

2.1. Network Packet Capture Differences on AWS

Amazon offers a virtual experience like operating infrastructure in a private data center. On an EC2 instance, the customer has control over the host operating system configuration. For network security, the customer can define custom routes, and rules. Network packets on AWS are like any other network packets with a few exceptions, as outlined below:

- Customers have no access to hardware routers and switches to attach packet capture devices to the network (Amazon Web Services, 2017k).
- AWS applies custom headers after the packet leaves a source host and removes them before receipt by the destination host; from the host perspective, the data will look the same (Mueller, 2016).
- The AWS custom mapping service removes the need for ARP, which does not exist on AWS (Matthews, 2016).
- A security appliance will not be able to intercept traffic from one endpoint to another in the same VPC (Matthews, 2016).
- VPC Flow Logs exclude some types of traffic (Amazon Web Services, 2017e).
- Promiscuous mode on a network interface will not show traffic from other hosts (Amazon Web Services, 2017k).

Many concepts translate directly from data center networks to the AWS cloud environment, but the software used to implement networks and some of the terminology will be different. This AWS document defines AWS specific terminology and concepts in such a way that network engineers used to working in a traditional data center can understand the AWS environment and tools: https://aws.amazon.com/blogs/apn/amazon-vpc-for-on-premises-network-engineers-part-one/

2.2. Packet Capture on an EC2 Instance

An EC2 instance offers root access to a host operating system. A customer can install any required software on the host. It is possible to install Wireshark on an instance and capture packets or run tcpdump and output the packets to a pcap file. For this reason, packet capture executed on a single Linux or Windows host would be the same on AWS as any other environment with the few exceptions noted in the last section.

Because every host can capture packets, a security team could implement a mechanism whereby every host continuously logs packets to an S3 bucket. Although this scenario is possible, it would be problematic for the following reasons:

- To securely log to an S3 bucket, the IAM role on every host would need access to an S3 log bucket (Amazon Web Services, 2017g).
- Because an EC2 instance or Lambda function can have only one IAM role profile, every EC2 or Lambda role would need this S3 access added to its permissions (Amazon Web Services, 2017l).
- A bucket policy would need to allow any EC2 instance or Lambda role to write to the S3 bucket (Amazon Web Services, 2017g).
- Every host would need a software configuration that includes the application or scripts that log to the packets to the S3 bucket.
- Every host would need access to S3. NACLs, security group rules and routes on route tables will need to allow access to the S3 IP ranges published here by Amazon: http://docs.aws.amazon.com/general/latest/gr/aws-ip-ranges.html
- Monitoring could ensure that every host was correctly logging to the S3 bucket trigger alerts on failure.
- If packet capture fails, individual hosts would need inspection.
- To encrypt log files, all hosts would need access to encryption keys or mechanism.

- Every host would have overhead associated with logging packets; any issue with the packet logging could negatively affect the applications running on that host (Poritskiy, 2017)

Not only is this implementation challenging from an operational overhead perspective, but it also poses added security risks. In an interview with Computer World, security expert Bruce Schneier said, "Complexity is the worst enemy of security" (Chan, 2012). The complexity of this packet capture solution creates many points of potential compromise. Every host needs a secure configuration. Malware on any compromised host may turn off this logging or alter logs to produce false information. Many points of failure exist where misconfiguration or interruption of networking any point could lead to missing or corrupted data. An attacker has multiple points at which to obtain access to alter data in transit or within the S3 bucket.

2.3. Routing Packets Through a Security Appliance

Instead of configuring every host, a security appliance can capture the traffic and then send it to the ultimate destination (Purcell, 2013). A well-architected design will ensure packets cannot reach another network without passing through this host. When the security appliance receives traffic, it can inspect, reject, or log the packets (SANS Institute, 2016).

Traditionally, a device connected to the span port of network hardware would capture and inspect traffic. Any approach requiring access to physical network hardware and wires will not be possible as AWS customers will not have access (Amazon Web Services, 2017k). To get around this limitation, AWS customers can create network routing that forces traffic to pass through security appliances. AWS offers customers the ability to define subnets, routes, and route tables. Routes will limit the paths data can take as it traverses the different networks internal and external to AWS. Customers can set up instances that receive the data forward the data to the intended destination.

Rather than rely on host configuration, security appliances can live at privilege boundaries. Privilege boundaries are points where the level of permissions change within

an application flow (OWASP, 2017) and the same concept can apply to network architecture and threat modeling. For the packets to pass from one segment of the network with different levels of trust than another, the traffic must pass through the security appliance. A privilege boundary could be at one or more of the following points in the network:

- AWS to the Internet
- AWS to the corporate data center (hybrid cloud)
- AWS to a partner network
- AWS to another cloud provider
- One AWS region to another region
- One AWS account to another AWS account

The following sections will explain different architectures involving a security appliance to achieve full packet capture on AWS. Each architecture should have a different network configuration as shown in the related diagrams, but certain principles always apply. Every route must force traffic through the security appliance before it can reach another network segment. Any misconfiguration could allow packets to bypass logging and inspection. Only the security appliance should exist in the subnet designated to connect to other networks. Route tables must force traffic destined for other networks to the host which can receive and send the traffic to the ultimate destination, and that no alternate routes exist. Monitoring can ensure that packet capture is successful and can respond to failure by disallowing traffic or alerting an administrator.

Assigning a small group of hosts to collect traffic rather than every host on the network incurs less overhead because there are fewer hosts to manage:

- Packet capture no longer relies on the configuration of every host in the network to perform packet capture.
- This design ensures all traffic routes through a limited number of hosts assigned for traffic inspection and logging.

- A restricted set of hosts with write access to the log bucket reduces the risk of tampering or inadvertent corruption or deletion.
- A smaller highly skilled security and networking team can manage the network and packet capture hosts.

3. Packet Capture Architecture

3.1. Overview

A packet capture architecture on AWS will consist of network routing and hosts which can receive, forward, and capture packets. Instead of a span port on a network appliance, the design must route all the traffic to the security appliances used to capture the packets, ensure the data cannot bypass the packet capture, and secure the packet capture hosts and the logs. Two approaches to implement packet capture on AWS include a NAT Architecture or a Proxy Architecture.

3.2. NAT Architecture

Amazon allows customers to create a NAT instance on AWS (Amazon Web Services, 2017j). The NAT instance sends traffic from the VPC to the Internet via the AWS Internet Gateway and converts private IP addresses to public IP addresses. A NAT instance offers a way for hosts that exist in a private network to reach the Internet without exposing the hosts directly to the Internet. Often NAT instances allow other EC2 instances in private networks to reach software update services hosted by software vendors.

NAT instance configuration differs from a typical EC2 instance. Amazon offers NAT AMIs which customers can deploy in their AWS accounts. A route associated with subnets in the VPC that need access to external networks will reference the NAT instance ID. Traffic destined for that route will traverse the NAT instance. Note that intra-VPC traffic will not traverse the NAT. As previously explained, traffic within the VPC will always be point-to-point between the source and destination host. This AWS NAT instance documentation includes the diagram in Figure 1 describes the typical way to

deploy a NAT on AWS:

