Network Security & Access Control in AWS

Ian Massingham, Technical Evangelist

@IanMmmm
AWS Account Security

Day One Governance
Account Governance – New Accounts

- AWS Account Ownership
- AWS Account Contact Information
- AWS Sales and Support Relationship
- InfoSec's Cross-Account Roles
- AWS CloudTrail
- AWS Account Credential Management ("Root Account")
- Federation
- AWS Config

Baseline Requirements
Account Governance – Existing Accounts

- AWS Account Ownership
- AWS Account Contact Information
- AWS Sales and Support Relationship
- InfoSec's Cross-Account Roles
- AWS CloudTrail
- Federation
- AWS Config
- AWS Account Credential Management ("Root Account")

Baseline Requirements
AWS Identity & Access Management

Overview of Core Principles
AWS Identity & Access Management

IAM Users

IAM Groups

IAM Roles
Policy specification basics

- JSON-formatted documents
- Contain a statement (permissions) that specifies:
  - Which actions a principal can perform
  - Which resources can be accessed

```json
{
  "Statement":[{
    "Effect":"effect",
    "Principal":"principal",
    "Action":"action",
    "Resource":"arn",
    "Condition":{
      "condition":{
        "key":"value"
      }
    }
  }
}
```

You can have multiple statements and each statement is comprised of PARC.
Managing your policies

IAM policies

Managed policies

Inline polices

Resource-based policies
IAM policies

- Managed policies (newer way)
  - Can be attached to multiple users, groups, and roles
  - AWS managed policies: Created and maintained by AWS
  - Customer managed policies: Created and maintained by you
    - Up to 5K per policy
    - Up to 5 versions of a policy so you can roll back to a prior version
  - You can attach 10 managed policies per user, group, or role
  - You can limit who can attach which managed policies

- Inline policies (older way)
  - You create and embed directly in a single user, group, or role
  - Variable policy size (2K per user, 5K per group, 10K per role)
Resource-based policies

IAM policies live with:
- IAM users
- IAM groups
- IAM roles

Some services allow storing policy with resources:
- S3 (bucket policy)
- Amazon Glacier (vault policy)
- Amazon SNS (topic policy)
- Amazon SQS (queue policy)
- AWS KMS (key policy)

```
{
    "Statement":
    {
        "Effect": "Allow",
        "Principal": {"AWS": "111122223333"},
        "Action": "sqs:SendMessage",
        "Resource":
            "arn:aws:sqs:us-east-1:444455556666:queue1"
    }
}
```
AWS CloudTrail
You are making API calls...

On a growing set of AWS services around the world...

CloudTrail is continuously recording API calls

Store/Archive

Troubleshoot

Monitor & Alarm

AWS Management Console

SDK

AWS CLI

You are making API calls...

On a growing set of AWS services around the world...

CloudTrail is continuously recording API calls

Store/Archive

Troubleshoot

Monitor & Alarm

AWS Management Console

SDK

AWS CLI
Use cases enabled by CloudTrail

IT and security administrators can perform security analysis.

IT administrators and DevOps engineers can attribute changes on AWS resources to the identity, time and other critical details of who made the change.

DevOps engineers can troubleshoot operational issues.

IT Auditors can use log files as a compliance aid.

[Security at Scale: Logging in AWS White Paper](#)
Encrypted CloudTrail log files using SSE-KMS

By default, CloudTrail encrypts log files using S3 server side encryption.

Additional layer of security for your log files by encrypting with your KMS key.

Application logic for ingesting and processing log files stays the same.

S3 will decrypt on your behalf if your credentials have decrypt permissions.
Encrypting your log files using SSE KMS

Step 1: Create or use an existing KMS Key and apply policy

Step 2: Grant decrypt access to log readers

Step 3: Specify KMS key to CloudTrail

Step 4: S3 GetObject API call

Step 5: Decrypted CloudTrail log files
CloudTrail log file integrity validation

Validate that a log file has not been changed since CloudTrail delivered the log file to your S3 bucket.

Detect whether a log file was deleted or modified or unchanged.

