Hardening AWS Environments

and

Automating Incident Response

for

AWS Compromises
Agenda: Preparing for an Incident within AWS

Incident Handling

Automatic Collection of Evidence

Hardening the AWS Environment
AWS Key Compromise

Dev put AWS keys on Github. Then BAD THINGS happened

Security

Bots are crawling all over GitHub seeking secret keys, a developer served with a $2,375 Bitcoin mining bill found.

My AWS account was hacked and I have a $50,000 bill, how can I reduce the amount I need to pay?

For years, my bill was never above $350/month on my single AWS instance. Then over

Ryan Hellyer’s AWS Nightmare: Leaked Access Keys Result in a $6,000 Bill Overnight

My run in with Unauthorised Litecoin mining on AWS

Posted by Luke Chadwick on © December 16, 2013
How are Keys Compromised?

AWS Keys provided by AWS for AWS SDK

Keys may be stored in a code repository

Keys may be stored on another AWS Instance
More Serious Attacks

Backdooring an AWS account

So you've pwned an AWS account—congratulations—now what? You're eager to get to the data theft, *amirite*? Not so fast grasshopper, have you disrupted logging? Choice! Time to look around and understand what you have.

No one wants to get inside and find the temporary credentials. A common method to access these is the `aws sts get-session-info` command.

Exploring an AWS account post-compromise

Your instinct is probably to type “whoami” and luckily AWS has an equivalent.

```
aws sts get-caller-identity
```

It won’t give you much but it will start painting the picture. The information returned is “not secret” but it can be painful to obtain otherwise. For example, AWS Resource Names (ARNs) are everywhere.
IR-Phases

- Preparation
- Detection & Analysis
- Containment Eradication & Recovery
- Post-Incident Activity
Where we help

- Preparation
- Detection & Analysis
- Containment Eradication & Recovery
- Post-Incident Activity
DevSecOps
IR Workflow as it relates to AWS

Locating an Instance
Across AWS

Managing credentials

Understanding where your config is not best practice
# AWS Services to Enable Today

<table>
<thead>
<tr>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudWatch Metrics</td>
</tr>
<tr>
<td>CloudTrail</td>
</tr>
<tr>
<td>AWS Config</td>
</tr>
<tr>
<td>CloudWatch Events</td>
</tr>
<tr>
<td>EC2-Run / IAM</td>
</tr>
</tbody>
</table>
Increasing Visibility with CloudWatch
Increasing Visibility with CloudTrail
EC2-Run Example

How to get going with EC2-Command or Simple Server Management

http://amzn.to/2aiq8kc

Installation is Easy!

```bash
#!/bin/bash
cd /tmp
curl https://amazon-ssm-u ... amazon-ssm-agent.rpm -o amazon-ssm-agent.rpm
yum install -y amazon-ssm-agent.rpm
```
Why would you want to do this?

Can come in useful in a security incident.

Out of band management.

IAM Role driven.
Account Access Scenario

Imagine you are completely locked out.
Select Run a Command

A command document includes the information about the command you want to run. Select a command document from the following list and then specify parameters for the command.

### Command document

<table>
<thead>
<tr>
<th>Name</th>
<th>Owner</th>
<th>Platform type</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS-ConfigureCloudWatch</td>
<td>Amazon</td>
<td>Windows</td>
</tr>
<tr>
<td>AWS-ConfigureWindowsUpdate</td>
<td>Amazon</td>
<td>Windows</td>
</tr>
<tr>
<td>AWS-FirstWindowsUpdates</td>
<td>Amazon</td>
<td>Windows</td>
</tr>
<tr>
<td>AWS-InstallApplication</td>
<td>Amazon</td>
<td>Windows</td>
</tr>
<tr>
<td>AWS-InstallMissingWindowsUpdates</td>
<td>Amazon</td>
<td>Windows</td>
</tr>
<tr>
<td>AWS-InstallPowerShellModule</td>
<td>Amazon</td>
<td>Windows</td>
</tr>
<tr>
<td>AWS-InstallSpoolerWindowUpdates</td>
<td>Amazon</td>
<td>Windows</td>
</tr>
<tr>
<td>AWS-JoinDirectoryServiceDomain</td>
<td>Amazon</td>
<td>Windows</td>
</tr>
<tr>
<td>AWS-ListWindowsInventory</td>
<td>Amazon</td>
<td>Windows</td>
</tr>
<tr>
<td>AWS-RunPowerShellScript</td>
<td>Amazon</td>
<td>Windows</td>
</tr>
<tr>
<td>AWS-RunPowershell</td>
<td>Amazon</td>
<td>Linux</td>
</tr>
<tr>
<td>AWS-UpdateEC2Config</td>
<td>Amazon</td>
<td>Windows</td>
</tr>
<tr>
<td>AWS-UpdateSSMAgent</td>
<td>Amazon</td>
<td>Windows, Linux</td>
</tr>
</tbody>
</table>