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_NAT_Instance.html

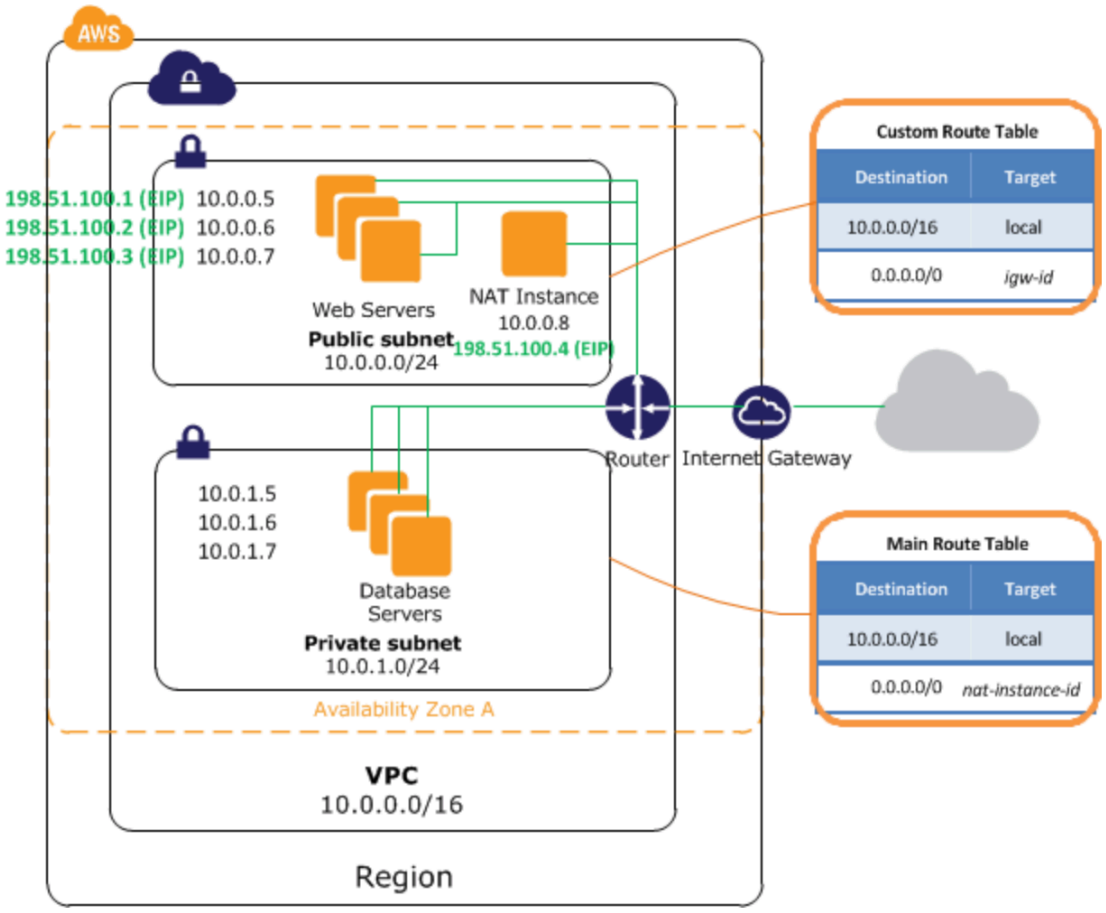


Figure 1. Outbound-Only NAT Configuration (Amazon Web Services, 2017j)

As shown in the above sample architecture from Amazon, hosts that exist in the NAT subnet and have a route to the Internet can bypass the NAT. NATs typically handle outbound traffic, but the NAT instance can receive inbound traffic and forward to hosts inside the private network according to the NAT documentation (Amazon Web Services, 2017j). To ensure all traffic traverses the NAT instance, create two or more Interfaces on the NAT instance. Assign an external IP address to one of the interfaces in a subnet with a route to other networks. Do not put any other hosts in that subnet to avoid any traffic inadvertently bypassing the NAT instance. Assign the rest of the network interfaces to

private subnets with no access outside the VPC. Use the private network interface ID in the route tables used by subnets for any other hosts. Any traffic must then traverse the NAT to come from or reach external networks as shown in Figure 2 below:

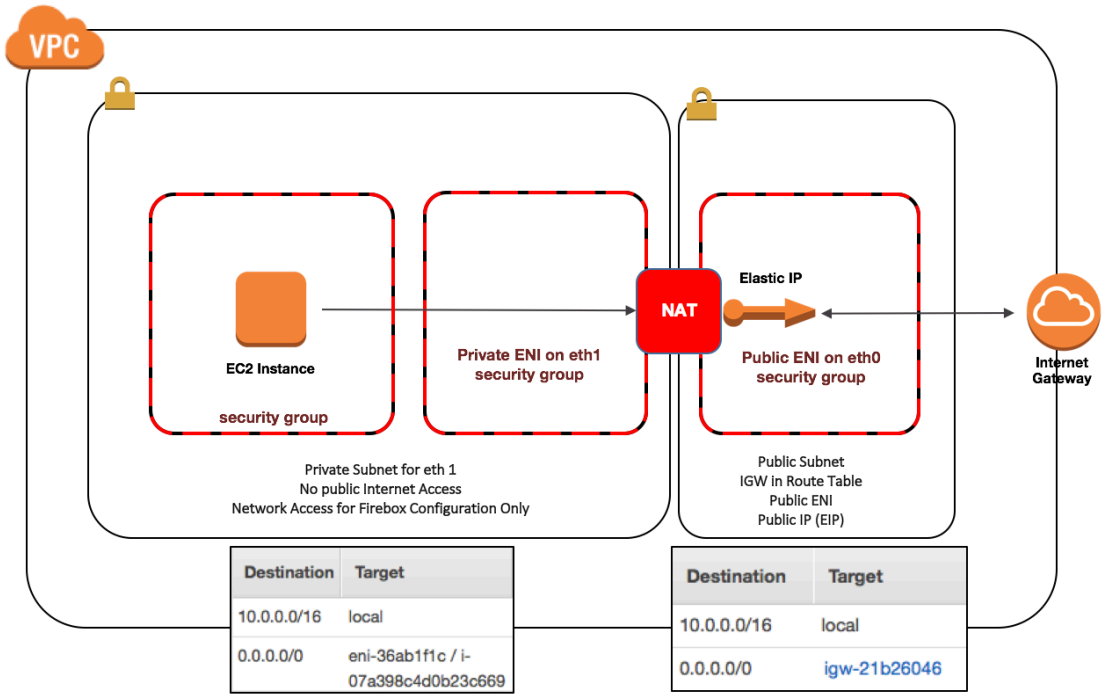


Figure 2. NAT Architecture – NAT Captures Packets

A NAT instance receiving traffic from hosts can log the traffic via tcpdump to an S3 bucket or other desired location instead of having every single host capture traffic. For a high availability configuration, deploy multiple instances in a high availability configuration as defined in the AWS documentation: https://aws.amazon.com/articles/2781451301784570 (Varia, 2015). AWS offers a NAT Gateway to give customers high availability without having to implement the high availability configuration. It does not appear that an administrator can alter the AMI used for the NAT Gateway AMI (Amazon Web Services, 2017i). For this reason, customers cannot change the NAT Gateway configuration or use it for packet capture.

3.3. Load-Balanced Proxy Architecture

Another approach to architecting a packet capture solution involves implementing a load-balanced proxy architecture with EC2 instances that live in an Auto-Scaling Group behind an Elastic Load Balancer (ELB). The ELB distributes traffic to instances which act as traffic proxies. An AWS Auto-Scaling Group will add new hosts as the traffic increases and terminate hosts as it decreases.

Amazon offers this example configuration of a Squid Proxy shown in Figure 3 which acts as an outbound proxy for layer 7 HTTP and HTTPS traffic. In this configuration, hosts proxy traffic from a private network to the Internet. Internal hosts cannot directly access the Internet. These hosts need to reach services on the Internet. The requests go to the Squid hosts which in turn send the request to the Internet and return the response to the requesting host. https://aws.amazon.com/articles/6463473546098546

