# Case Study: Seagate's Amazon AWS Cloud Security


Rodney Beede
October 2016

## Introduction

- Rodney Beede
  - IT Cloud Security Architect & Engineer
  - Seagate Technology
- M.S. in Computer Science
  - University of Colorado at Boulder
  - “A Framework for Benevolent Computer Worms” 2012
- Doing computer security since 2001
  - Primary interests are web and cloud security
  - Authored chapter “Object Storage” in the OpenStack Security Guide
  - Discovered CVE-2013-3627: McAfee Agent v4.6 Denial of Service
  - Discovered data disclosure vuln in Google Spreadsheets
- Tech blog
  - https://www.rodneybeede.com/
  - The views expressed in this blog are my personal view and have not been reviewed or approved by Seagate.

# Synopsis

**Case Study: Seagate's Amazon AWS Cloud Security**

Overview of the architecture developed by Seagate for use in its IT AWS cloud deployments. Coverage includes use of next generation firewalls and cloud network security controls to secure internet and internal traffic.

A technical dive into how the security team at Seagate enabled business flexibility for rapid deployment while balancing security requirements by leveraging Amazon cloud security technologies will be explored.

The audience will also learn about the security tradeoffs, compensating auditing controls, and limitations of AWS in regards to cloud network security and user management.

Additionally, a consolidated checklist based on industry whitepapers and cloud security leaders, as used by Seagate, for evaluation of cloud security readiness will be provided.

# Agenda

- 7 Key Questions
- Challenges for Enterprise Organizations
- Balancing Rapid Business and Security
- The Architecture
- Security Limitations, Tradeoffs, and Options
- IaaS Checklist



Goal of this presentation is to give a small introduction into SecDevOps. Further deep dives into topical areas will be made based on audience participation and feedback.

# Asset Risk Evaluation


Image from openclipart.org

The Seven Key Questions:

1. How would we be harmed if the asset became public and widely distributed?
2. How would we be harmed if an employee of our cloud provider accessed the asset?
3. How would we be harmed if the process or function was manipulated by an outsider?
4. How would we be harmed if the process or function failed to provide expected results?
5. How would we be harmed if the information/data was unexpectedly changed?
6. How would we be harmed if the asset was unavailable for a period of time?
7. How does this affect our compliance obligations?

Taken from - "**Cloud Risk Thoughts: Deciding What, When, and How to Move to the Cloud**" - Rich Mogull of Securosis
https://securosis.com/blog/cloud-risk-thoughts-deciding-what-when-and-how-to-move-to-the-cloud/

Posted December 1, 2009. Accessed August 16, 2016.



I always share this for any of my cloud security presentations. This is what the business must ask themselves before you can understand their security needs. Graphic from openclipart.org.

## Challenges for Enterprise Organizations

1. Still want to login to everything
    a. Not scalable for thousands of cloud systems
2. Still want to do traditional patching
3. Business applications not designed for HA or cloud
    a. More sensitive to (virtual) hardware downtime
    b. Not accustomed to throw it away and replace in minutes
    c. Requires more quality control before deployment (not logging in to fix after)
4. Lack of configuration automation
5. Paying for network bandwidth or storage as you go model is different
    a. Having a 200GB disk with 80% unused space still costs full price



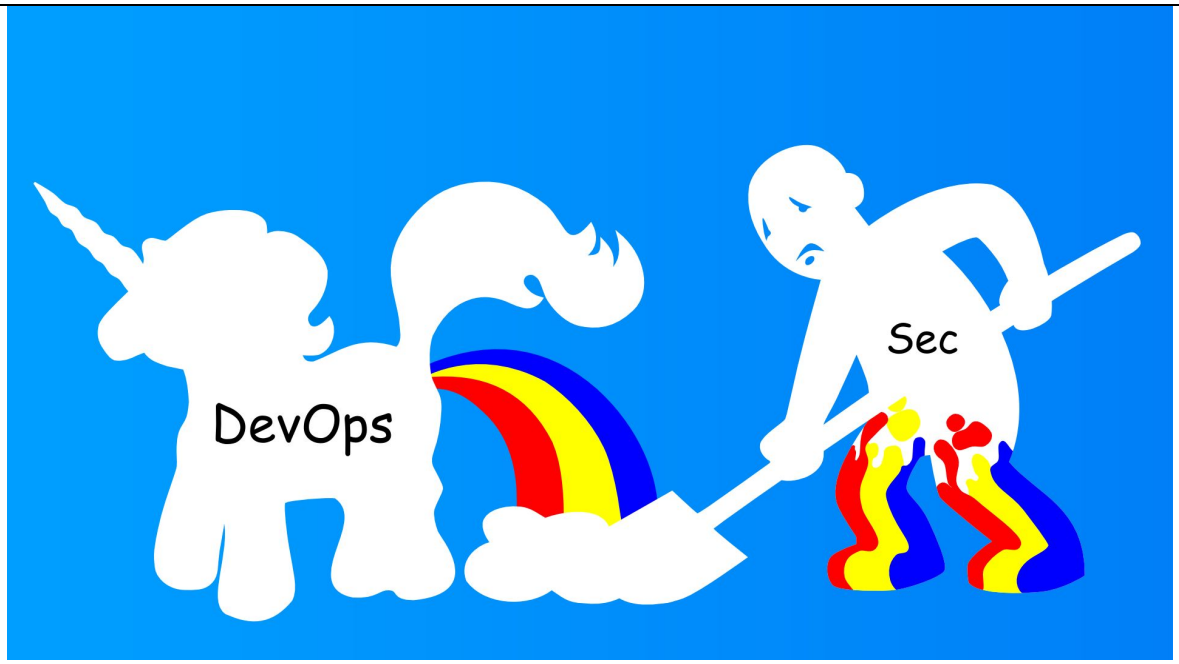
Disk usage example: In VMware on-premise environment you can thin provision and share unused disk space but in cloud you still pay for that space monthly

## Balancing Rapid Business and Security

- Shiny new cloud APIs and tools
  - Look at all the stuff we can just turn on with a click
- Automation of business processes
  - OS or App config (Puppet, Chef, Ansible)
  - Business forces itself out of manual deployment (short-term)
  - Business invests time and engineering into automation
    - **Enables faster disaster recovery**
    - **Environment change control**
- Business Goal
  - Cut down deploy time from weeks to days
  - “Oh no IT takes forever” becomes “IT got my site up fast!”
- **But where does security fit in?**
  - Viewed as another last-minute delay to my project
  - Why can’t you guys get with the new DevOps model?



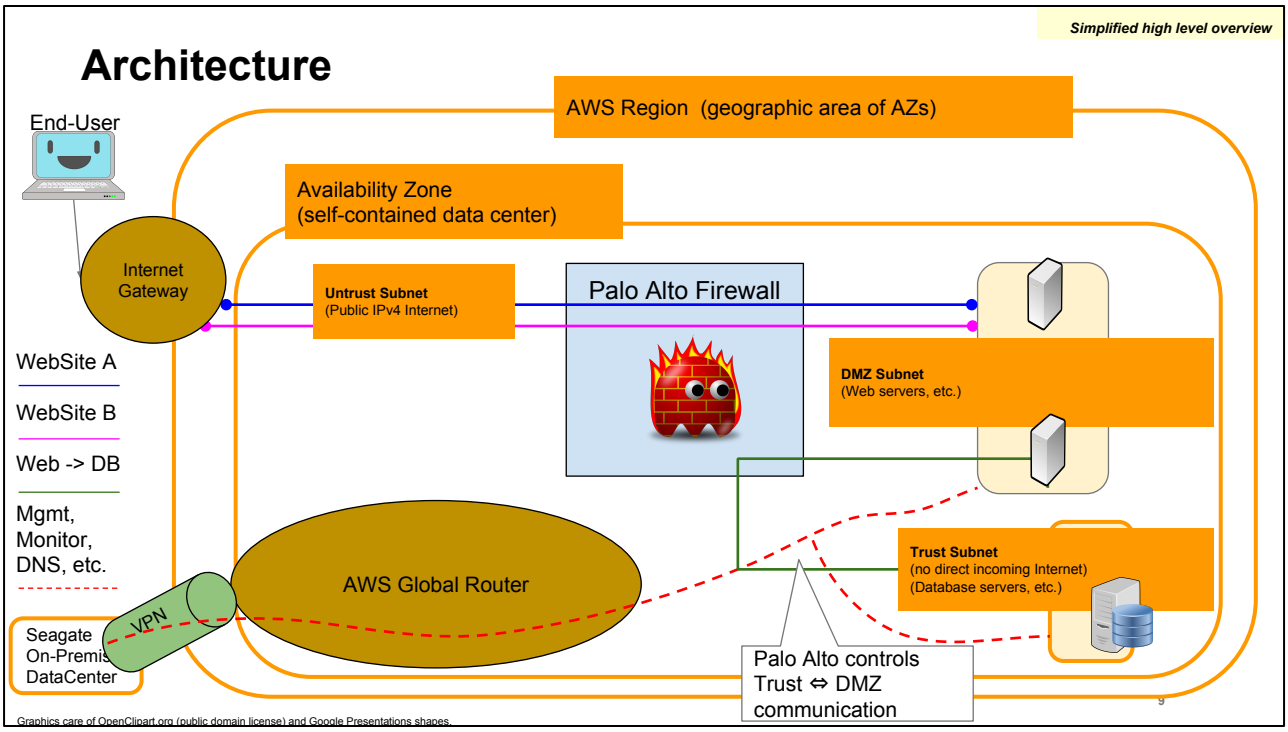
Short-term meaning the manual method was a short-term way or gain but changing to automation requires long-term investment to mature.

