RSA CONFERENCE 2014
FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Share.
Learn.
Secure.
Capitalizing on
Collective Intelligence

Why AWS CloudHSM Can Revolutionize AWS

SESSION ID: CSV-R04A

Oleg Gryb
Security Architect at Intuit
@oleggryb

Todd Cignetti
Sr. Product Manager at AWS Security

Subra Kumaraswamy
Chief Product Security at Apigee

Concerns over Data Security in Cloud!

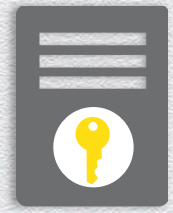
- Data Classification is cumbersome and not followed
- Cloud Security policies and processes may not be transparent
- Perimeter and Infrastructure security not adequate to protect data in cloud
- Data sovereignty and Privacy concerns
- Key management and control of the encryption keys

Why CloudHSM ?

- Lower the barrier for developers to encrypt sensitive data
  - ✓ API and SDKs to abstract complexity
- Encrypted data opaque to Cloud Service Provider's operations staff and cyber criminals
- Horizontally Scalable – CloudHSM as a Service!
- Pay-as-you-go
- Meet compliance goals when data is resident in cloud e.g. PCI-DSS, HIPAA

HSM – Hardware Security Module

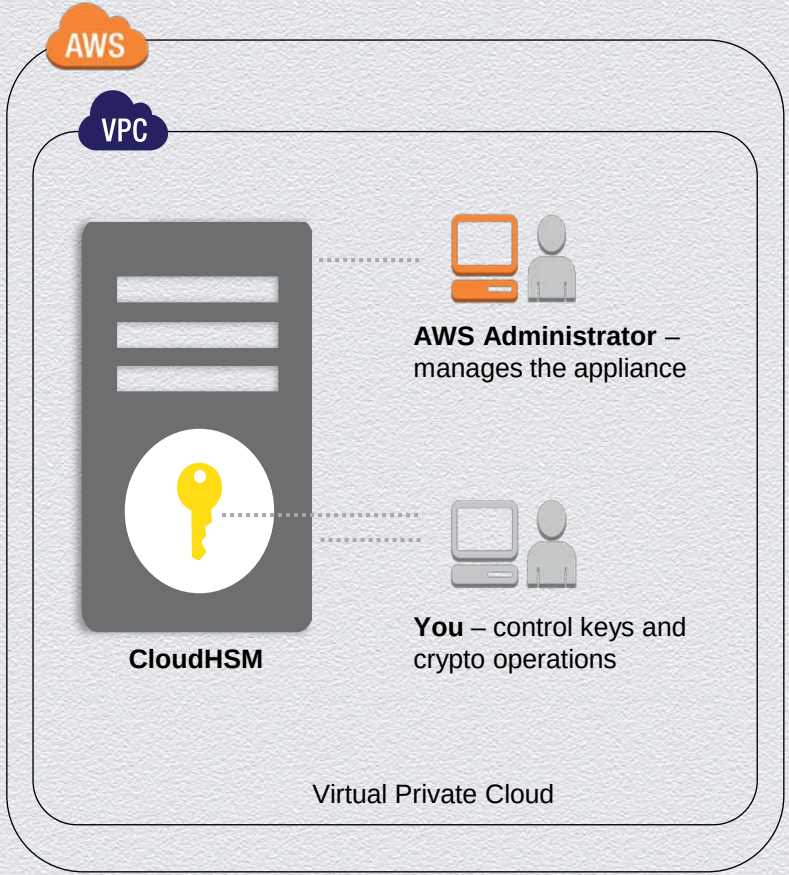
- Hardware device for crypto ops and key storage
- Provides strong protection of private keys
  - Physical device control does not grant access to the keys
  - Security officer controls access to the keys
  - Appliance administrator has no access to the keys
- Certified by 3rd parties to comply with security standards



HSM

AWS CloudHSM

- An AWS service
- You receive dedicated access to HSM appliances
- HSMs are located in AWS datacenters
- Managed & monitored by AWS
- You control the keys
- HSMs are inside your VPC – isolated from the rest of the network



Passing Secrets and EC2 Instance Verification

What is your name, DOB and SSN?