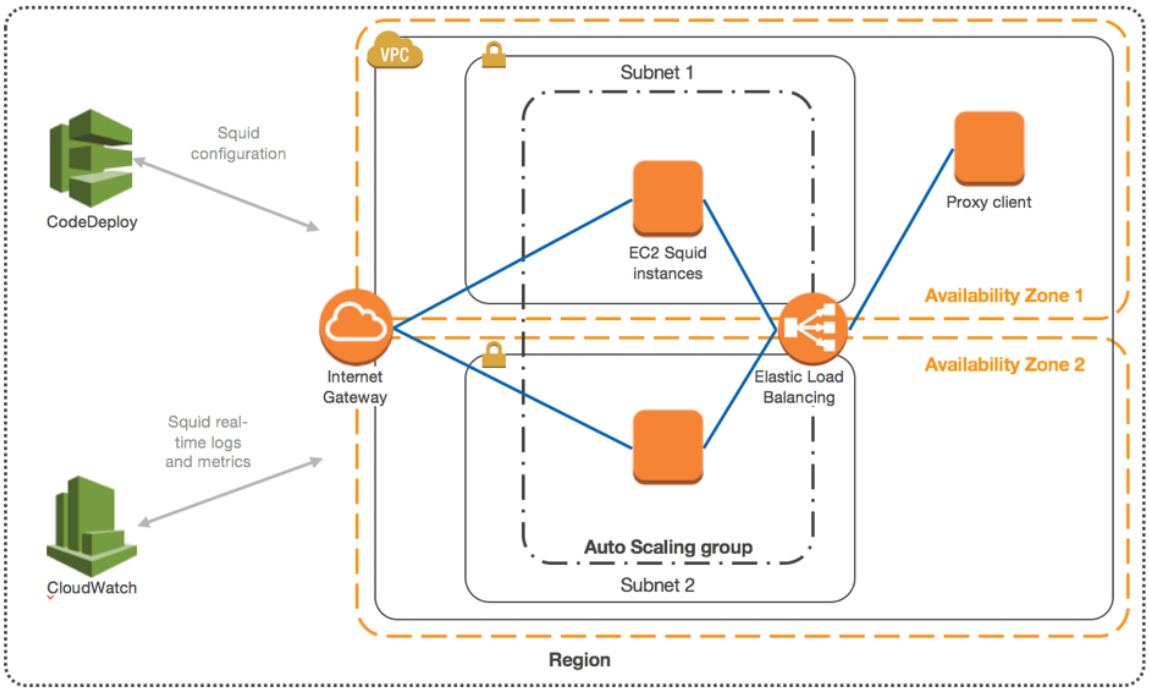


Figure 3. Load-Balanced Proxy Architecture

A proxy architecture will not use an AWS NAT route table entry. Proxy hosts are standard EC2 instances running proxy software. The hosts receive requests for traffic and send the traffic to the destination. The proxy hosts could log packets as the traffic passes

from the internal network through the proxy host to the external network. Various types of proxies work at different layers in the OSI or TCP models to implement this architecture.

Squid proxy only handles HTTP and HTTPS traffic. This architecture also only handles outbound traffic only. By default, Squid proxy caches web pages to return requested content more quickly which may or may not be desirable. The Squid Proxy configuration rules can limit which URLs hosts on the network can access. Other types of proxies will work at a lower level of the network stack.

The proxy implementation must consider state management if needed. A proxy may receive traffic and only pass it along, or it may try to put the packets back together and conduct a more detailed analysis. As with any of the solutions described here, administrators need to ensure traffic can only reach the other networks via the packet capture host. No paths should exist which may bypass the proxy.

One of the benefits of the proxy architecture is the ability to scale automatically. A proxy configuration also overcomes VPC peering limitations with transitive routing (Amazon Web Services, 2017h). VPC peering will only allow traffic between directly peered VPCs. If a host tries to send traffic to a host in a VPC that is only accessible via a route through a third VPC, that request will fail. With a proxy architecture, Amazon will allow proxied traffic because it will appear as if it is coming from the proxy in the directly peered VPC as shown in Figure 4.

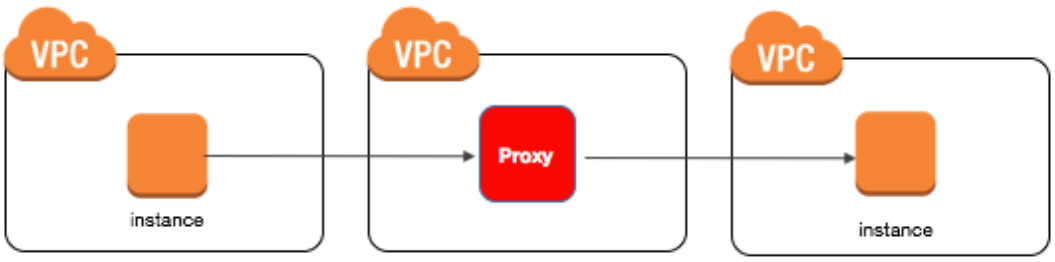


Figure 4. Proxy to overcome VPC peering limitations.

Some downsides exist with a proxy configuration. Every host requires additional configuration to send packets through the proxy (Amazon Web Services, 2017m). The configuration on each host specifies where to send the traffic and on what ports. A misconfigured host would send requests to the wrong destination. A correctly defined network will block the misconfigured traffic. A misconfigured network would allow traffic to bypass packet capture. The proxy software needs to be able to handle all types of traffic to capture all packets, not just HTTP and HTTPS. Alternatively, a network admin may only let HTTP and HTTPS traffic originate from the internal network. As with a NAT, any stateful packet inspection would need to handle session management across multiple hosts and handle termination of instances correctly to ensure complete packet capture and graceful failover.

3.4. Security Appliances for Packet Capture

Many vendors offer security solutions in the AWS Marketplace (https://aws.amazon.com/marketplace), where customers can choose a BYOL (bring-your-own-license) or a pay-as-you-go model, depending on the offerings from the vendor. Security appliances may act as a next-gen firewall, UTM (Unified Threat Management) or other types of traffic inspection hosts. Compared to a NAT or proxy alone, a security appliance has the added benefit of inspecting traffic as it traverses the hosts and many will handle both inbound and outbound traffic.

The operating system of the vendor AMI may offer a means for full packet capture. In this case, the hosts handle all the processing and storing of packet data and will need to handle the required load. Depending on the licensing model of the product, customers may incur more fees as new hosts come online. In this configuration, any issues related to packet capture could slow down network requests as well. A compromised host may in turn compromise logs stored directly on the appliance and storage locations accessible to the instance. AWS destroys any ephemeral data on a terminated instance so packet capture implementation should ship the logs to an alternate storage location.

The cost of transferring log data may also affect the choice of a security appliance. Traffic coming into AWS is free. Traffic leaving AWS incurs outbound data transfer fees. Traffic between peered VPCs or between regions also incurs charges but at a lower price point. Some customers may incur lower costs using Direct Connect (Amazon Web Services, 2017a). For the current pricing associated with outbound data transfer and Direct Connect refer to AWS VPC pricing (https://aws.amazon.com/vpc/pricing). For an explanation of calculating the cost of different architectures on AWS see the Appendix.

If the security appliance has a means for capturing packets via an API or CLI command, a separate host can call that command to log traffic data as shown in Figure 5. The packet logging host can exist in a private subnet which has access to the required management ports on the security appliance. A private network interface on the security appliance will be accessible from a network dedicated for this purpose. Only hosts in this designated network will have access to the log storage location. The packet capture host will not directly have access to the log data which AWS customers can store in a private network, not accessible from the Internet or other locations.