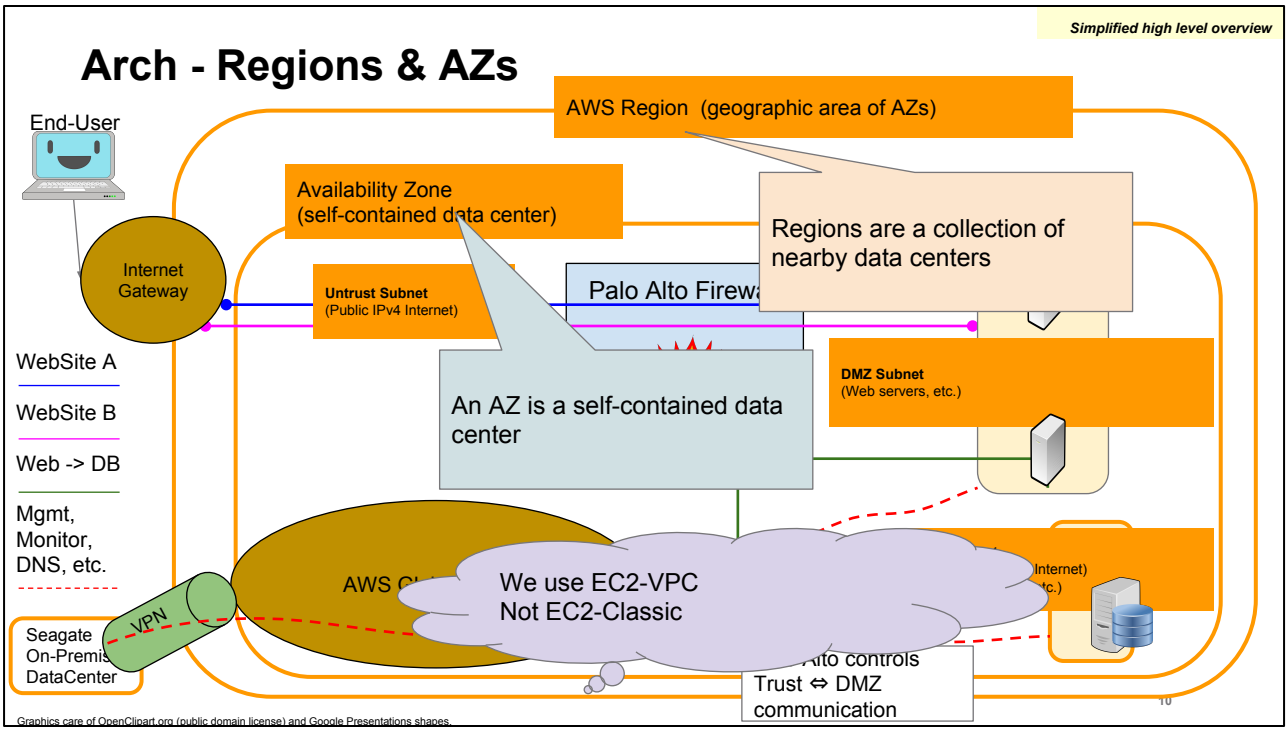


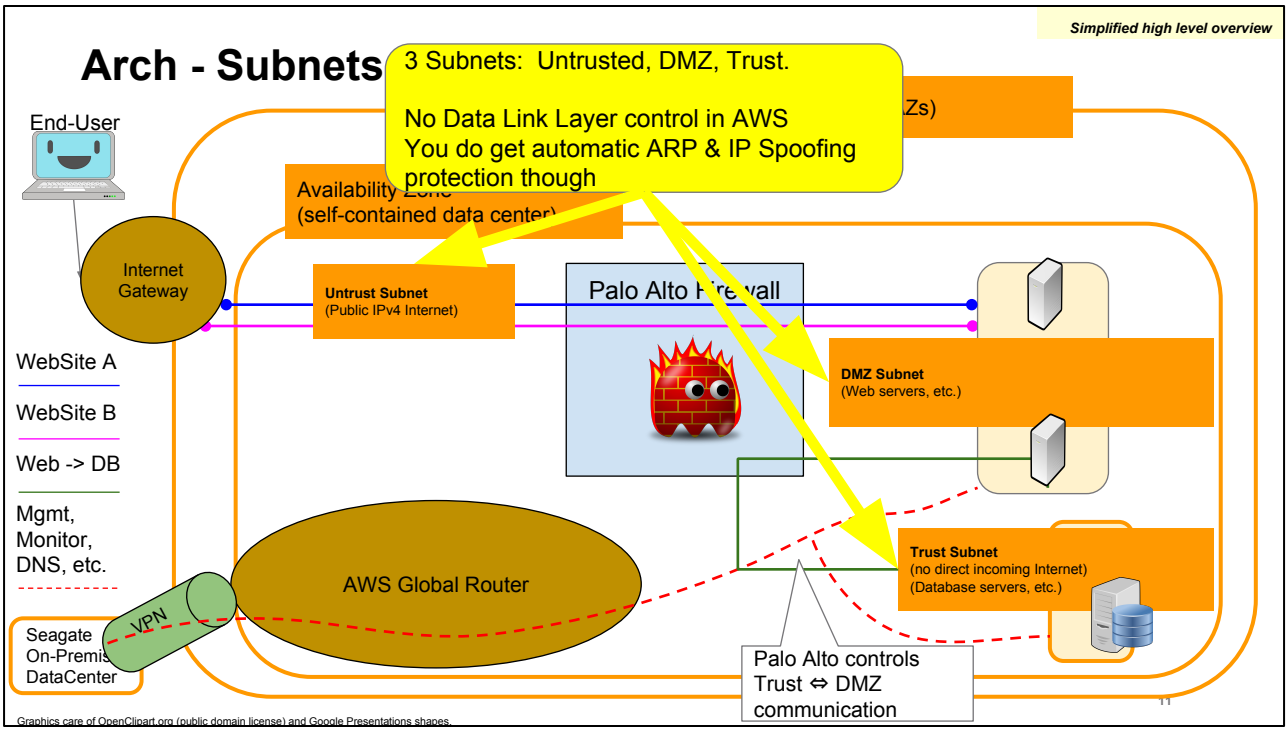

Credit: Pete Cheslock at #DevOpsDaysAustin Twitter of his parody. https://twitter.com/petecheslock/status/595617204273618944; May 2015.