**Description:** Export metrics and log files from your instances to Amazon CloudWatch.
Find the instance

<table>
<thead>
<tr>
<th>Name</th>
<th>Instance ID</th>
<th>Instance State</th>
<th>Availability Zone</th>
<th>Ping Status</th>
<th>Last Ping Date</th>
<th>Agent Version</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>i-0043B1e</td>
<td>running</td>
<td>us-west-2a</td>
<td>Online</td>
<td>July 20, 2016</td>
<td>1.2.200.3</td>
<td>Link</td>
</tr>
</tbody>
</table>
Input the command

```bash
chmod 0600 /home/ec2-user/.ssh/authorized_keys
```
Execute the command

At the end you can simply click run and you've taken back the instance.
How does it work?

Instance Launches with IAM Role
Giving it EC2-Command Access

Client Polls SSM API
Client Receives Work

SecOps Asks EC2 to Run Command

Work Executed

AWS
Recommended viewing

AWS re:Invent 2015 | (SEC316) Harden Your Architecture w/ Security Incident Response Simulations

IAM Role Advice

1. Use least privilege roles.
2. Audit their usage with CloudTrail
How do IAM roles work?

Create IAM Role with some permissions

Attach to instance at runtime

Instance Assumes Role

Credentials Rotated Regularly

STS
IAM Limits

1. Instance profiles can't be detached.
2. Instance profiles can't be added to a running instance.
Config is a relatively new service that performs inventory, tracks changes, and can enforce compliance.
Config vs Config Rules

AWS Config

Timeline of Changes

Config Rules:

Run periodically and evaluate compliance.
Wizard Driven Setup

What to check

Where to store it

What to notify
Config Timeline
Config Rules

Add AWS managed rule

AWS Config evaluates your AWS resources against this rule when it is triggered.

Name: restricted-ssh
A unique name for the rule. 64 characters max. No special characters or spaces.

Description: Checks whether security groups that are in use disallow unrestricted incoming SSH traffic.

Managed rule name: INCOMING_SSH_DISABLED

Trigger
AWS Config evaluates resources when the trigger occurs.

Trigger type: Configuration changes

Scope of changes: Resources

Resources: EC2: SecurityGroup

Resource identifier (optional)
This is Config running the first evaluation of the rule.
This is Config reporting on non-compliance.
Config and Lambda

Security improves with automated response.
CloudWatchEvents and Lambda

Video Demonstration
Recommended Viewing

There's also a great presentation about this:


http://amzn.to/2aN6Js5
Access Advisor

Great tutorial on getting going with access advisor: http://amzn.to/2aN6Js5
Access advisor shows the service permissions granted to this user and when those services were last accessed. You can use this information to refine your policies. [Learn more]

Note: recent activity usually appears within 4 hours. Access Advisor tracking began on Oct 1, 2015. [Learn more]

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Access by Entities</th>
<th>Last Accessed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon S3</td>
<td></td>
<td>Not accessed in the tracking period</td>
</tr>
</tbody>
</table>
Advice to take away

Use custom policies

Audit them using access Advisor

Revoke permissions you don't need
Tool Gaps
Mission

Be a free open source incident response toolkit tailored for Amazon Web Services. Help first responders by automating workflows using Amazon's very own boto3 pip module.
The Question?