Hey EC2 instance, what is your:

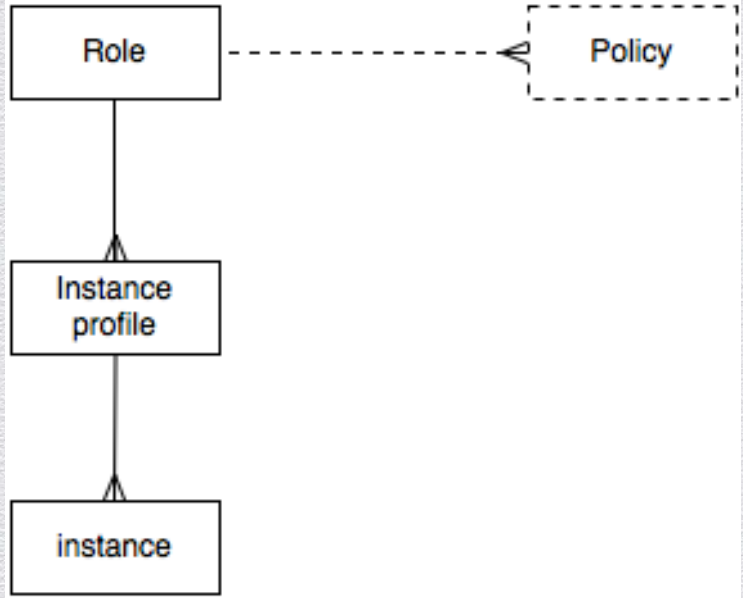
- Internal/external IP's
- instance-id
- public-hostname
- local-hostname
- instance-type

What is your start time?

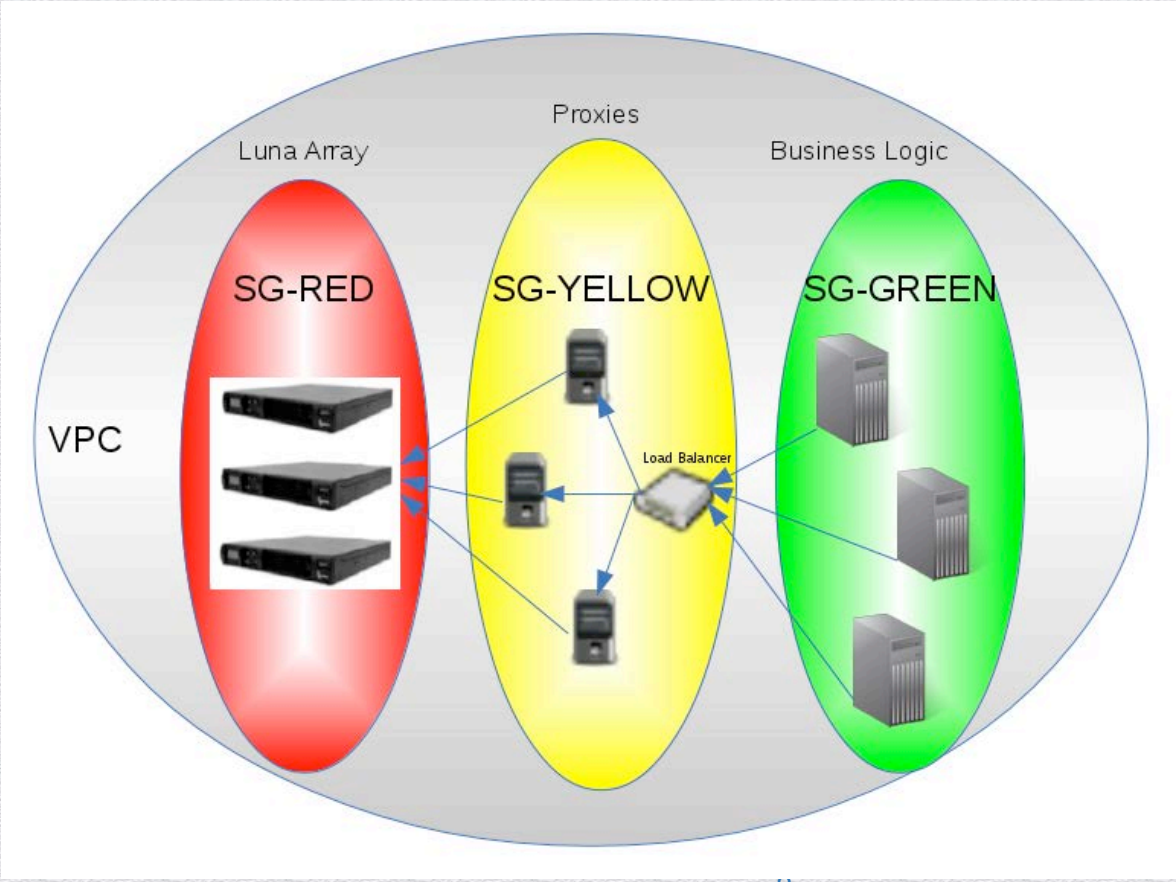
Did you receive the secrets in the past?