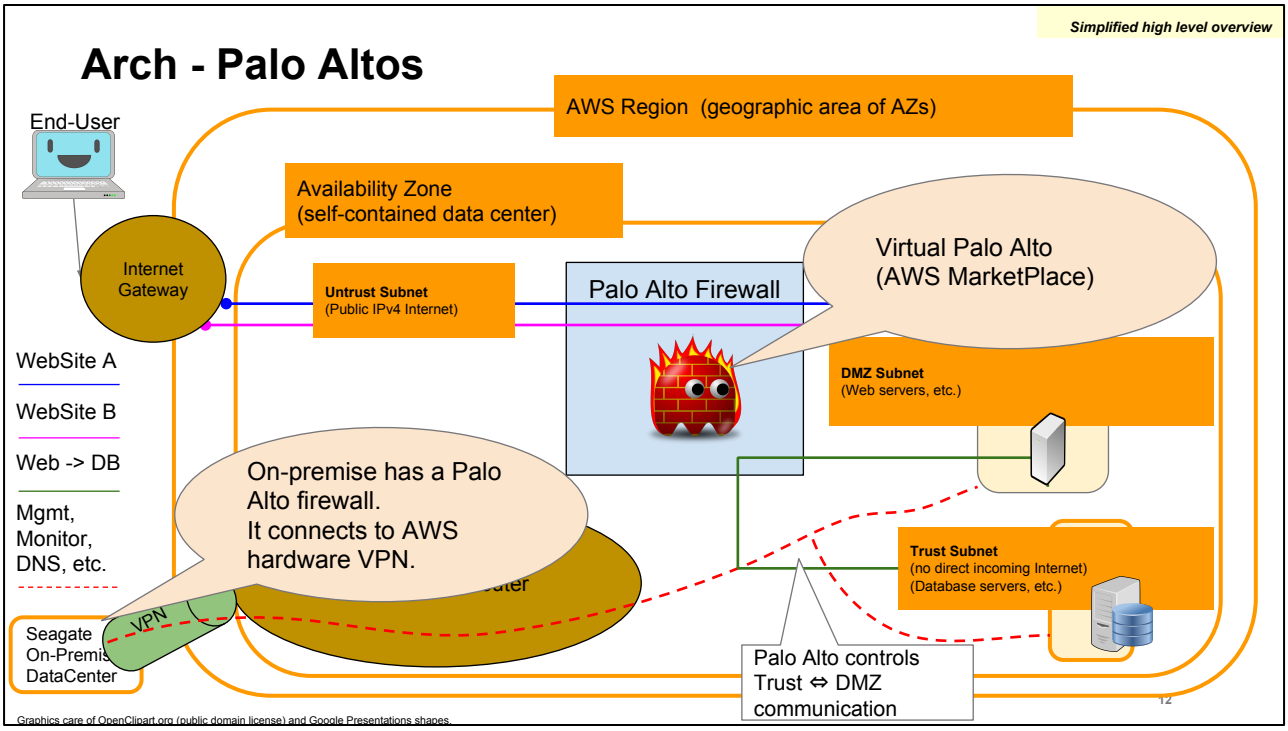


Doesn’t have to be like this of course.

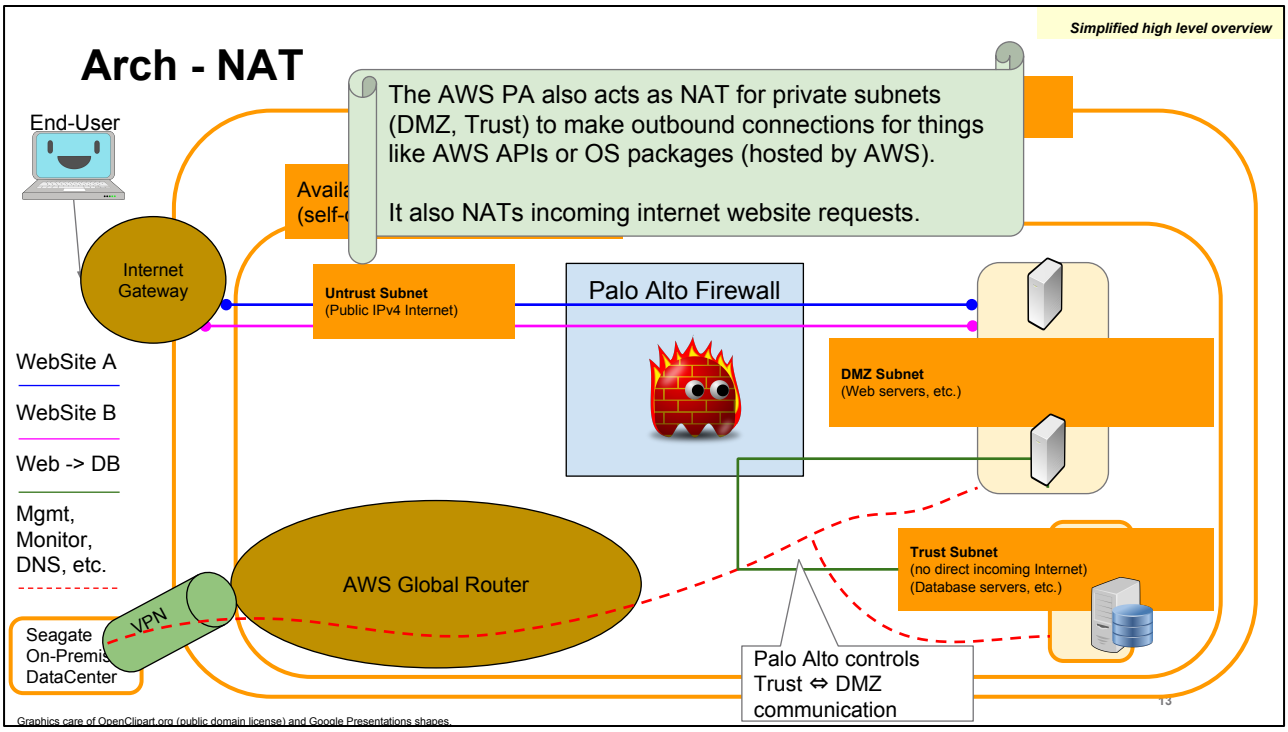

Simplified high level overview
Architecture
End-User
AWS Region (geographic area of AZs)
Availability Zone
(self-contained data center)
Internet
Gateway
Untrust Subnet
(Public IPv4 Internet)
Palo Alto Firewall
DMZ Subnet
(Web servers, etc.)
WebSite A
WebSite B
Web -> DB
Mgmt,
Monitor,
DNS, etc.
AWS Global Router
Trust Subnet
(no direct incoming Internet)
(Database servers, etc.)
VPN
Seagate
On-Premis
DataCenter
Palo Alto controls
Trust ⇔ DMZ
communication
Graphics care of OpenClipart.org (public domain license) and Google Presentations shapes.
9


Simplified high level overview
Arch - Regions & AZs
End-User
AWS Region (geographic area of AZs)
Availability Zone
(self-contained data center)
Regions are a collection of nearby data centers
Internet Gateway
Untrust Subnet
(Public IPv4 Internet)
Palo Alto Firewa
WebSite A
WebSite B
Web -> DB
Mgmt, Monitor, DNS, etc.
An AZ is a self-contained data center
DMZ Subnet
(Web servers, etc.)
AWS Cl
We use EC2-VPC
Not EC2-Classic
Internet)
tc.)
VPN
Seagate On-Premis DataCenter
Alto controls
Trust ⇔ DMZ
communication
Graphics care of OpenClipart.org (public domain license) and Google Presentations shapes.
10