Can we leverage the AWS API to perform incident response?
Host Based

vs

Key Based
Key Compromise

aws_ir username, ssh-key

Perform Host Process on Compromised Hosts
In key compromise we always want to disable the key.
Perform Host Process on Compromised Hosts

aws_ir username, ssh-key

1. AWS-IR hunts instance
2. AWS-IR stops egress
3. AWS-IR Generates Case Number and Bucket
4. AWS-IR takes snapshot
5. AWS-IR uses margarita shotgun to acquire memory to bucket
6. AWS-IR collects metadata, console logs, and screenshot
7. AWS-IR powers down instance
Key Compromise Demo
Now to host based compromises with AWS_IR
AWS_IR Usage

[krug@bb-8 lots_of_haxx ]$ aws_ir
aws_ir host_compromise
usage:
aws_ir host_compromise
ip user ssh_key_file
Step 1

1. AWS-IR hunts instance
Step 2

2. AWS-IR stops egress
Step 3

3. AWS-IR Generates Case Number and Bucket
Step 4

4. AWS IR takes snapshot
Step 5

5. AWS-IR uses margarita shotgun to acquire memory to bucket
Step 6

6. AWS-IR collects metadata, console logs, and screenshot
Step 7

7. AWS-IR powers down instance
The whole picture

aws_ir username, ssh-key

1. AWS-IR hunts instance
2. AWS-IR stops egress
3. AWS-IR Generates Case Number and Bucket
4. AWS IR takes snapshot
5. AWS-IR uses margarita shotgun to acquire memory to bucket
6. AWS-IR collects metadata, console logs, and screenshot
7. AWS-IR powers down instance
Margarita Shotgun
Kernel warehouse is a ruby gem that builds all the modules for all support AWS linux variants.
You can host your own or use ours.
ThreatResponse Workstation
Starting ThreatResponse Workstation

$ aws_ir host_compromise 52.42.254.41 ec2-user key.pem

... 

... 

Processing complete: Launch an analysis workstation with the command

aws_ir -n cr-16-072816-a4d6 create_workstation us-west-2

$ aws_ir host_compromise -c 52.42.254.41 ec2-user key.pem
ThreatResponse Dashboard
**ThreatResponse Acquire**

<table>
<thead>
<tr>
<th>InstanceId</th>
<th>Public IP Address</th>
<th>Region</th>
<th>Action</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-0d9c5d50</td>
<td>null</td>
<td>us-east-1</td>
<td>ADD CREDENTIALS</td>
<td>ADD TO CASE</td>
</tr>
<tr>
<td>i-06144796</td>
<td>null</td>
<td>us-east-1</td>
<td>ADD CREDENTIALS</td>
<td>ADD TO CASE</td>
</tr>
<tr>
<td>i-08144796</td>
<td>null</td>
<td>us-east-1</td>
<td>ADD CREDENTIALS</td>
<td>ADD TO CASE</td>
</tr>
<tr>
<td>i-09144799</td>
<td>null</td>
<td>us-east-1</td>
<td>ADD CREDENTIALS</td>
<td>ADD TO CASE</td>
</tr>
<tr>
<td>i-0a14479a</td>
<td>null</td>
<td>us-east-1</td>
<td>ADD CREDENTIALS</td>
<td>REMOVE</td>
</tr>
<tr>
<td>i-0b14479b</td>
<td>null</td>
<td>us-east-1</td>
<td>ADD CREDENTIALS</td>
<td>ADD TO CASE</td>
</tr>
</tbody>
</table>
Analyze - Memory

Process and Analyze Assets for cr-16-071619-cdd5

Memory

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date Created UTC</th>
<th>Analyze</th>
</tr>
</thead>
<tbody>
<tr>
<td>54.193.96.69-memlima</td>
<td>1023 MB</td>
<td>2016-07-16 19:54:49+00:00</td>
<td></td>
</tr>
<tr>
<td>54.215.134.74-memlima</td>
<td>1023 MB</td>
<td>2016-07-16 20:41:58+00:00</td>
<td></td>
</tr>
</tbody>
</table>
ThreatResponse Analyze - Disk

Process and Analyze Assets for cr-16-071619-cdd5

Disks

<table>
<thead>
<tr>
<th>Id</th>
<th>Instance</th>
<th>Volume</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>snap-a3d00be1</td>
<td>i-22990038e</td>
<td>vol-43c159c9</td>
<td></td>
</tr>
<tr>
<td>snap-acd403a</td>
<td>i-d0000105</td>
<td>vol-3ec55bea</td>
<td></td>
</tr>
<tr>
<td>snap-5263ce060</td>
<td>i-46b001ee4</td>
<td>vol-71cf49f</td>
<td></td>
</tr>
<tr>
<td>snap-cdd7d9f9</td>
<td>i-70c77f9ad</td>
<td>vol-1f62f73f</td>
<td></td>
</tr>
</tbody>
</table>
Video Tour ThreatResponse Disk Analysis

Video Demonstration
Securing your AWS Infrastructure

- CloudTrail - 2 Failures
  - General Checks: Failed 1 of 1
    - No multi-region trails.
  - Checked 1 Resources: Found 1 to be investigated.
    - krug-uswest-1-cloudtrail
      - Log archival is not enabled.