Use the tool as an aid in your IT security, audit and compliance processes.
AWS Config
AWS Config

- Get inventory of AWS resources
- Discover new and deleted resources
- Record configuration changes continuously
- Get notified when configurations change
AWS Config
AWS Config

EC2 VPC vpc-27148f46
at October 25, 2015 9:00:46 AM PDT (UTC-07:00)

06th October 2015
9:08:08 PM

14th October 2015
4:37:56 AM

14th October 2015
9:51:28 AM

17th October 2015
9:51:14 PM

18th October 2015
3:51:42 AM

Configuration Details
Config Rules (preview)

- Set up rules to check configuration changes recorded
- Use pre-built rules provided by AWS
- Author custom rules using AWS Lambda
- Invoked automatically for continuous assessment
- Use dashboard for visualizing compliance and identifying offending changes
AWS Config – Rules
(example – instances must be tagged with a DataClassification)
AWS Network Security – Global Networking

Building a Robust Internet Architecture
Amazon CloudFront, AWS WAF and Amazon Route 53
CloudFront - Shield custom origin

- Shield your custom origin
- Whitelist Amazon CloudFront IP range
AWS Network Security - VPC

Building a Trust Zone architecture
VPC CIDR 10.1.0.0/16

Availability Zone A
- Private subnet
- Web
- Back end

Availability Zone B
- Private subnet
- Web
- Back end

ELB
Security Groups

VPC CIDR 10.1.0.0/16

Availability Zone A
- Private subnet
- Web
- Back end
- sg_ELB_FrontEnd (ELB Security Group)
- sg_Web_Frontend (Web Security Group)
- sg_Backend (Backend Security Group)

Availability Zone B
- Private subnet
- ELB
- Web
- Back end
- sg_ELB_FrontEnd (ELB Security Group)
- sg_Web_Frontend (Web Security Group)
- sg_Backend (Backend Security Group)
### Security Groups

**VPC Dashboard**

<table>
<thead>
<tr>
<th>Filter by VPC:</th>
<th>None</th>
</tr>
</thead>
</table>

#### Virtual Private Cloud

- **Your VPCs**
- **Subnets**
- **Route Tables**
- **Internet Gateways**
- **DHCP Options Sets**

#### Create Security Group

<table>
<thead>
<tr>
<th>Name tag</th>
<th>Group ID</th>
<th>Group Name</th>
<th>VPC ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sg_ELB_FrontEnd</td>
<td>sg-9142e0f7</td>
<td>sg_ELB_FrontEnd</td>
<td>vpc-9df15bf9 (10.50.0.0/16)</td>
<td>ELB Security Group</td>
</tr>
<tr>
<td>sg_Backend</td>
<td>sg-3e40e258</td>
<td>sg_Backend</td>
<td>vpc-9df15bf9 (10.50.0.0/16)</td>
<td>Backend Security Group</td>
</tr>
<tr>
<td>sg_Web_Frontend</td>
<td>sg-7640e210</td>
<td>sg_Web_Frontend</td>
<td>vpc-9df15bf9 (10.50.0.0/16)</td>
<td>Web Security Group</td>
</tr>
</tbody>
</table>
### Security Groups

<table>
<thead>
<tr>
<th>Name tag</th>
<th>Group ID</th>
<th>Group Name</th>
<th>VPC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sg_ELB_FrontEnd</td>
<td>sg-9142e0f7</td>
<td>sg_ELB_FrontEnd</td>
<td>vpc-9df15bf9</td>
<td>ELB Security Group</td>
</tr>
<tr>
<td>sg_Backend</td>
<td>sg-3e40e258</td>
<td>sg_Backend</td>
<td>vpc-9df15bf9 (10.50.0.0/16)</td>
<td>Backend Security Group</td>
</tr>
<tr>
<td>sg_Web_Frontend</td>
<td>sg-7640e210</td>
<td>sg_Web_Frontend</td>
<td>vpc-9df15bf9</td>
<td>Web Security Group</td>
</tr>
</tbody>
</table>

#### sg_ELB_FrontEnd

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP (80)</td>
<td>TCP (6)</td>
<td>80</td>
<td>0.0.0.0/0</td>
</tr>
<tr>
<td>HTTPS (443)</td>
<td>TCP (6)</td>
<td>443</td>
<td>0.0.0.0/0</td>
</tr>
</tbody>
</table>
## Security Groups

![Security Group Table]

### Filters
- **Full Name Tag**: Enter any text to search for specific VPC security groups.

### Columns
- **Name Tag**: Tag associated with the security group.
- **Group ID**: Unique identifier for the security group.
- **Group Name**: Name of the security group.
- **VPC**: Virtual Private Cloud where the group resides.
- **Description**: Description of the security group.