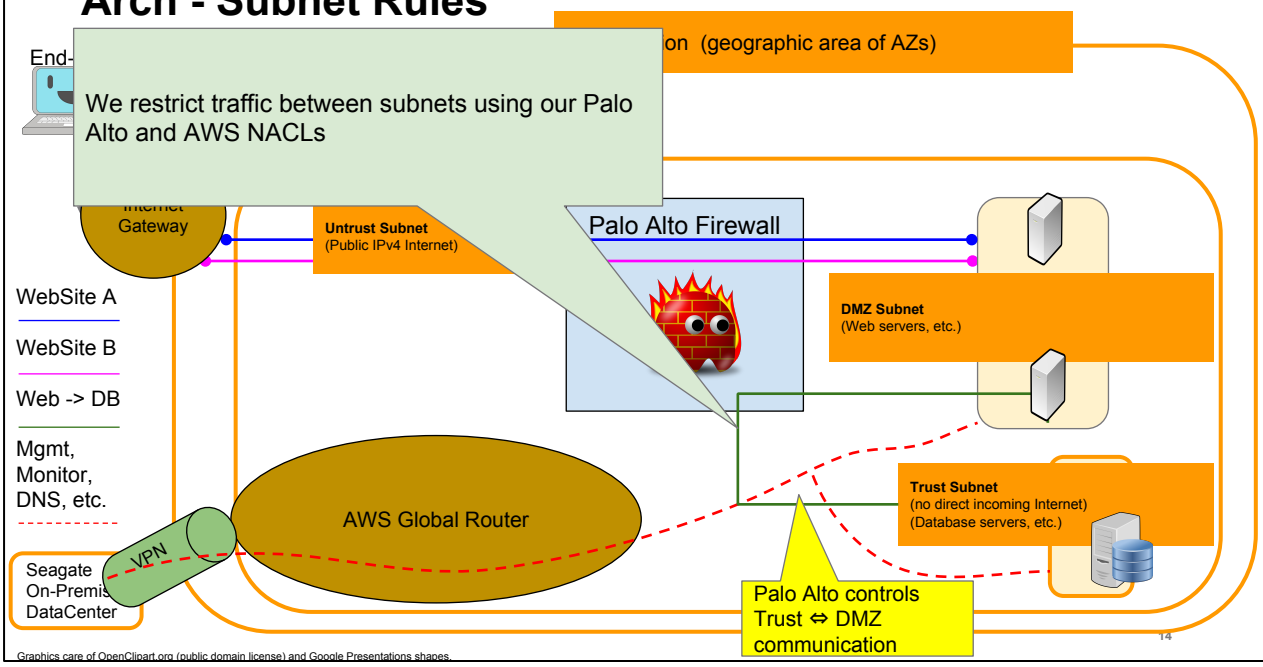

Simplified high level overview
Arch - Subnets
3 Subnets: Untrusted, DMZ, Trust.
No Data Link Layer control in AWS
You do get automatic ARP & IP Spoofing protection though
End-User
Internet Gateway
Availability Zone
(self-contained data center)
Untrust Subnet
(Public IPv4 Internet)
Palo Alto Firewall
DMZ Subnet
(Web servers, etc.)
Trust Subnet
(no direct incoming Internet)
(Database servers, etc.)
WebSite A
WebSite B
Web -> DB
Mgmt, Monitor, DNS, etc.
AWS Global Router
VPN
Seagate On-Premis DataCenter
Palo Alto controls Trust ⇔ DMZ communication
Graphics care of OpenClipart.org (public domain license) and Google Presentations shapes.
11



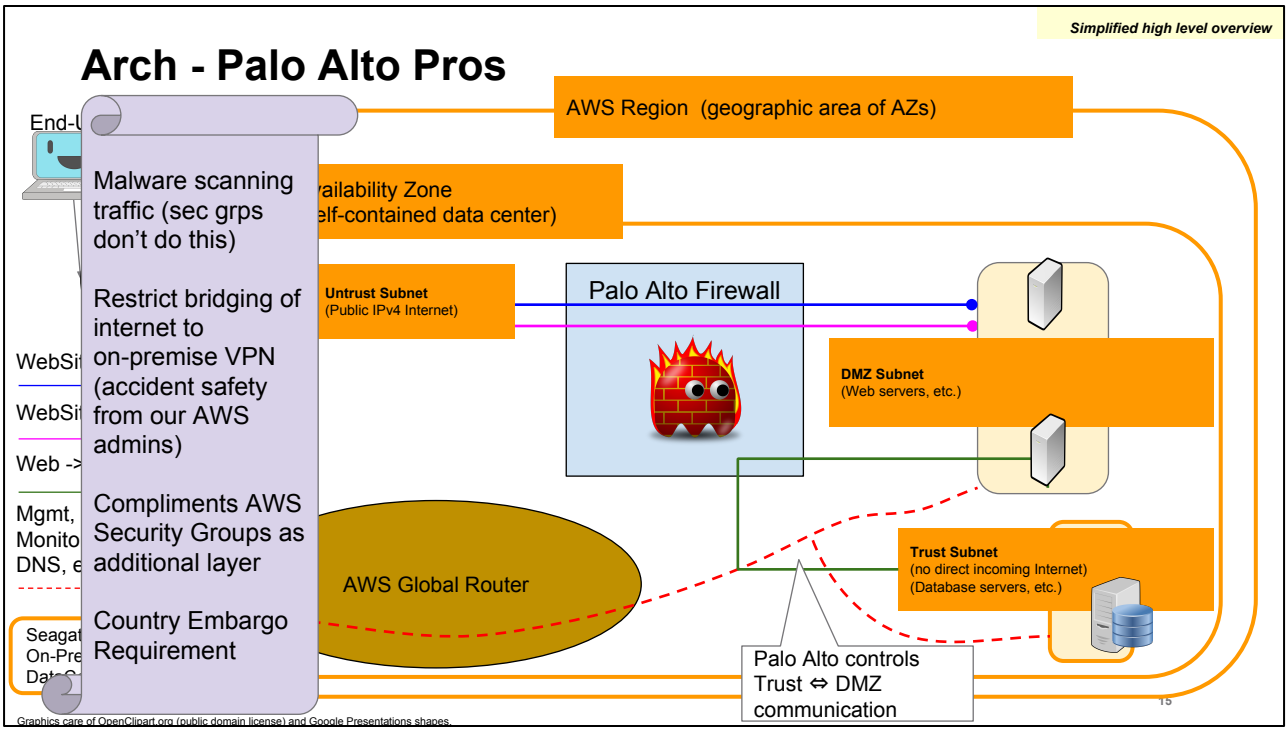

Only needed 1 firewall on one side of the VPN.


Simplified high level overview
Arch - NAT
The AWS PA also acts as NAT for private subnets (DMZ, Trust) to make outbound connections for things like AWS APIs or OS packages (hosted by AWS).
It also NATs incoming internet website requests.
End-User
Internet Gateway
Untrust Subnet
(Public IPv4 Internet)
Palo Alto Firewall
DMZ Subnet
(Web servers, etc.)
WebSite A
WebSite B
Web -> DB
Mgmt, Monitor, DNS, etc.
AWS Global Router
Trust Subnet
(no direct incoming Internet)
(Database servers, etc.)
VPN
Seagate On-Premis DataCenter
Palo Alto controls Trust ⇔ DMZ communication
Graphics care of OpenClipart.org (public domain license) and Google Presentations shapes.
13

# Arch - Subnet Rules




Graphics care of OpenClipart.org (public domain license) and Google Presentations shapes.




PA also can detect things like Heartbleed and block them. AWS secgrps don't do that. Example: AWS ELB were vulnerable so you had to wait until AWS patched all their ELB before Heartbleed was stopped.

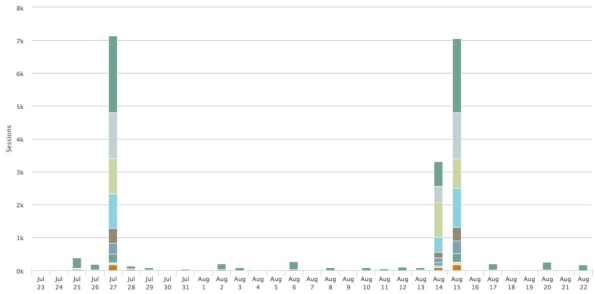
# Palo Alto - Advanced Threat Prevention