Check if Amazon EC2 Instance is in Role

- Role -> Instance profile -> Instance -> Policy Mapping
- Policies are not required to implement “Instance in Role” verification
- It’s not a traditional use of AWS roles, but it fits a generic RBAC model well



Suggested Architecture



Enabling Luna HA Array

- You'll need two or more Luna devices configured absolutely the same
- You'll need to use an haAdmin utility on a client:

- To create a new HA group:

```
$ sudo ./vtl haAdmin -newGroup -serialNum zzzzzzzz \
      -label <group-name> -password <partition-level-pin>
```

- To add a new member to an existing group:

```
$ sudo ./vtl haAdmin -addMember -serialNum xxxxxxxxx \
  -group yyyyyyyy -password <partition-level-pin>
```

Automation

- Python tool that does it all: http://sf.net/p/lunamech
- **Examples.**

Create a single partition

- luna_mech -p -g -r <path-to-partition.cfg>

Create a single Luna

- luna_mech -l -g -r <path-to-luna.cfg>

Create an HA array of Luna's

- luna_mech -a -g -r <path-to-luna-array.cfg>

Summary

The reasons why AWS CloudHSM can change a traditional mindset:

- Keys are owned and accessible by customers **only**
- Multiple layers of security are available:
  - Network level (SSL, certificates)
  - Application level (managers, admin credentials and partition PIN)
- Clusters of HSM's (HA arrays) are supported

Thank You !

Oleg Gryb
Security Architect at Intuit
Twitter: @oleggryb

Subra Kumaraswamy
Chief Product Security at Apigee

Todd Cignetti
Sr. Product Manager at AWS Security

Appendix – More Details about Cloud HSM and its Usage

- What is CloudHSM - 13
- Cloud HSM Details - 14
- Key Storage and Secure Operations - 15
- Customer responsibility - 16
- Client certificates problem and solution – 17,18,19
- Enabling Java client through JCA – 20
- Suggested architecture, security groups – 21
- EC2 Instance Verification - 22

What is CloudHSM ?

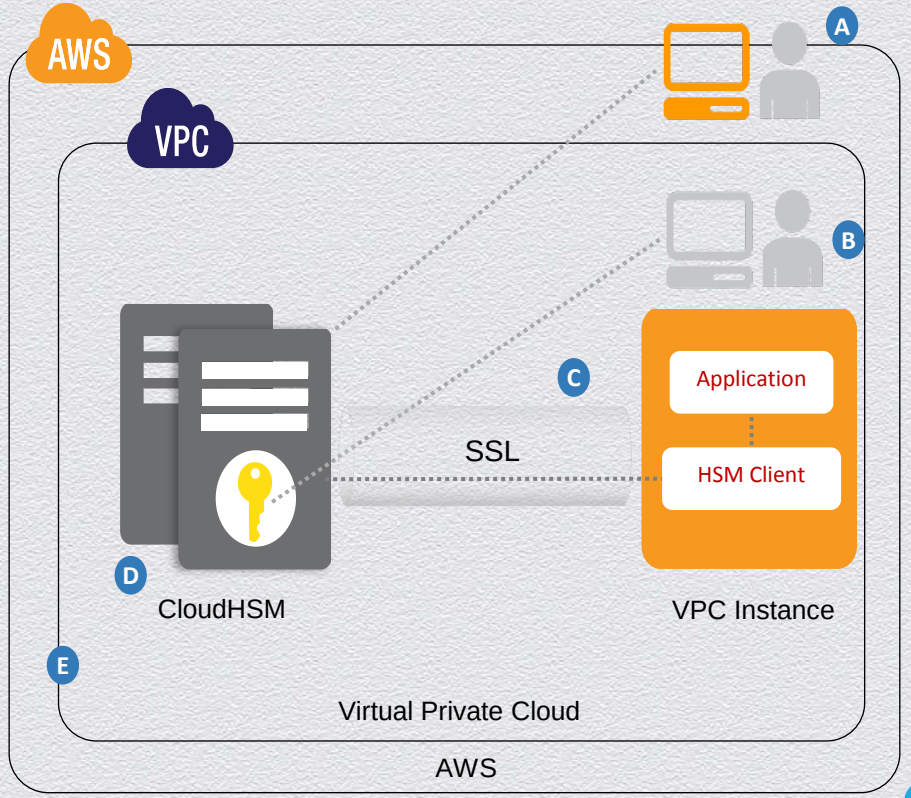
- Provides the same benefits of your Enterprise HSM except it is Co-located with your AWS deployment zone
- “Hardware Security Module as a Service” provided by Cloud Service Providers e.g. AWS
- Usually “Single Tenant” and dedicated to your Enterprise
- Keys never leave HSM!
- APIs supported by HSM vendor SDK e.g. SafeNet
- Protected by multiple layers of security :
  - Subnets @ network level
  - Client/server certs and registration
  - Manager password (connects through ssh)
  - Admin password (init work, create partition)
  - Partition level pin