### Security Group Details
- **sg_ELB_FrontEnd**: Group ID `sg-9142e0f7`, VPC `vpc-9df15bf9`, Description: ELB Security Group.
- **sg_Backend**: Group ID `sg-3e40e258`, VPC `vpc-9df15bf9` (10.50.0.0/16), Description: Backend Security Group.
- **sg_WEB_FrontEnd**: Group ID `sg-7640e210`, VPC `vpc-9df15bf9`, Description: Web Security Group.

### Inbound Rules
- **Type**: HTTPS, Protocol: TCP, Port Range: 8443, Source: sg-9142e0f7.

---

*Image of the AWS management console with security group settings.*
### Security Groups

<table>
<thead>
<tr>
<th>Name tag</th>
<th>Group ID</th>
<th>Group Name</th>
<th>VPC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sg_ELB_FrontEnd</td>
<td>sg-9142e0f7</td>
<td>sg_ELB_FrontEnd</td>
<td>vpc-9df15bf9</td>
<td>ELB Security Group</td>
</tr>
<tr>
<td>sg_Backend</td>
<td>sg-3e40e258</td>
<td>sg_Backend</td>
<td>vpc-9df15bf9</td>
<td>Backend Security Group</td>
</tr>
<tr>
<td>sg_Web_Frontend</td>
<td>sg-7640e210</td>
<td>sg_Web_Frontend</td>
<td>vpc-9df15bf9</td>
<td>Web Security Group</td>
</tr>
</tbody>
</table>

**sg-3e40e258 | sg_Backend**

**Inbound Rules**

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPS*</td>
<td>TCP (6)</td>
<td>8443</td>
<td>sg-7640e210</td>
</tr>
</tbody>
</table>
Network Access Control Lists (NACLS)
And what if instances in a private subnet need to reach outside the VPC?

They have no route to the IGW and no public IP address.
Why go outside?

- AWS API endpoints
- Regional services
- Third-party services
To NAT, or not to NAT…

• Leave NAT for less bandwidth-critical connectivity

• Don’t bottleneck high-bandwidth-out workloads

• Run high-bandwidth components from public subnets

• Goal is full-instance bandwidth out of VPC
EC2 status checks

CloudWatch

per-instance metrics:
- StatusCheckFailed_System
- StatusCheckFailed_Instance
Amazon CloudWatch alarm actions

Instance status check fails?
- REBOOT

System status check fails?
- RECOVER

Instance retains:
- Instance ID
- Instance metadata
- Private IP addresses
- Elastic IP addresses
- EBS volume attachments
A few things to remember...

Amazon EC2 Auto Recovery

- Recover action only applies to system status checks
- Limited to C3, C4, M3, R3, and T2 instance types
- Cannot use local instance store
- Cannot be dedicated instances
- Use EC2ActionsAccess AWS Identity and Access Management (IAM) role
Amazon EC2 Auto Recovery

CloudWatch Console

Metric = StatusCheckFailed_System

Set your failed check threshold

Choose recover action

Choose 1-minute period and statistic minimum
Amazon EC2 Auto Reboot

CloudWatch Console

Metric = StatusCheckFailed_Instance

Choose reboot action

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate thresholds.