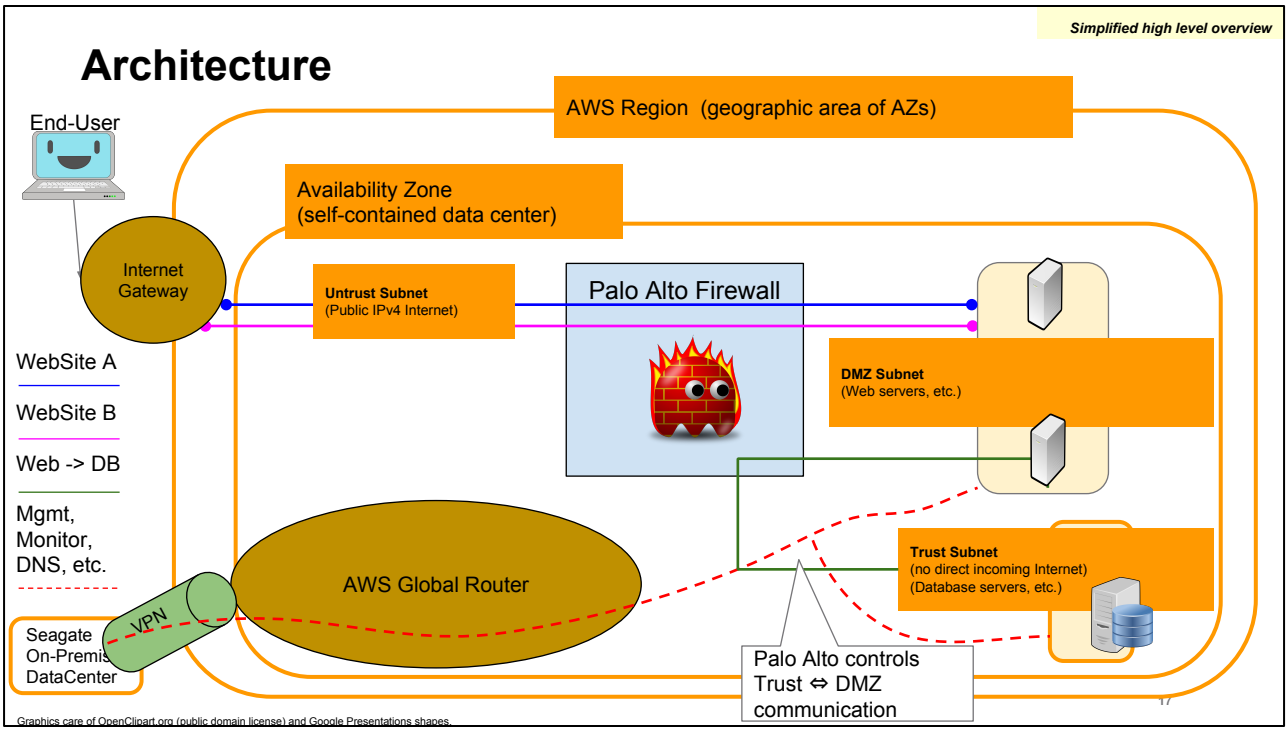
Sample of threat attempts seen in Seagate AWS blocked by Palo Alto:

- HTTP Directory Traversal
- HTTP /etc/passwd attempt
- Bash Remote Code Execution
- Apache Wicket XSS
- WordPress Login Brute Force
- PHP CGI Query String Parameter Handling Code Injection
- Joomla Remote Code Execution

Metrics over 30 days:

Top spikes just over 7,000




Simplified high level overview
Architecture
End-User
AWS Region (geographic area of AZs)
Availability Zone
(self-contained data center)
Internet
Gateway
Untrust Subnet
(Public IPv4 Internet)
Palo Alto Firewall
DMZ Subnet
(Web servers, etc.)
WebSite A
WebSite B
Web -> DB
Mgmt,
Monitor,
DNS, etc.
AWS Global Router
Trust Subnet
(no direct incoming Internet)
(Database servers, etc.)
VPN
Seagate
On-Premis
DataCenter
Palo Alto controls
Trust ⇔ DMZ
communication
Graphics care of OpenClipart.org (public domain license) and Google Presentations shapes.
17

# AWS NACL

- 6 Subnets
  - 2 AZs
  - 3 Subnets each
- Inbound/Outbound NACL rules
  - Typically same set
- Rules only apply:
  - Subnet-to-Subnet
  - Not internal subnet traffic
- NACL is stateless
  - Rule order, First match
- Security Groups applied too
  - NACL Deny ⇒ Denied *(Despite Sec Grp Rules)*

| Name | Subnet ID | VPC | CIDR |
|---|---|---|---|
| DMZ Subnet Primary | subnet-00000001 | vpc-abcdefgh (203.0.113.0/20) \| ... | 203.0.115.0/24 |
| DMZ Subnet Secondary | subnet-00000002 | vpc-abcdefgh (203.0.113.0/20) \| ... | 203.0.121.0/24 |
| Trust Subnet Primary | subnet-00000003 | vpc-abcdefgh (203.0.113.0/20) \| ... | 203.0.116.0/23 |
| Trust Subnet Secondary | subnet-00000004 | vpc-abcdefgh (203.0.113.0/20) \| ... | 203.0.122.0/23 |
| Untrust Subnet Primary | subnet-00000005 | vpc-abcdefgh (203.0.113.0/20) \| ... | 203.0.118.0/24 |
| Untrust Subnet Secondary | subnet-00000006 | vpc-abcdefgh (203.0.113.0/20) \| ... | 203.0.124.0/24 |

**subnet-00000001 (203.0.115.0/24) | DMZ Subnet Primary**

Summary | Route Table | **Network ACL** | Flow Logs | Tags

Edit

**Network ACL:** acl-fedbca09 | DMZ

**Inbound:**

| Rule # | Type | Protocol | Port Range / ICMP Type | Source | Allow / Deny |
|---|---|---|---|---|---|
| 50 | ALL Traffic | ALL | ALL | 203.0.115.0/24 | ALLOW |
| 51 | ALL Traffic | ALL | ALL | 203.0.121.0/24 | ALLOW |
| 100 | ALL Traffic | ALL | ALL | 203.0.113.0/20 | DENY |
| 500 | ALL Traffic | ALL | ALL | 198.51.100.0/24 | ALLOW |
| 600 | ALL Traffic | ALL | ALL | 192.168.0.0/16 | ALLOW |
| 1000 | ALL Traffic | ALL | ALL | 0.0.0.0/0 | ALLOW |
| * | ALL Traffic | ALL | ALL | 0.0.0.0/0 | DENY |

Enforcement of Palo Alto by using AWS NACLs. Avoids accidentally putting system on public Internet due to Security Group misconfiguration. Layers of defense.

# NACL Detail

| Rule # | ort Range / ICMP Type | Source | Allow / Deny |
|---|---|---|---|
| 50 | | 203.0.115.0/24 | ALLOW |
| 51 | | 203.0.121.0/24 | ALLOW |
| 100 | | 203.0.113.0/20 | DENY |
| 500 | | 198.51.100.0/24 | ALLOW |
| 600 | | 192.168.0.0/16 | ALLOW |
| 1000 | | 0.0.0.0/0 | ALLOW |
| * | | 0.0.0.0/0 | DENY |

Rule allows subnet itself. Why? I thought internal subnet traffic wasn't affected by NACL?

Palo Alto NAT requires this specific ALLOW.

This blocks entire VPC range ⇒ Block subnet-to-subnet routing.

Forces use of Palo Alto to cross subnets.

Incoming VPN traffic allowed

Makes above (VPN) rule redundant. VPN rule in place incase someone deletes this.

Hard-coded by AWS.
Always exists.
Always last.

# Security Groups *(2 examples)*

Security Group: sg-12345678 (Default - Nothing In or Out)

Description | **Inbound** | Outbound | Tags

Edit

| Type | Protocol | Port Range | Source |
|---|---|---|---|
| | | *This security group has no rules* | |

My favorite. If Seagate AWS admin fails to pick the right group it defaults to allowing nothing in or out.

Makes the admin think before they click.