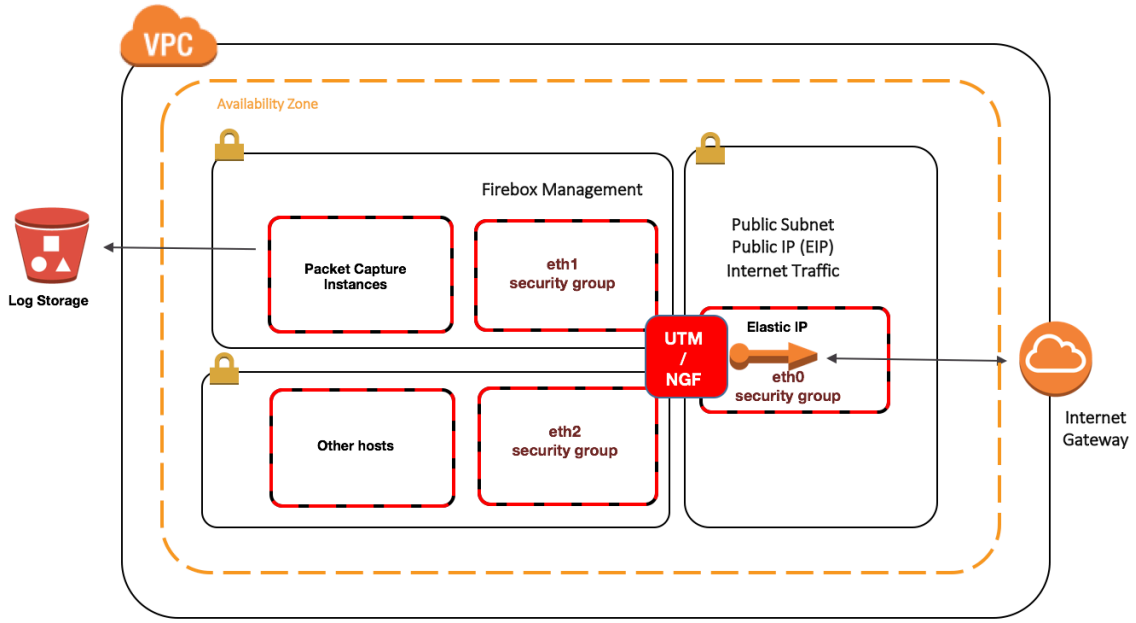


Figure 5. Security appliance with multiple interfaces and network segmentation

All other hosts would then connect to the security appliance on a separate private interface and would not have access to the logs or management ports on the security appliance. Using an alternate host configuration for full packet capture, different hosts can handle some of the processing such as compression, storage, and analysis. The system design could ensure that if logging fails, the security appliance suffers minimal impact – or if logging is mandatory, traffic should stop altogether and send alerts.

3.5. Transit VPC

A transit VPC (Barr, 2016) uses a hub and spoke architecture to connect different AWS VPCs and non-AWS networks (See Figure 6). As described by AWS Solution Architect, Rob Alexander in his AWS re:Invent presentation, *From One to Many: Evolving VPC Design* (Alexander, 2016), connecting every account to every account results in excessive complexity. By routing all the traffic through a transit VPC there are fewer connections to manage and AWS customers incur less operational overhead.

Routing all traffic through a transit VPC offers an added security benefit as well. VPNs can connect the networks traversing the transit VPC and perform packet capture at the same time. Customers can use these connections to direct traffic through a transit VPC before it crosses network boundaries, logging, and monitoring all data flowing in and out of AWS accounts and VPCS. Having a single point of access to and from external networks limits the points at which attackers can infiltrate malware and exfiltrate data.

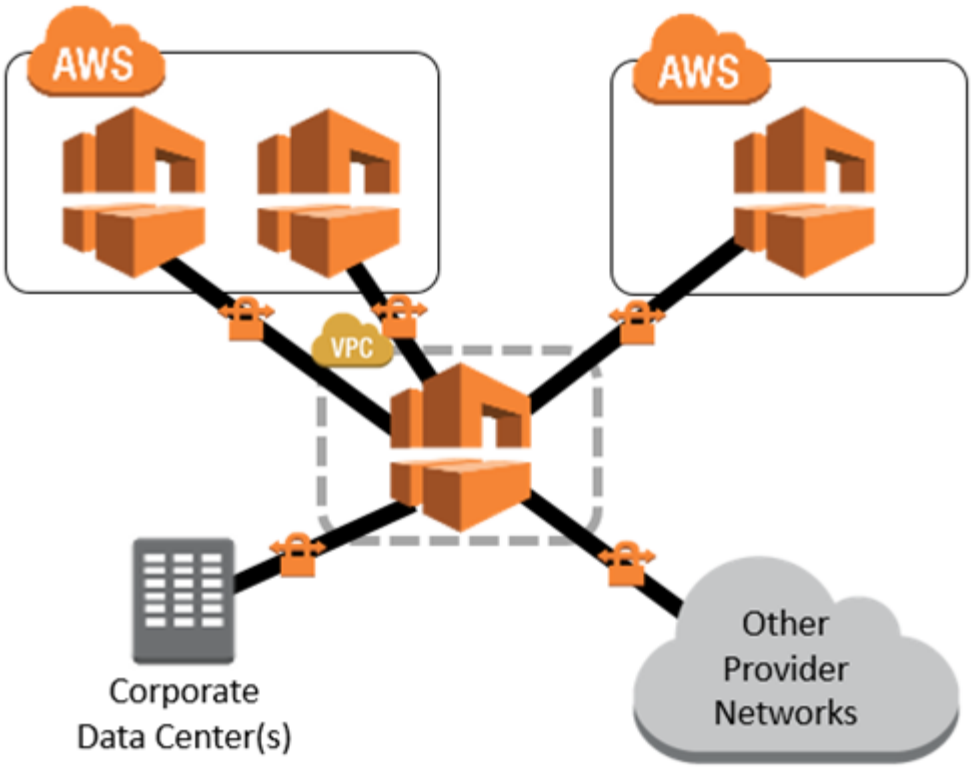


Figure 6. Transit VPC (Barr, 2016)

A transit VPC could be a single point of failure for an entire ecosystem of networks if not architected for high availability. The design for this networking infrastructure must be highly scalable and resilient. Some customers may need a multi-region configuration with automated failover. Complete automation and the ability to quickly re-deploy the network may be enough in other cases. VPC peering connections have a cost associated with the traffic passing over them as mentioned previously. Companies need to weigh the cost of data transfer versus a breach. Having a single point in the network for added scrutiny by a highly skilled network and security team may be worth the investment to prevent downtime and data loss.

When creating a packet capture solution in a multi-region environment, an AWS customer should consider a replicated architecture in each region. Sending traffic from one AWS region to another real-time may produce delays due to latency and other issues as the traffic traverses the Internet. Capturing the packets in one region and then

replicating to another as needed will be a better approach for performance reasons. Compression before transferring data will reduce costs. Cross-region replication for AWS S3 may be a good choice for consolidating packet capture files. S3 will automatically copy files in a bucket to an alternate region Barr, J. (2015, 03 24). Companies will need to consider cross-region encryption and key management as well.

3.6. Secure Log Storage

Amazon offers many different services for data storage. Which service a company chooses will depend on factors such as time to implement, cost, length of storage, amount of data, type of data, speed to retrieve data, encryption, and data processing requirements. Explaining all the AWS service storage options is beyond the scope of this paper, but the *AWS Storage Services Overview* white paper covers this topic. (Amazon Web Services, 2016).

Secure storage of packet capture files will ensure they are available in the case of an event or incident that requires investigation. Storage and handling processes and policies should consider the chain of custody in the event legal proceedings need the logs to prove damages in a court case. Cases have occurred in which employees or attackers deleted data belonging to a company causing irreparable harm. This action put Code Spaces out of business (Marks, 2014). Code Spaces is the most widely known case in which unauthorized access to an account allowed an attacker to delete so many resources the company went out of business. Organizations should store logs in a manner that prevents people with malicious intent from deleting data.

Companies storing data in S3 should configure an S3 Endpoint unless it needs to leave the AWS network (Amazon Web Services, 2017g). Administrators should implement restrictive policies on the bucket, the endpoint, and IAM roles to limit access. (Amazon Web Services, 2017l). S3 versioning could track any improper changes or deletion of log files (Amazon Web Services, 2017f). Replication could copy files to a separate read only bucket in another account, to which the security appliance has no access (Amazon Web Services, 2017d). Policies could limit access to read logs to specific users from explicit source IP addresses and require multi-factor authentication.