Name: HA-NAT-Reboot
Description: Reboot Amazon Linux HA NAT instance
Whenever: StatusCheckFailed_Instance is > 1 for 2 consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

EC2 Action
Whenever this alarm: State is ALARM
Take this action:
- Recover this instance
- Stop this instance
- Terminate this instance
- Reboot this instance

This will reboot your EC2 instance (i-05a8c203)

Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 2 minutes.
HA NAT with EC2 Auto Recovery + Auto Reboot

Average tested recovery time: ~ 1 to 4 minutes
Could be shorter or longer depending on nature of failure
Pick a NAT, any NAT

Amazon Linux NAT Amazon Machine Image (AMI)

Amazon Linux AMI VPC NAT x86_64 HVM GP2
Root device type: ebs   Virtualization type: hvm
AWS region

- Public-facing web app
- Internal company app

Customer network

VPN connection
But apps want to leverage…

Amazon S3

…as a primary data store
You really don’t want to do this:

- Availability Zone A
  - Private subnet
  - Intranet app
  - Virtual Private Gateway
- Availability Zone B
  - Private subnet
  - Intranet app

AWS region

Internet

Amazon S3

VPN connection

Customer VPN

Customer network

Customer border router
So do this instead:

VPC Endpoints

- No IGW
- No NAT
- No public IPs
- Free
- Robust access control
VPC endpoints

From the Amazon VPC User Guide:

“Currently, we support endpoints for connections with Amazon S3 within the same region only. We'll add support for other AWS services later.”

$ aws ec2 describe-vpc-endpoint-services
  SERVICENAMES   com.amazonaws.us-west-2.s3
Creating S3 VPC endpoint

```bash
aws ec2 create-vpc-endpoint
--vpc-id vpc-40f18d25
--service-name com.amazonaws.us-west-2.s3
--route-table-ids rtb-2ae6a24f rtb-61c78704
```

<table>
<thead>
<tr>
<th>Route Table</th>
<th>Destination</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10.1.0.0/16</td>
<td>Local</td>
</tr>
<tr>
<td></td>
<td>Corp CIDR</td>
<td>VGW</td>
</tr>
<tr>
<td></td>
<td>Prefix List for S3 us-west-2</td>
<td>VPCE</td>
</tr>
</tbody>
</table>
Creating S3 VPC endpoint

```
aws ec2 create-vpc-endpoint
--vpc-id vpc-40f18d25
--service-name com.amazonaws.us-west-2.s3
--route-table-ids rtb-2ae6a24f rtb-61c78704
```
Prefix lists

aws ec2 describe-prefix-lists
PREFIXLISTS pl-68a54001 com.amazonaws.us-west-2.s3
CIDRS 54.231.160.0/19

• Logical route destination target
• Dynamically translates to service IPs
• S3 IP ranges change over time
• S3 prefix lists abstract change
Prefix lists

... and use them in security groups!
Controlling VPC access to Amazon S3

**IAM policy on VPCE:**

```json
{
    "Statement": [
        {
            "Sid": "vpce-restrict-to-backup-bucket",
            "Principal": "*",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::backups-reinvent2015",
                "arn:aws:s3:::backups-reinvent2015/*"
            ]
        }
    ]
}
```
Controlling VPC access to Amazon S3

S3 bucket policy:

```
{
    "Statement": [
        {
            "Sid": "bucket-restrict-to-specific-vpce",
            "Principal": "*",
            "Action": "s3:*",
            "Effect": "Deny",
            "Resource": 
                ["arn:aws:s3:::backups-reinvent2015",
                 "arn:aws:s3:::backups-reinvent2015/*"],
            "Condition": {
                "StringNotEquals": {
                    "aws:sourceVpce": "vpce-bc42a4e5"
                }
            }
        }
    ]
}
```
Controlling VPC access to Amazon S3

Recap on security layers:

1. Route table association
2. VPCE policy
3. Bucket policy
4. Security groups with prefix list
Endpoints in action

AWS region

Compliance

Backups

VPCE1

VPCE2

Compliance app

Intranet apps

Private subnet

Private subnet

Private subnet
Endpoints in action
VPC Flow Logs
VPC Flow Logs

- Agentless
- Enable per ENI, per subnet, or per VPC
- Logged to AWS CloudWatch Logs
- Create CloudWatch metrics from log data
- Alarm on those metrics