Security Group: sg-abcdefg09

Description | **Inbound** | Outbound | Tags

***"Common Services"***

Edit

| Type | Protocol | Port Range | Source |
|---|---|---|---|
| HTTP | TCP | 80 | 0.0.0.0/0 |
| SSH | TCP | 22 | sg-c001deed (Lambda Security Group) |
| SSH | TCP | 22 | 198.51.100.0/24 |
| RDP | TCP | 3389 | 198.51.100.0/24 |
| HTTPS | TCP | 443 | 0.0.0.0/0 |

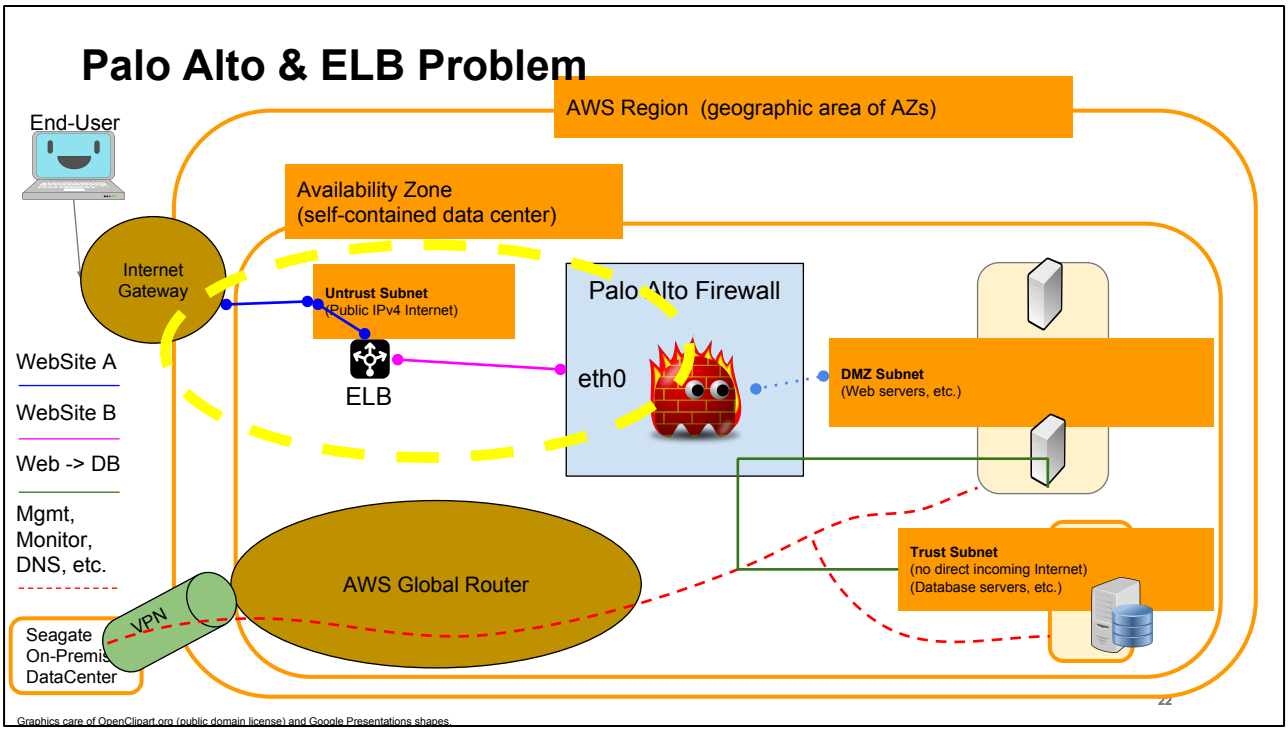
Neat grouping trick. Instead of by IP just add an instance to this other security group. No need to allow entire subnet. You protect east-west traffic now.

Remember that security group rules are additive allows. Anything not specifically allowed is implicitly denied. NACLs apply first and may deny despite ALLOW in security group.

# Palo Alto Limitations

- Using Elastic Load Balancer for internet traffic
  - ELB offers low cost geo-location load balancing
    - ***(same region, multiple AZs)***
  - Not supported in PA 7.0
    - **Seagate had to use NAT so PA had multiple EIPs for each site**
    - **Largest instance could handle 240 IPs**
    - **Not an easy to scale solution**
    - **Initial recommendation was purchase marketplace load balancer product**
  - Support added in 7.1
    - **Flips mgmt interface from eth0 to eth1**
    - **Required redeployment to achieve (AZ downtime)**
  - For internal (private subnet) ELB worked with PA just fine
- AWS Routing Tables feature limitation
  - You can't add a more specific route
    - **VPC has 203.0.113.0/24 ⇒ local**
    - **You can't add 203.0.113.0/28 ⇒ PaloAltoEth3 to the route table**

- *Following slides show examples*




AWS ELB pool members are defined by **instance-id** not IP. So everything always goes to eth0 on the primary IP only. [hold-over from EC2-Classic where your "private-ip" was never reserved but also changed if you stop/start the instance]. Palo Alto 7.0 only used eth0 for mgmt and not data pane. Version 7.1 added toggle switch to support this so PA is a pool member and can forward the traffic on.

# AWS Routing Table Limitation

**rtb-711fasdf | DMZ Primary**

Summary | **Routes** | Subnet Associations | Ro

**Edit**

| Destination | Target | | Propagated |
|---|---|---|---|
| 203.0.113.0/20 | local | Active | No |
| 0.0.0.0/0 | eni-12345678 / i-abcdefg12 | Active | No |
| 198.51.100.0/24 | vgw-2b2b2b2b | Active | Yes |

You **cannot** override subnet-to-subnet communication with more specific directed routes:

203.0.116.0/23 → Palo Alto eni-1234abcd *eth#*

To Palo Alto eth#



AWS has had multiple customers ask for this feature but no target date. This is why a NACL must be used to enforce subnet-to-subnet communication through the Palo Alto.

# AWS Routing Table Limitation

Want instance subnet-to-subnet communication?

- Have to manually add routes in each instance to Palo Alto on subnet
- We use Puppet to automate this

Instance Routing Table Examples

**Instance in Trust subnet (203.0.116.0/23)**

| Destination | Gateway | Genmask |
|---|---|---|
| 0.0.0.0 | 203.0.116.1 | 0.0.0.0 |
| 203.0.116.0 | 0.0.0.0 | 255.255.254.0 |
| 203.0.115.0 | 203.0.117.254 | 255.255.255.0 |

**Instance in DMZ subnet (203.0.115.0/24)**

| Destination | Gateway | Genmask |
|---|---|---|
| 0.0.0.0 | 203.0.115.1 | 0.0.0.0 |
| 203.0.116.0 | 203.0.115.254 | 255.255.254.0 |
| 203.0.115.0 | 0.0.0.0 | 255.255.255.0 |

The .1 is the AWS Global Router reserved address

.254 is the Palo Alto on the same subnet.

Manual route addition.

Without this no subnet-to-subnet communication can take place (NACL block).



AWS has had multiple customers ask for this feature but no target date. This is why a NACL must be used.

## Other Protection Mechanisms

- Separate AWS Accounts
  - Dev, Test, Prod
  - Consolidated billing
  - Separates AWS IAM roles and credentials
- Multi-factor Authentication
  - ALL AWS accounts must have this*
  - Physical or Virtual token
- Auditing via AWS API
  - List of all instances (assets) collected automatically
    - **Automatically added to security monitoring processes**
  - List of IAM accounts auto-reconciled for current entitlement approval
  - Management of guest instances' (VMs) privileged credentials in vault
  - Detection of unmanaged instances



*We have cases where accounts are setup but the MFA requires manual steps by the user and isn't always followed. Trialed IAM policy to enforce but ran into tricky areas around MFA resets and API usage. For now using manual auditing but working on moving to federated SSO.

## Future Possibility: Event-Driven AWS Security: A Practical Example



1. "Would you like the ability to revert unapproved security group (firewall) changes in Amazon Web Services in 10 seconds, without external tools?"
2. AWS CloudWatch as *event driven security*
   a. "Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS."
3. Author provides example of reverting Security Group (firewall) change automatically using native AWS capabilities.
   a. About 100 lines of code
4. Setup
   a. Uses CloudTrail to feed logs into CloudWatch
   b. Configure IAM roles to allow auto-revert (no security credentials to manage)
   c. Create an AWS Compute "Lambda" function (revertSecurityGroup)
   d. Add an EventTrigger in CloudWatch
5. Demos proof of concept (no change approval mechanism yet)
6. Seagate today has roles for what users can do and monitors/audits changes via Security Monkey

# AWS CloudWatch Pricing

Comes with “Free Tier”

- A $64.20 / month value
- 3 dashboards, 50 metrics
- Basic monitoring at 5 minute interval
- 5GB log ingestion storage

Amazon CloudWatch Dashboards
- $3.00 per dashboard per month

Detailed Monitoring for Amazon EC2 Instances
- $3.50 per instance per month for Detailed Monitoring at 1-minute frequency

Amazon CloudWatch Custom Metrics
- $0.50 per metric per month

Amazon CloudWatch Alarms
- $0.10 per alarm per month

Amazon CloudWatch API Requests
- $0.01 per 1,000 GetMetricStatistics, ListMetrics, or PutMetricData requests

Amazon CloudWatch Logs*
- $0.50 per GB ingested**
- $0.03 per GB archived per month***
- Data Transfer OUT from CloudWatch Logs is priced equivalent to the “Data Tra OUT from Amazon EC2 to Internet” tables on the EC2 Pricing Page.