Details

- **CloudHSM Uses SafeNet® Luna SA HSMs**
  - **Well known and trusted HSM**
  - **Designed for validation by third parties to government standards**
  - **Supports standard APIs**
    - **PKCS#11**
    - **MS CAPI/CNG**
    - **Java JCA/JCE (Java Cryptography Architecture/Java Cryptography Extensions)**
  - **SafeNet HSM Client replaces existing crypto provider library and uses back-end HSM hardware to implement crypto**

Key Storage and Secure Operations for AWS Workloads

- **A** AWS manages the HSM appliance but does not have access to your keys
- **B** You control and manage your own keys
- **C** Application performance improves (due to close proximity with AWS workloads)
- **D** Secure key storage in tamper-resistant hardware available in multiple regions and AZs
- **E** CloudHSMs are in your VPC and isolated from other AWS networks



Customer Responsibility

- After it has been provisioned there are many manual steps:
  - You'll need to initialize the HSM to establish your credentials and control of the HSM
  - You'll need to configure a server and generate server side certificates
  - You'll need to generate a client cert on each client, scp its public portion to the server and register it
  - These steps are an essential part of the security model

Client Certificates

- Normally, you need to generate a client cert on each client and then scp public portion of it to the server:
  - `cd /usr/lunasa/bin`
  - `./vtl createCert -n <cert_name>`
- On the server a normal client registration process requires a client's IP:
  - This is good security wise
  - This is not so good when your clients are running in ASG (which is very common)

Solution for Client Certs

- Fortunately, there is a solution confirmed by both SafeNet and AWS folks:

- The same client cert can be used on all clients
- A generic client name can be used instead of a client's IP when a client is registered on a server

- On client you can run:
  - `cd /usr/lunasa/bin`
  - `./vtl createCert -n <cert_name>`

- On server you can use <cert_name> instead of IP:
- `c reg -c <client_name> -h <cert_name>`

Solution for Client Certs Problem

- The solution is good if you look at it from usability point of view
- From security point of view – probably not that good since once the cert is compromised, it can be used from any IP
- Looks like a necessary tradeoff

- Mitigating controls:
  - More scrutiny @ network level (security groups)
  - Strong and well protected partition level pins

Enabling Java Clients through JCA

- AWS CloudHSM is completely compatible with JCA
- Safenet provider is available and should be integrated with Java env:

1. Add LunaProvider.jar to CLASSPATH
2. Add the provider to java.security

…

```
9=sun.security.smartcardio.SunPCSC
security.provider.10=com.safenetinc.luna.provider.LunaProvi
der
```

Suggested Architecture – SG's

- It's probably not a good idea to allow network access to HSM from all business apps – it's enough to have one compromised node (especially when client certs are shared).

- SG-RED – Luna HA Array subnet. Allows connections to proxy layer only
- SG-YELLOW – Proxies (more than one for scalability). Allows inbound connections from business layer
- SG-GREEN (can be more than one) can connect to proxy layer

Passing Secrets to Cloud Instances

- [illegible]

Manual Setup - Server

1. 'hsm init' command to initialize the device
2. 'sysconf re' command to regenerate server side certificates
3. 'ntls bind' command to restart Luna's network interfaces
4. 'hsm login' to as admin to Luna
5. 'par cr ...' to create a partition
6. 'c reg ...' to register client (normally client IP is required here and this is a pain)
7. 'c a ...' to assign a partition to a client

RSA CONFERENCE 2014
FEBRUARY 24 – 28 | MOSCONE CENTER | SAN FRANCISCO

Cloud Security & Virtualization