4. Packet Capture – A Working Example

4.1. Overview

The following sample code shows how to automate the deployment of a security appliance on AWS which is used to facilitate packet capture. As with the example in "Balancing Security and Innovation with Event Driven Automation" (Radichel, 2016), the source exists in a Github repository and the README.md file explains how to download and run the code.

https://github.com/tradichel/PacketCaptureAWS

This source code deploys a WatchGuard Firebox Cloud as a NAT instance in an AWS VPC. All traffic in the VPC flows through the NAT when traversing the network to or from the Internet. The Firebox has a public network interface for access to and from the Internet, one private network interface for management of the Firebox and one private interface for all other hosts as shown in Figure 7 below:

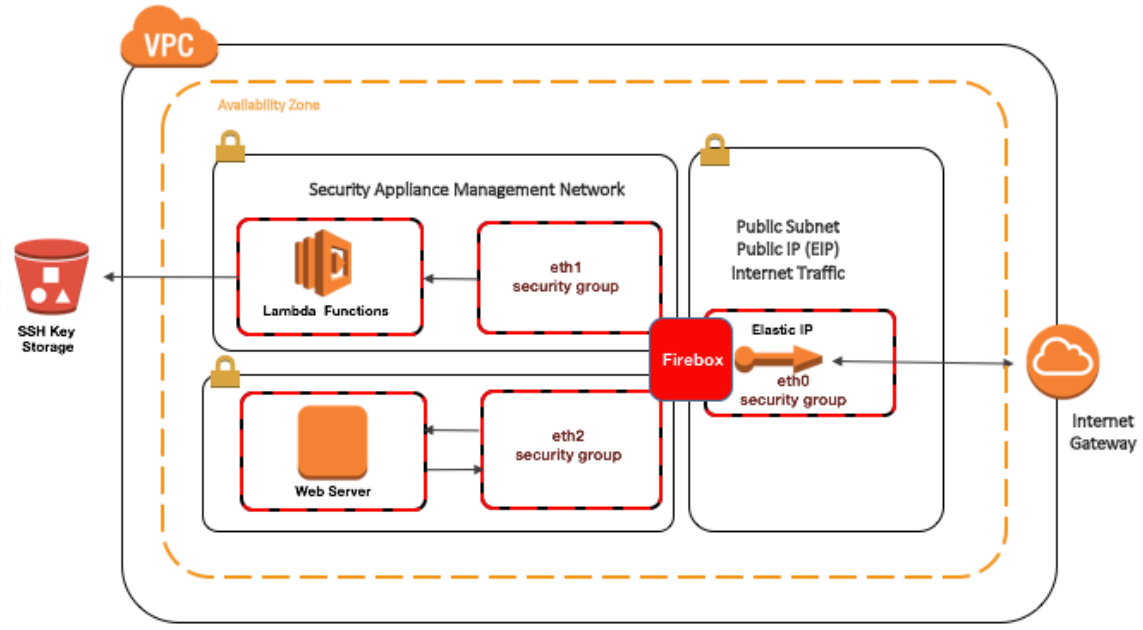


Figure 7. Sample packet capture architecture with AWS Lambda functions.

The image shows subnets and security groups which segregate traffic. Internet traffic must pass through the Firebox. The port for the CLI (command line interface) is only accessible from the management network. Subnet routes, NACLs, and security group rules limit access to the S3 bucket.

4.2. Protecting the Key in the S3 Bucket

The Firebox offers configuration via a Command Line Interface (CLI) using an SSH connection. Amazon provides what it calls EC2 Keys for EC2 instances which are SSH keys. The script generates an EC2 key which an administrator or software can use to execute Firebox CLI commands. In this case, a Lambda function executes the commands and the CLI is only accessible from within the VPC. The script uploads the key to an S3 bucket which is only accessible via an S3 endpoint. The endpoint, networking and bucket policies limit access as follows:

- The key is only available for download within the defined management network.
- The key is only accessible for download by a Lambda function or EC2 instance with the correct IAM role profile and security group.
- The Firebox ENI does not have access to download the key because it does not have the role or security group that allows access to the S3 bucket.
- Administrators have access to upload the bucket when coming from a defined CIDR block.
- Administrators must have MFA enabled to upload to the bucket.
- The bucket policy enforces server-side encryption of all files placed into the bucket.
- Only the subnet for the management network has a route that allows access to the bucket via the S3 endpoint.

4.3. AWS Resources and Firebox Configuration

For this packet capture demonstration, the code deploys a Firebox and all the AWS resources shown in Figure 7. The scripts create networking, interfaces, an Elastic IP address, roles, policies and a KMS key. A web server is instantiated to test sending traffic to and from a host in the VPC.

Two AWS Lambda functions configure the Firebox. The first function, "ConfigureFirebox" configures the Firebox to act as an NTP server and adds rules needed for software updates by EC2 instances. A second lambda function, "ConfigureSNAT" enables traffic from the Internet to reach a web server. A third Lambda function captures packets sent to the Firebox at the time of execution. The "CapturePackets" Lambda function uses the Firebox CLI tcpdump command (WatchGuard Technologies, 2017) and prints the output to the CloudWatch Logs associated with the Lambda function.

4.4. Run the Code

A combination of bash scripts, Python, the AWS CLI (command line interface) and AWS CloudFormation deploys the sample architecture. Executing ./run.sh and typing "Y" at the prompt to use default values. After this point, deployment, and configuration of all the AWS resources completes with no further manual steps. The script creates all the resources, configures the security appliance, executes packet capture using tcpdump, and stores the log files in an S3 log bucket. For a more detailed explanation of a simpler version of the same script refer to "How to Automate Deployment of a WatchGuard Firebox Cloud on AWS" (Radichel, 2017)

Bash is a tool many security and network professionals have used so will be familiar. As noted, this paper extends the information presented in "Balancing Security and Innovation with Event Driven Automation" (Radichel, Sans Institute Reading Room, 2016) which explained CloudFormation. Since then, AWS altered CloudFormation to support both JSON and YAML. YAML is simply another format to represent data and this code uses the YAML format. The example code in the last paper used Node.js as the programming language for the AWS Lambda. This example creates Lambda functions

that run Python. As before, the README.md file has instructions for setting up the AWS environment and retrieving the code.

https://github.com/tradichel/PacketCaptureAWS

4.5. View the Packets

Once the script completes execution, the "CapturePackets" Lambda function should have the packets in the logs. To view captured packets, log into the AWS Console, navigate to the Lambda service and click on the Packet Capture Function in the list as shown in Figure 8:

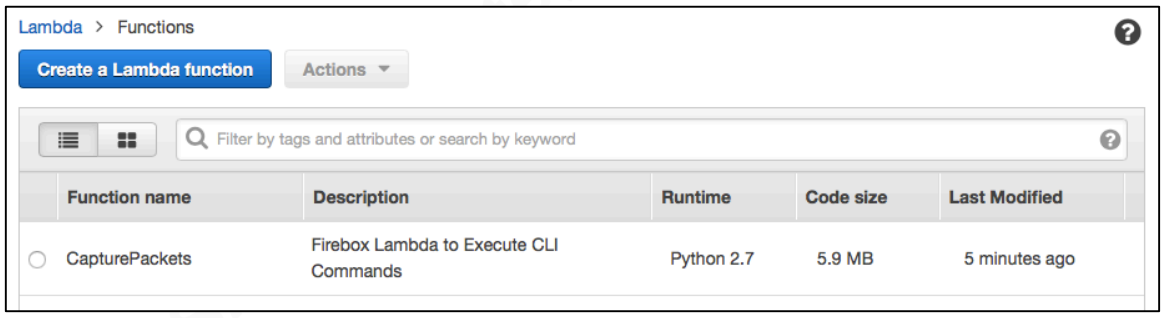


Figure 8. CapturePackets Lambda Function

Click on Monitoring, then View Logs in CloudWatch.