<table>
<thead>
<tr>
<th>Interface</th>
<th>Source IP</th>
<th>Source port</th>
<th>Protocol</th>
<th>Packets</th>
<th>Accept or reject</th>
</tr>
</thead>
<tbody>
<tr>
<td>eni-b30b9cd5</td>
<td>119.147.115.32</td>
<td>10.1.1.179</td>
<td>6000</td>
<td>22 6 1 40</td>
<td>1442975475 1442975535 REJECT OK</td>
</tr>
<tr>
<td>eni-b30b9cd5</td>
<td>169.54.233.117</td>
<td>10.1.1.179</td>
<td>21188</td>
<td>80 6 1 40</td>
<td>1442975535 1442975595 REJECT OK</td>
</tr>
<tr>
<td>eni-b30b9cd5</td>
<td>212.7.209.6</td>
<td>10.1.1.179</td>
<td>3389</td>
<td>3389 6 1 40</td>
<td>1442975596 1442975655 REJECT OK</td>
</tr>
<tr>
<td>eni-b30b9cd5</td>
<td>189.134.227.225</td>
<td>10.1.1.179</td>
<td>39664</td>
<td>23 6 2 130</td>
<td>1442975656 1442975716 REJECT OK</td>
</tr>
<tr>
<td>eni-b30b9cd5</td>
<td>77.85.113.238</td>
<td>10.1.1.179</td>
<td>0 0 1 100</td>
<td>1442975656 1442975716 REJECT OK</td>
<td></td>
</tr>
<tr>
<td>eni-b30b9cd5</td>
<td>10.1.1.179</td>
<td>198.60.73.8</td>
<td>512</td>
<td>123 17 1 76</td>
<td>1442975776 1442975836 ACCEPT OK</td>
</tr>
</tbody>
</table>
VPC Flow Logs: Automation

- Compliance app
- Private subnet

Flow Log group

Filter on all SSH REJECT

If SSH REJECT > 10, then...

CloudWatch Logs

Metric filter

CloudWatch alarm

Source IP

Elastic Network Interface

Amazon SNS

AWS Lambda
VPC Flow Logs

AWS VPC Flow - Live - Overview
# VPC Flow Logs

![VPC Flow Logs](image.png)

## Table:

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Port</th>
<th>Protocol</th>
<th>Status</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>50.62.129.199</td>
<td>53</td>
<td>UDP</td>
<td>REJECT</td>
<td>10/01/2015 12:07:06 PM</td>
</tr>
<tr>
<td>111.72.252.91</td>
<td>27254</td>
<td>TCP</td>
<td>REJECT</td>
<td>10/01/2015 12:07:06 PM</td>
</tr>
<tr>
<td>109.53.37.95</td>
<td>31894</td>
<td>TCP</td>
<td>REJECT</td>
<td>10/01/2015 12:07:07 PM</td>
</tr>
<tr>
<td>192.129.233.106</td>
<td>58617</td>
<td>UDP</td>
<td>REJECT</td>
<td>10/01/2015 12:07:07 PM</td>
</tr>
<tr>
<td>69.85.183.27</td>
<td>123</td>
<td>UDP</td>
<td>ACCEPT</td>
<td>10/01/2015 12:08:07 PM</td>
</tr>
<tr>
<td>DB1</td>
<td>123</td>
<td>UDP</td>
<td>ACCEPT</td>
<td>10/01/2015 12:08:07 PM</td>
</tr>
<tr>
<td>DB2</td>
<td>123</td>
<td>UDP</td>
<td>ACCEPT</td>
<td>10/01/2015 12:09:10 PM</td>
</tr>
</tbody>
</table>

## Diagram:

- **External**
- **DMZ**
- **Partially Open**
- **Internal Zone**

**Legend:**
- **DB servers**
- **Monitoring**
- **Web servers**
- **App servers**

Selected paths and nodes:

- **Default path**
- **Security tags**
- **Common SG**
- **DB servers**
VPC Flow Logs

- Amazon Elasticsearch Service (ES)
- Amazon CloudWatch Logs subscriptions
- Kibana

Refreshment Break
Please be back for 15:10