Amazon CloudWatch Events - Custom Events****
- $1.00 per million custom events generated*****

*As of August 2016. Consult AWS’s public website: https://aws.amazon.com/cloudwatch/pricing/*



$64.20 / month based on if we used the monitoring on all 4 Seagate IT BDC AWS accounts. Excludes S3 storage cost.

# AWS CloudTrail

Required for recording API events into CloudWatch

Includes

- Identity of API caller (IAM)
- Timestamp
- Source IP
- Request parameters
- Response elements

Free Tier

- One trail per region per account
- Search up to 7 days of history for free

Pricing

- $2.00 per 100,000 events *(after free tier)*
- S3 storage cost of $0.03 / GB-month *(during and after free tier)*

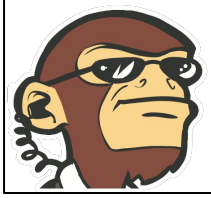
# Limitations of CloudWatch/CloudTail


OpenClipart.org

1. Functions on per-account basis
    a. Meaning manual configuration for each account to get started
    b. Each region has to be setup as well
2. No centralized console for all accounts
3. No out-of-box security reports

Seagate is trying out Security Monkey

a. Netflix open source project
b. Offers consolidated view over all AWS accounts
c. Has out-of-the-box useful security reports
d. Has potential lower cost as Seagate scales out more apps into cloud



# Security Monkey - Open Source Security Tool

- By Netflix
  - Hosts over 2,700 services in AWS(ref)
  - Other familiar project Chaos Monkey(ref)
  - https://securitymonkey.readthedocs.org/en/latest/index.html

- Runs in Amazon AWS
  - Monitors configuration changes in AWS
  - Searchable reporting
  - Alerts on insecure security settings
  - Requires no service account username + password
    - **AWS SDK obtains temporary credentials**
    - **Authorized by IAM roles**
    - **No service account means no password to manage!**

# Security Monkey - Security Group Alert

Security Monkey

Security Monkey Search Reports Settings Signed in as security@seagate.com

Common Services (sg-123 in vpc-abc)

| | |
|---|---|
| Technology | securitygroup |
| Region | us-east-1 |
| Account | Staging - Seagate |

Discovery Timeline 84

Jumplist of revisions Security Monkey has discovered.

Aug 16, 2016 12:58:01 PM

Aug 11, 2016 1:59:05 PM

Item Comments 2

A discussion of this item. Item revisions may also have comments.

Add Comment

Issues 1

Attention! The following issues have been raised and need to be fixed or justified.

| Issue | Score | Notes |
|---|---|---|
| Security Group contains 0.0.0.0/0 | 5 | 0.0.0.0/0 on -1 None |

Justify

Security Monkey

Security Monkey Search Reports Settings Signed in as rodney.d.beede@seagate.com

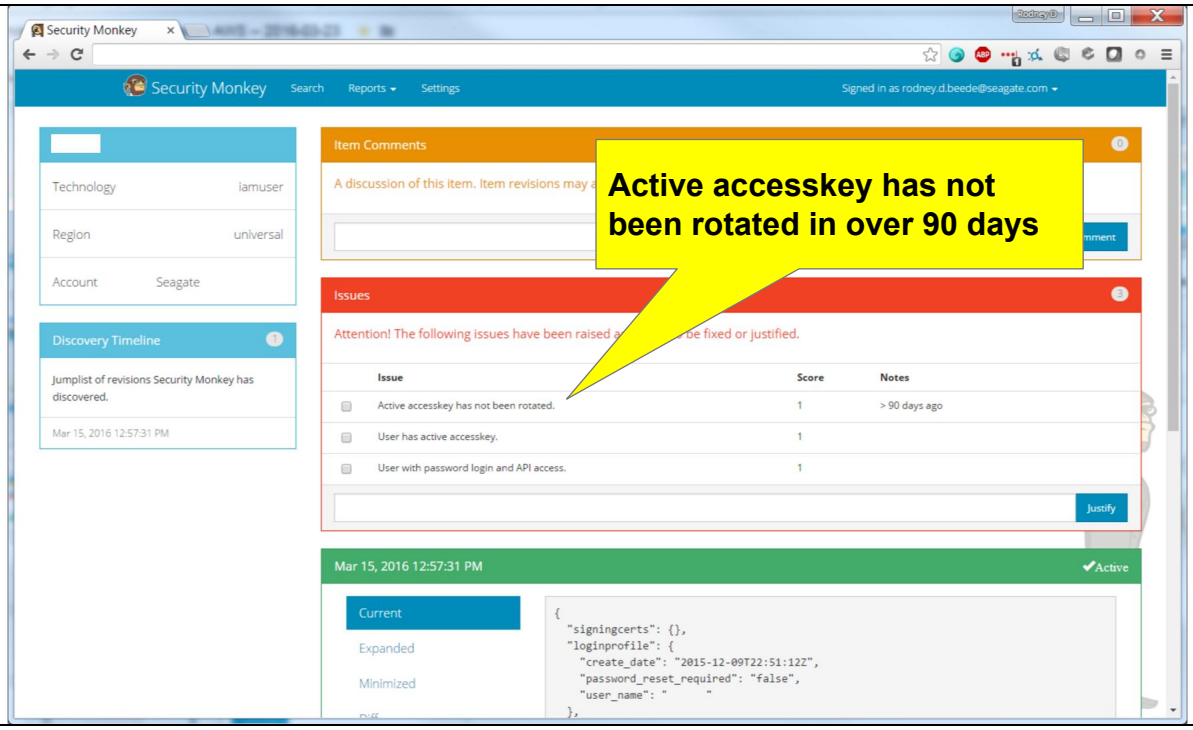
Expanded

Minimized

Diff

**Shows what changed**


```
"rules": [
  {
    "rule_type": "egress",
    "from_port": null,
    "ip_protocol": "-1",
    "to_port": null,
    "owner_id": null,
    "group_id": null,
    "cidr_ip": "0.0.0.0/0",
    "name": null
  },
  {
    "rule_type": "ingress",
    "from_port": null,
    "ip_protocol": "-1",
    "to_port": null,
    "owner_id": null,
    "group_id": null,
    "cidr_ip": "0.0.0.0/0",
    "name": null
  },
  {
    "rule_type": "ingress",
    "from_port": "22",
    "ip_protocol": "tcp",
    "to_port": "22",
    "owner_id": null,
    "group_id": null,
    "cidr_ip": "0.0.0.0/0",
    "name": null
  }
],
"region": "us-east-1",
"description": "With eSecurity permission!",
"assigned_to": [
```


Security Monkey
Search
Reports
Settings
Signed in as rodney.d.beede@seagate.com
Technology
iamuser
Region
universal
Account
Seagate
Discovery Timeline
Jumplist of revisions Security Monkey has discovered.
Mar 15, 2016 12:57:31 PM
Item Comments
A discussion of this item. Item revisions may a
Active accesskey has not been rotated in over 90 days
Issues
Attention! The following issues have been raised a
be fixed or justified.
Issue
Score
Notes
Active accesskey has not been rotated.
1
> 90 days ago
User has active accesskey.
1
User with password login and API access.
1
Justify
Mar 15, 2016 12:57:31 PM
Active
Current
Expanded
Minimized
{
"signingcerts": {},
"loginprofile": {
"create_date": "2015-12-09T22:51:12Z",
"password_reset_required": "false",
"user_name": "
},

# IaaS Cloud Checklist

| Status | Summary | References |
|---|---|---|
| ✓ | Cloud provider has produced current (non-expired) regulatory/industry compliance certifications<br><br>*(ISO 27001, PCI, SOC 1/2/3, HIPAA, SSAE16,...)* | Our Own Company Policy |
| ✓ | Cloud provider has produced CSA CAIQ *(version 3.0.1 or later)* | https://cloudsecurityalliance.org/group/consensus-assessments/ |