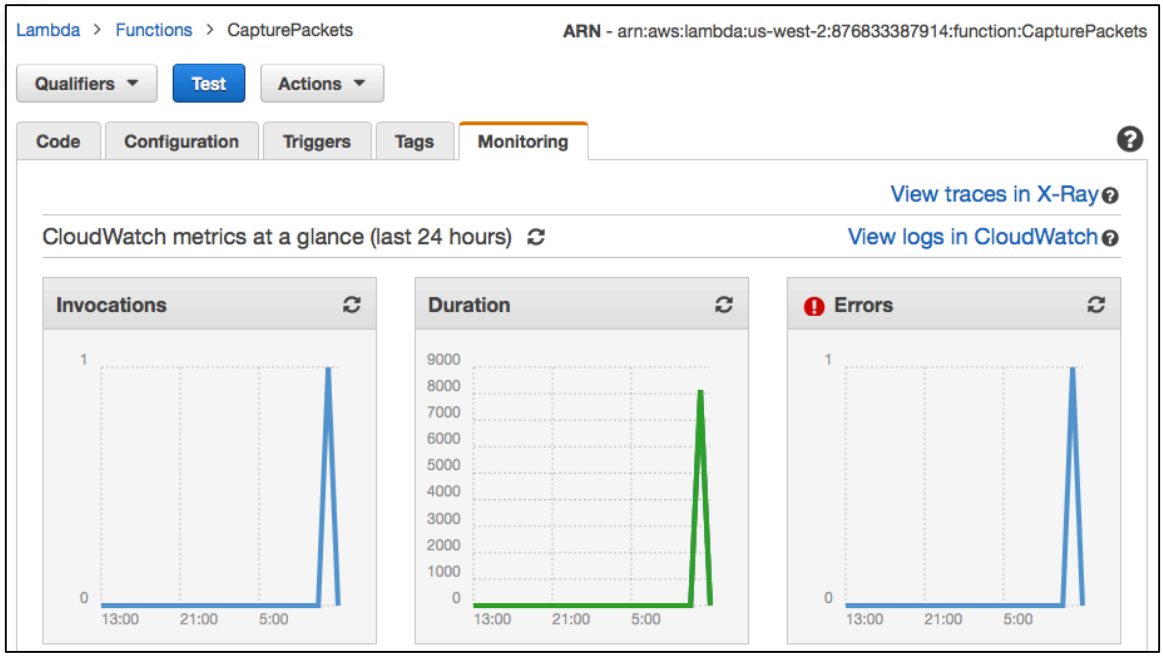
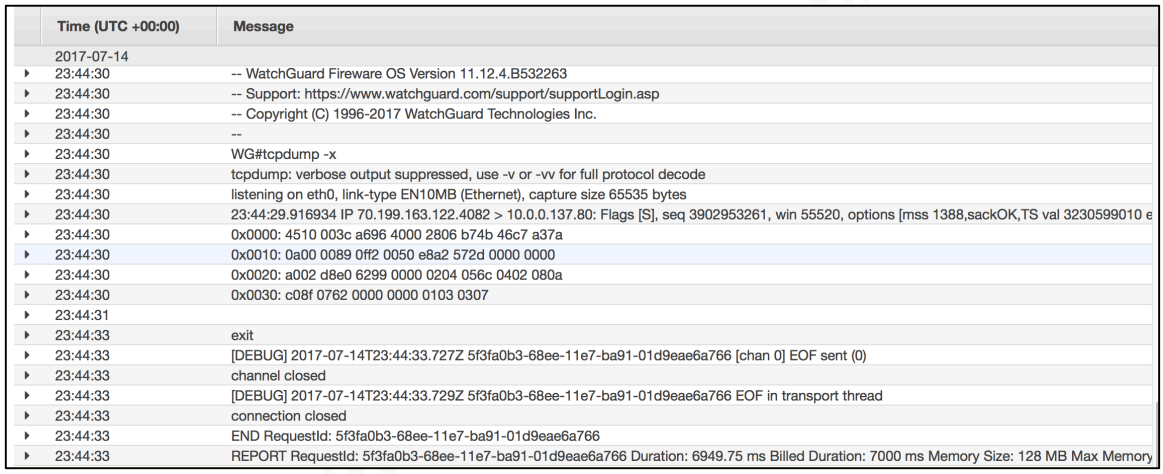


Figure 9. CloudWatch monitoring for AWS Lambda functions.

Click on the item in the list of logs to view the details from that execution which will show the output of the packet capture.



Time (UTC +00:00)	Message
2017-07-14	
23:44:30	-- WatchGuard Fireware OS Version 11.12.4.B532263
23:44:30	-- Support: https://www.watchguard.com/support/supportLogin.asp
23:44:30	-- Copyright (C) 1996-2017 WatchGuard Technologies Inc.
23:44:30	--
23:44:30	WG#tcpdump -x
23:44:30	tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
23:44:30	listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
23:44:30	23:44:29.916934 IP 70.199.163.122.4082 > 10.0.0.137.80: Flags [S], seq 3902953261, win 55520, options [mss 1388,sackOK,TS val 3230599010 e
23:44:30	0x0000: 4510 003c a696 4000 2806 b74b 46c7 a37a
23:44:30	0x0010: 0a00 0089 0ff2 0050 e8a2 572d 0000 0000
23:44:30	0x0020: a002 d8e0 6299 0000 0204 056c 0402 080a
23:44:30	0x0030: c08f 0762 0000 0000 0103 0307
23:44:31	
23:44:33	exit
23:44:33	[DEBUG] 2017-07-14T23:44:33.727Z 5f3fa0b3-68ee-11e7-ba91-01d9eae6a766 [chan 0] EOF sent (0)
23:44:33	channel closed
23:44:33	[DEBUG] 2017-07-14T23:44:33.729Z 5f3fa0b3-68ee-11e7-ba91-01d9eae6a766 EOF in transport thread
23:44:33	connection closed
23:44:33	END RequestId: 5f3fa0b3-68ee-11e7-ba91-01d9eae6a766
23:44:33	REPORT RequestId: 5f3fa0b3-68ee-11e7-ba91-01d9eae6a766 Duration: 6949.75 ms Billed Duration: 7000 ms Memory Size: 128 MB Max Memory

Figure 9. CloudWatch Logs with packet capture data.

Note that if no traffic was traversing the Firebox at the time of execution, the Lambda function has nothing to write to CloudWatch. Simulate traffic by opening a browser and typing in the IP address of the web server (http://x.x.x.x). Traffic will flow through the Firebox to the web server. Click the "Test" button in the AWS Console to start the Lambda function again. If the Lambda function is running while the Firebox is processing the traffic, packets will appear in the logs.

4.6. Limitations and Improvements

This demonstration is not a production-ready example of packet capture on AWS but proves that packet capture on AWS is possible (Figure 9). To create a robust solution, the architectural considerations outlined in this paper would drive a solution for packet capture solution that meets the needs of an organization. The solution would need to consider the cost of storage, scalability, monitoring, maintenance, and security of the solution itself.

Teri Radichel, teri@radicalsoftware.com

The Lambda function writes the logs to CloudWatch which has advantages and disadvantages. Logging to CloudWatch ensures that the security professionals implementing this solution or developers in the account cannot alter the data. Various tools can pull the data out of CloudWatch and transform it for further analysis. One downside is that the output is not in pcap format. Many tools analyze files in pcap format, and alternatively, the code could write pcap files to a secure S3 bucket.

Some organizations with limited downtime requirements may require a more scalable and redundant solution. A highly-available architecture needs at least two instances in each VPC, in case one host experiences issues. A larger size would handle more load, but a scalable architecture behind using auto-scaling groups and elastic load balancers would grow and shrink on demand.

A Lambda function only captures logs for a short time, which may be useful in the case of an incident for an organization that does not capture packets on a constant basis. For continuous packet capture, a scalable solution with EC2 instances configured to capture packets would be more appropriate. Large organizations with dedicated security teams may consider the transit VPC architecture. Smaller organizations may choose to initiate packet capture on an as-needed basis or outsource network log analysis to a third party managed security provider.

5. Conclusion

Although AWS customers cannot access span ports, options still exist for full packet capture on AWS. Packet capture on AWS depends on architectures that route all packets to hosts responsible for packet capture. A proxy or NAT architecture can capture traffic as it passes from one point in the network to another. Security appliances designed to receive and route traffic can both inspect the packets and offer packet capture functionality.

Packet capture architectures must take into consideration the need to secure the packet capture architecture using AWS constructs such as IAM roles, policies, network routing, subnets, and security groups. Creating an automated deployment for a packet

capture architecture will allow a security team to test the implementation in advance and quickly re-deploy in the case of system failure, disaster recovery or failover.

A transit VPC may be a desirable choice for customers with a complicated environment to decrease the number of connected accounts or VPCs. The transit VPC can also serve as a checkpoint to capture and inspect traffic as if flows from one network to another. A highly skilled security and networking team can maintain this critical point in the infrastructure and heavily monitor this traffic, in addition to the intra-VPC traffic data available via VPC Flow Logs.