- CloudWatch - 1 Failure
  - General Checks: Failed 1 of 1
    - 0 Billing alerts are enabled in CloudWatch.
  - Checked 0 Resources: Found 0 to be investigated.

+ IAM - 15 Failures

+ S3 - 5 Failures
<table>
<thead>
<tr>
<th>S3 Checks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Versioning</td>
</tr>
<tr>
<td>Logging</td>
</tr>
<tr>
<td>Open Permissions</td>
</tr>
</tbody>
</table>
# IAM Checks

<table>
<thead>
<tr>
<th></th>
<th>MFA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rotated Credentials</td>
</tr>
<tr>
<td></td>
<td>Administrator Access Policy</td>
</tr>
</tbody>
</table>
Other Checks

VPCs: Flow Logging

CloudTrail: MultiRegion & validation
## Other Checks

<table>
<thead>
<tr>
<th>Disable access keys on the root account</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ensures an IAM role exists</td>
</tr>
<tr>
<td>Cloudwatch Billing Alerts</td>
</tr>
</tbody>
</table>
AWS Trusted Advisor

Security Checks

- **Security Groups - Specific Ports Unrestricted**
  - Checks security groups for rules that allow unrestricted access (0.0.0.0/0) to specific ports.
  - 33 of 47 security group rules allow unrestricted access to a specific port.

- **IAM Use**
  - Checks for your use of AWS Identity and Access Management (IAM).
  - At least one IAM user has been created for this account.

- **MFA on Root Account**
  - Checks the root account and warns if multi-factor authentication (MFA) is not enabled.
  - MFA is enabled on the root account.
AWS Config

AWS Config - $.003 per configuration item

AWS Config Rules - $2 per rule per month for $20,000 evaluations.
Review of Tools

Margarita Shotgun

AWS-IR Cli

ThreatResponse WebApp

ThreatPrep Advising
Brief: What's going on in Open Cloud Security
Evolve your understanding through experimentation!
Don't!

Wait to try out some of these tools

---

Do!

Have a test environment

Security simulations

IR Game Days
What does that even mean?

Test environments

Build a Continuous Integration Culture

Have separate AWS accounts for Dev, Test, etc...

Use consolidated billing.
Mixed Environment

AWS

www test

www test

www prod

VPC
Separation

- **AWS**
  - **www test**

- **VPC**
  - Consulting firm + your engineers

- **AWS**
  - **www test**

- **VPC**
  - Engineers you outsourced

- **AWS**
  - **www prod**

- **VPC**
  - Engineers that work for you

- **AWS**
  - Consolidated Billing
What do all these engineers have in common?
About Security Simulation

1. Basically you fake a hack or two.
2. Some percentage of employees know.
3. Some percentage don't know.
4. Process it like a real exercise.
PSA : Tell Amazon if you do these.

AWS Policies do allow for security simulation and IR game days. They just ask that you let them know in advance.
Other Projects in the Space
Simian Army
Captiol One Cloud Custodian

https://github.com/capitalone/cloud-custodian

Rule Engine

Can create lambda functions for you

Around since April 2016
## Feature Comparison

<table>
<thead>
<tr>
<th>Item</th>
<th>Incident Handling</th>
<th>Forensics</th>
<th>Compliance</th>
<th>Continuous Monitoring</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS IR</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Threat Prep</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Marganta Shotgun</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Security Monkey</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Cloud Custodian</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Future of the Tools
OUR TEAM

Andrew Krug
Creator ThreadResponse @andrewkrug

Alex McCormack
Creator ThreadResponse @amccormack

Joel Ferrier
Creator Marginata Shodgun @joelferrier

Jeff Parr
Front End Guru @jpar

Join Us!
Become a contributor today!

This could be you.
Making open source software is fun.
Thanks Amazon Web Services

Don Bailey
Zack Glick
Henrik Johansson
Where to get the software?

http://www.threatresponse.cloud

Releasing soon!

Signup for a notification.
Q&A

http://www.threatresponse.cloud