# Checklist - Identity and Authentication

| | | |
|---|---|---|
| ✓ 🚩 ✂ | The cloud provider web portal performs user identity using Company Single Sign-On.<br><br>*(Example: SAML)* | Our Own Company Policy |
| ✂ ✓ 🚩 | Any other authentication that does not derive from Company SSO must conform to the authentication standard.<br><br>(Two-factor, password length, complexity, expiration, reuse, guess prevention, temp passwords are random, temp passwords forced to change, etc.) | Our Own Company Policy |
| ✓ 🚩 ✂ | Encrypted communications are used for credential (password, keys, tokens, session ids) transmission with approved secure algorithms and code.<br><br>Example: TLS, SSL, HTTPS | AWS Security Best Practices; aws.amazon.com<br><br>Our Own Company Policy |

# Checklist - Identity and Authentication

| | | |
|---|---|---|
| ✓ | Company personnel have appropriate role based access and entitlements to cloud management portal. | Our Company Policy |
| ✓ | Cloud operator/admin access or user accounts are reconciled on a nightly basis to ensure users still have a valid entitlement. | AWS Security Audit Guidelines |
| ✓ | All cloud provider portal accounts are assigned to individuals and not used as a shared interactive account. | AWS Security Best Practices: aws.amazon.com |

# Checklist - Identity and Authentication

| | | |
|---|---|---|
| ✓ | Service accounts are only used for non-interactive use for authorized processes and managed by the business owner. | Our Company Policy |
| ✓ | The “root” (aka super-admin) cloud provider portal account is restricted in use to only initial provisioning of individual admin accounts and emergencies.<br><br>Individual admin accounts are issued for day-to-day operations. | Lock away your AWS account (root) access keys; aws.amazon.com<br><br>Security Best Practices for Amazon Web Services - Securosis.com<br><br>AWS Security Best Practices; aws.amazon.com |
| ✓ | The “root” (aka super-admin) cloud provider portal account has its credentials stored in a Company IT approved enterprise vault.<br><br>Access to the credential requires approval and is logged. | Our Company Policy |
| ✓ | The “root” (aka super-admin) cloud provider portal account does not have any API access keys, tokens, ssh keys, or other credentials. | Best Practices for Managing AWS Access Keys |

# Checklist - Identity and Authentication

| | | |
|---|---|---|
| ✓ 🚩 | The “root” (aka super-admin) cloud provider portal account requires multi-factor authentication. The MFA token or secret is stored in a physical vault or in an enterprise approved vault.<br><br>Access to the MFA token(s) requires approval and is logged. | Security Best Practices for Amazon Web Services - Securosis.com<br><br>AWS Security Best Practices; aws.amazon.com |
| ✓ 🚩 | The “root” (aka super-admin) cloud provider portal account has random answers to any security challenge questions (i.e. password reset questions) with the answers stored in an enterprise approved vault.<br><br>Answer if no security challenge questions exist or they are disabled. | Security Best Practices for Amazon Web Services - Securosis.com |
| ✓ 🚩 | All (individual or service account) cloud management portal accounts require multi-factor authentication. The MFA secret/token is not recorded in email. | Security Best Practices for Amazon Web Services - Securosis.com<br><br>AWS Security Best Practices; aws.amazon.com |
| ✓ 🚩 | API access keys/tokens are not embedded in any code or scripts.<br><br>Temporary rotating security credentials are used whenever possible. | Security Best Practices for Amazon Web Services - Securosis.com<br><br>AWS Security Best Practices; aws.amazon.com<br><br>Best Practices for Managing AWS Access Keys |

# Checklist - System Security Controls

| | | |
|---|---|---|
| ✓ ⚑ | Access and security logs are centralized to an approved enterprise solution for monitoring by security and follow policy for log retention. | Security Best Practices for Amazon Web Services - Securosis.com |
| ✓ ⚑ | Modifications to cloud configuration is logged and audited.<br><br>*(Example: CloudTrail or Security Monkey for AWS)* | Security Best Practices for Amazon Web Services - Securosis.com |
| ✓ ⚑ | Guest virtual servers are security hardened at launch time according to Company IT Policy.<br><br>*(Example: Predefined customized template with CIS hardening or at immediate launch vendor generic stock image is hardened automatically via automation [i.e. Puppet, Chef, Ansible, etc.])* | AWS Security Best Practices; aws.amazon.com |
| ✓ ⚑ | Security scans are scheduled and run on a weekly basis.<br><br>These scans look for malware (anti-virus) and OS patch levels on guest virtual systems.<br><br>As required, written permission from the cloud provider is in place before the scan is run.<br><br>Results are reported back to the company central enterprise solution. | AWS Security Best Practices; aws.amazon.com<br><br>AWS Vulnerability and Penetration Testing Approval Requirement |

# Checklist - Network Security Controls

| | | |
|---|---|---|
| ✓ 🚩 | Firewalls meet Company IT Policy and are controlled by Firewall Operations.<br><br>*(Example: Firewall can detect and block Heartbleed [CVE-2014-0160] and other advanced types of attacks)* | Our Company Policy |
| ✓ 🚩 | Security Groups are used to only allow least-privilege necessary traffic going east-west.<br><br>*(Example: BizAppA cannot talk to systems for BizAppD because they have no common or shared need/function)* | Security Best Practices for Amazon Web Services - Securosis.com |
| ✓ 🚩 | Firewall rules, Network ACLs, and/or Security Groups limit allowed traffic sources to guest virtual server instances management ports (ssh, rdp). | AWS Security Best Practices; aws.amazon.com |
| ✓ 🚩 | Network ACLs are used to enforce cross-zone traffic (e.g. DMZ<->Trust) to go through the next generation firewall. | Security Best Practices for Amazon Web Services - Securosis.com<br><br>AWS Security Best Practices; aws.amazon.com |

# Checklist - Network Security Controls

| | | |
|---|---|---|
| ✓ | Private networks use NAT where needed for outgoing internet traffic. | Security Best Practices for Amazon Web Services - Securosis.com |
| ✓ | Incoming internet traffic terminates in the DMZ zone. | Our Company Policy |
| ✓ | Cloud provider provides solution/support for DDoS attacks. | AWS Security Best Practices: aws.amazon.com |

# Checklist - Business Security Controls

| | | |
|---|---|---|
| ✓ | Data is encrypted at rest where required by regulatory obligations *(e.g. PCI, HIPAA)*. | AWS Security Best Practices: aws.amazon.com<br><br>Our Company Policy |
| ✓ | Cloud provider provides written contractual SLA for responding to security incidents either reported by customer or from third parties. | AWS Security Best Practices: aws.amazon.com<br><br>Our Company Policy |
| ✓ | Our company has a written security incident response procedure.<br><br>This procedure is practiced at least annually. | Our Company Policy |

# Checklist - Business Security Controls

| | | |
|---|---|---|
| ✓ | Billing and invoicing is done through proper financial accounting channels and procedures to company finance.<br><br>*(I.e. Not billed to a travel AMEX card)* | Our Company Policy |
| ✓ | Separate business accounts are setup for development, staging, production, and billing.<br><br>This implies that separate virtual private networks exist as well and that the resources and access for development versus production are compartmentalized. | Security Best Practices for Amazon Web Services - Securosis.com<br><br>AWS Security Best Practices; aws.amazon.com |

# References


- https://securosis.com
  - Very good material on cloud security and SecDevOps
- Some graphic clipart care of openclipart.org
- Why We Can't Have Nice Things, A Tale of Woe and Hope for the Future
  - Pete Cheslock
  - DevOps Days Austin
  - May 4-5, 2015
- https://securitymonkey.readthedocs.org/en/latest/index.html
- https://aws.amazon.com/documentation/
- Security Best Practices for Amazon Web Services - Securosis.com; January 2015
- AWS Security Best Practices; aws.amazon.com; November 2013
- Root Account Credentials vs. IAM User Credentials; aws.amazon.com; Accessed June 16, 2016
- Lock away your AWS account (root) access keys; aws.amazon.com; Accessed June 16, 2016
- Best Practices for Managing AWS Access Keys; aws.amazon.com; Accessed June 16, 2016
- AWS Security Audit Guidelines; aws.amazon.com; Accessed June 16, 2016
- Aws Multiple Account Security Strategy; awsstatic.com; February 9, 2016; Accessed June 17, 2016
- CAIQ - https://cloudsecurityalliance.org/group/consensus-assessments/