The sample code shows how to implement automated deployment of a security appliance in a NAT configuration and related EC2 instances for packet capture and testing. This sample code proves that packet capture is possible on AWS. Though not intended to be a production ready code base, it should offer a starting point for teams that want to implement packet capture on AWS.

Teri Radichel, teri@radicalsoftware.com

6. References

Alexander, R. (2016, December 1). *AWS re:Invent 2016: From One to Many: Evolving VPC Design (ARC302)*. Retrieved from YouTube: https://www.youtube.com/watch?v=3Gv47NASmU4

Amazon Web Services. (2016, December). *AWS Storage Services Overview*. Retrieved from aws.amazon.com: https://d0.awsstatic.com/whitepapers/Storage/AWS%20Storage%20Services%20Whitepaper-v9.pdf

Amazon Web Services. (2017a). *AWS DIrect Connect*. Retrieved from aws.amazon.com: https://aws.amazon.com/directconnect/

Amazon Web Services. (2017b). *Amazon EC2 Pricing*. Retrieved from aws.amazon.com: https://aws.amazon.com/ec2/pricing/on-demand/

Amazon Web Services. (2017c). *Endpoints for Amazon S3.* Retrieved from aws.amazon.com: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-endpoints-s3.html

Amazon Web Services. (2017d). *Example 2: Bucket Owner Granting Cross-Account Bucket Permissions*. Retrieved from aws.amazon.com: http://docs.aws.amazon.com/AmazonS3/latest/dev/example-walkthroughs-managing-access-example2.html

Amazon Web Services. (2017e). *Flow Log Limitations.* Retrieved from aws.amazon.com: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html#flow-logs-limitations

Teri Radichel, teri@radicalsoftware.com

Amazon Web Services. (2017f). *How Do I Enable or Suspend Versioning for an S3 Bucket?* Retrieved from aws.amazon.com: http://docs.aws.amazon.com/AmazonS3/latest/user-guide/enable-versioning.html

Amazon Web Services. (2017g). *How to Restrict Amazon S3 Bucket Access to a Specific IAM Role.* Retrieved from aws.amazon.com: https://aws.amazon.com/blogs/security/how-to-restrict-amazon-s3-bucket-access-to-a-specific-iam-role/

Amazon Web Services. (2017h). *Invalid VPC Peering Connection Configurations*. Retrieved from aws.amazon.com: http://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/invalid-peering-configurations.html

Amazon Web Services. (2017i). *NAT Gateways*. Retrieved from aws.amazon.com: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html

Amazon Web Services. (2017j). *NAT Instances*. Retrieved from aws.amazon.com: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_NAT_Instance.html

Amazon Web Services. (2017k). *Overview of Security Processes.* Retrieved from aws.amazon.com: https://aws.amazon.com/whitepapers/overview-of-security-processes/

Amazon Web Services. (2017l). *Using Instance Profiles.* Retrieved from aws.amazon.com: http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2_instance-profiles.html

Teri Radichel, teri@radicalsoftware.com

Amazon Web Services. (2017m). *Using an HTTP Proxy*. Retrieved from aws.amazon.com: http://docs.aws.amazon.com/cli/latest/userguide/cli-http-proxy.html

Amazon Web Services. (2017o). *VPC Security Capabilities.* Retrieved from aws.amazon.com: https://aws.amazon.com/answers/networking/vpc-security-capabilities/

Barr, J. (2015, 03 24). *New - Cross-Region Replication for Amazon S3.* Retrieved from AWS Blog: https://aws.amazon.com/blogs/aws/new-cross-region-replication-for-amazon-s3/

Barr, J. (2016, August 11). *AWS Solution – Transit VPC*. Retrieved from AWS Blog: https://aws.amazon.com/blogs/aws/aws-solution-transit-vpc/

Bejtlich, R. (2005). *The Tao of Network Security Monitoring: Beyond Intrusion Detection.* New York, NY: Addison-Wesley.

Chan, C.-S. (2012, December 17). *Complexity the worst enemy of security*. Retrieved from Computerworld: https://www.schneier.com/news/archives/2012/12/complexity_the_worst.html

Hall, S. (2009, 10 02). *Cyber Security Awareness Month - Day 2 - Port 0*. Retrieved from InfoSec Handlers Diary Blog: https://isc.sans.edu/diary/Cyber+Security+Awareness+Month+-+Day+2+-+Port+0/7216

Marks, H. (2014, July 3). *Code Spaces: A Lesson In Cloud Backup*. Retrieved from Network Computing: http://www.networkcomputing.com/cloud-infrastructure/code-spaces-lesson-cloud-backup/314805651

Teri Radichel, teri@radicalsoftware.com

Matthews, N. (2016, March 29th). *Amazon VPC for On-Premises Network Engineers, Part One.* Retrieved from aws.amazon.com: https://aws.amazon.com/blogs/apn/amazon-vpc-for-on-premises-network-engineers-part-one/

Mueller, S. (2016, November 30). *Another Day, Another Billion Packets.* Retrieved from YouTube: https://www.youtube.com/watch?v=St3SE4LWhKo

OWASP. (2017, May 31). *Application Threat Modeling*. Retrieved from OWASP: https://www.owasp.org/index.php/Application_Threat_Modeling#Trust_Levels

Poritskiy, A. (2017, July 11). *Measuring the impact of tcpdump on Very Busy Hosts*. Retrieved from Percona Database Performance Blog: https://www.percona.com/blog/2015/04/10/measuring-impact-tcpdump-busy-hosts/

Purcell, T. (2013, February 27). *Custom Full Packet Capture System.* Retrieved from SANS Reading room: https://www.sans.org/reading-room/whitepapers/logging/custom-full-packet-capture-system-34177

Radichel, T. (2016, 3 12). *Sans Institute Reading Room.* Retrieved from Balancing Security and Innovation With Event Driven Automation: https://www.sans.org/reading-room/whitepapers/incident/balancing-security-innovation-event-driven-automation-36837

Radichel, T. (2017, July 03). *Secplicity.* Retrieved from https://www.secplicity.org/2017/07/03/automate-deployment-watchguard-firebox-cloud-aws/: https://www.secplicity.org/2017/07/03/automate-deployment-watchguard-firebox-cloud-aws/

Teri Radichel, teri@radicalsoftware.com

SANS Institute. (2016, November). Intrustion Detection In Depth. *SANS Networking* . Las Vegas, Nevada, United States: SANS Institute.

Varia, J. (2015, October 27). *High Availability for Amazon VPC NAT Instances: An Example*. Retrieved from aws.amazon.com: https://aws.amazon.com/articles/2781451301784570

WatchGuard Technologies. (2017, June 7). *Watchguard.com.* Retrieved from Fireware Command Line Interface Reference v11.12.4: https://www.watchguard.com/help/docs/fireware/11/en-US/CLI/Fireware_CLI-Reference_(en-US)_v11-12-4.pdf

7. Appendix – AWS Cost and Service Selection

The cost of a packet capture solution on AWS will vary based on selected services and architecture. The size of packet capture logs by a host will differ based on the applications running on a host and the number and size of requests and responses. Data storage choices depend on the amount of the data stored, length of time, and retrieval time requirements. Data compression and encryption alter the size of data and will affect transfer and storage costs. The selected data storage service will depend on who can see the data and how they will query the data which drives data structure and security requirements.

To calculate charges for the example in this paper for a would be of limited value. It would prove that Amazon charges the advertised price for services but would not be relevant to any other given scenario. The proof-of-concept in this paper had limitations as noted, so showing the cost would not help for a production solution. AWS reduces the price of services on a regular basis so any pricing listed here may be out of date soon. It is more helpful to explain the factors that customers should consider when architecting a solution on AWS. A detailed explanation of AWS pricing, selecting the correct services and calculating the cost of a packet capture architecture would be a paper unto itself, so this appendix does not aim to explain this in detail, but rather the concepts and tools available for further analysis.

7.1. Service Pricing

AWS Services each have distinct pricing models. Service pricing factors include time, storage, data transfer, some other factors, or a combination. Services may have different prices in different regions. Data transferred into AWS is free. Data transfer to another AWS account incurs a charge. Data sent out of AWS incurs an even higher fee. Some services have different compute, memory, and bandwidth costs. Some services offer the ability to pay up front for a lower cost or bid on services for a lower price if the service is available.

Look at the price page for a service to understand all the pricing options for that service. The easiest way to find the price of a service is to search for the service name

plus "pricing" in Google. For example, search for "Lambda Pricing" to find this page: https://aws.amazon.com/lambda/pricing/. AWS Lambda charges based on the number of requests and duration of requests:

Requests

You are charged for the total number of requests across all your functions. Lambda counts a request each time it starts executing in response to an event notification or invoke call, including test invokes from the console.

- First 1 million requests per month are free
- $0.20 per 1 million requests thereafter ($0.0000002 per request)

Duration

Duration is calculated from the time your code begins executing until it returns or otherwise terminates, rounded up to the nearest 100ms. The price depends on the amount of memory you allocate to your function. You are charged $0.00001667 for every GB-second used.

On some pricing pages, an equation showing the calculation of the cost of that service included such as this example below found on the Lambda pricing page:

Pricing Example 1

If you allocated 512MB of memory to your function, executed it 3 million times in one month, and it ran for 1 second each time, your charges would be calculated as follows:

Monthly compute charges

The monthly compute price is $0.00001667 per GB-s and the free tier provides 400,000 GB-s.

Total compute (seconds) = 3M * (1s) = 3,000,000 seconds

Total compute (GB-s) = 3,000,000 * 512MB/1024 = 1,500,000 GB-s

Total compute – Free tier compute = Monthly billable compute GB- s

1,500,000 GB-s – 400,000 free tier GB-s = 1,100,000 GB-s

Monthly compute charges = 1,100,000 * $0.00001667 = $18.34

Monthly request charges

The monthly request price is $0.20 per 1 million requests and the free tier provides 1M requests per month.

Total requests – Free tier requests = Monthly billable requests

3M requests – 1M free tier requests = 2M Monthly billable requests

Monthly request charges = 2M * $0.2/M = $0.40

Total monthly charges

Total charges = Compute charges + Request charges = $18.34 + $0.40 = $18.74 per month

7.2. Calculating Costs

To calculate the cost first understand the formula used to bill for the service and the necessary inputs. For each input, figure out the value based on your current environment or estimated future usage. The values will vary based on the application and use. For example, a web server that serves up a page of text every hour user for one would generate much less traffic than a web server that serves up many large images every minute for 1000's of users. The large images will need more storage or a larger EC2 instance size and will incur more data transfer charges.

One problem translating solutions from a data center environment to a cloud solution is a misalignment of the requirements, architecture, and the service billing mechanism. For example, a business owner may want a desired amount of time to store data for a fixed cost when the service billing mechanism is data storage. The system must limit data stored by stopping when it reaches a maximum amount or periodically truncate the data to stay within a defined budget for a fixed time.

AWS offers a calculator which some customers may find helpful: https://calculator.s3.amazonaws.com/index.html. Other companies may find a spreadsheet and the pricing formulas to work just as well, especially if the price calculator does not include the needed services. AWS account representatives, AWS Support, and for large customers, AWS product teams can all help with cost calculations and choice of the correct services.

An AWS representative said at an AWS conference that if a system that costs too much, the solution is using the wrong services. Mismanagement of services can also drive up the cost. Monitoring systems and dynamically turning off systems not in use, allocating appropriate service levels and correctly leveraging AWS service options can all drive down cost.

7.3. Beta Test for Cost

After speaking with several AWS Solution Architects and a customer account representative, the overwhelming recommendation is to get a system running and beta test the architecture. A beta test analysis will help truly understand the costs of a system because even when trying to take every factor into account running the system will bring unforeseen factors to light. To capture the charges for each user or group of users in a

multi-user environment may require the use of AWS Tags which can label resources for later reporting, customized logging and monitoring.

7.4. Reducing Costs

After beta testing the system, the system can be optimized to limit costs the same way it would be optimized to improve performance. Find the item that is incurring the most fees and strategize on methods for reducing the charges for that part of the system. Dynamically change or cap system usage, or limit the inputs into the above pricing formula that caused the increased billing.

The cloud is a programmatic platform which creates great flexibility for tightly controlled and innovative solutions to meet the desired budget. For example, the AWS DynamoDB service bills customers based on allocated bandwidth. At the 2016 annual AWS conference, a solution architect specializing in SAAS (software as a service) architectures showed automated increase and decrease of requested bandwidth based on system load. Other companies have shown dynamic use of spot instances which allow customers to bid on unused compute capacity as explained in the AWS Spot Instance page: https://aws.amazon.com/ec2/spot/

Many services allow capping usage. For example, the Auto-Scaling Group service which increases and decreases EC2 Instances based on demand has a configuration option for a maximum number of hosts. When no cap is available, a customer can programmatically end service usage triggered by an event in the system such as a log line or value in a database. Automated truncation of data can help keep costs below the desired point for storage services. All services on AWS offer an API (application programming interface) to interact with the service.

Companies need to take maintenance expenses into account when considering service options on AWS. For example, some customers may think that running a database on an EC2 instance is less expensive than running an RDS (Relational Database Service) instance. However, the RDS instance comes with many services that typically require the services of an expensive DBA (Database Administrator). Automated processes reduce the potential for human error. High availability and replication may be easier. A checkbox turns on database encryption. High-value employees can spend time on work that is more

valuable to the business instead of managing database backups. For complex services that include many security features, the cost building out and keeping an environment secure may be lower when leveraging AWS service features. For example, CloudFormation offers some security benefits that would be expensive for a company to replicate and maintain in-house as explained in this article: https://www.secplicity.org/2017/05/04/cloudformation-benefits-secure-aws-deployments/

Innovation is one of the primary reasons companies use AWS. Although a company could spend a lot of time rebuilding something AWS offers for a fee, many customers pay for these services instead to free their time up to create new innovative products and services. For example, instead of taking time for not only set up but the on-going maintenance to run ElastiSearch or Hadoop clusters, a company may opt to pay AWS for those automated functions and spend time getting a product quickly to market before the competition. Creating new features not offered by anyone else that drives more revenue may offset any costs associated with buy vs. build decisions.

In some cases, AWS can offer these services more cost-effectively due to automation and economies of scale. Paying Amazon more for a service may cost more on the cloud bill but drive down internal expenses. AWS support can be an extension of internal staff to help solve problems more quickly. If implemented correctly, these services can also offer improved security through improved automation, monitoring, and segregation of duties between AWS and company staff. Improved security will help with incident response and potentially help avoid the cost of a data breach.

7.5. Service Level Agreement (SLA)

Each Amazon Web Service has a different SLA (service level agreements) or in some cases no SLA. Some service level agreements only apply when a customer architects the system for high availability. For example, the EC2 SLA states that a region is “available” is EC2 instances are operational if any availability zone. An availability zone is the equivalent of a data center in a region. That means a customer must run the EC2 instances for an application in multiple availability zones to receive the benefit of the service level agreement. Customers should review the service level agreement for each service if uptime requirements exist.

Teri Radichel, teri@radicalsoftware.com

7.6. Security Factors

Some services are HIPAA compliant while others are not. Some services are not available in GovCloud, the authorized AWS environment for US government agencies. Some services run inside a VPC, so the data is completely private. Some run inside the AWS network but external to a VPC. Some services need a connection to the Internet to function. Some services offer encryption and others do not. Encryption functionality of some services may not be applicable for specific applications. Understanding all these factors – and the business requirements -- will affect the choice of service when the solution must meet certain security standards or comply with government or industry laws and regulations.

7.7. Summary

In summary, choice of AWS services for a packet capture or any AWS architecture requires understanding the formulas the business requirements, the specific features for a service, and the pricing formula. The defined business requirements drive choice of service. The customer must know the expected or current inputs to the pricing formula for a service to calculate the cost. If the expense is too high, consider other services or service options. Customers can also use service cap and pricing options, programmatic limits or deletion of inputs, and dynamic allocation of resources to reduce costs.

Teri Radichel, teri@radicalsoftware.com