IR in the Cloud (AWS)

Jonathon Poling
SecureWorks
Principal Consultant
Agenda

• Overview of AWS Security Model
• Logging in AWS
• Monitoring/Alerting
• Environment Preparation
• Targeted Response
• Log Collection
• System/Data Analysis

Setup for Success
AWS **SHARED** Responsibility Model

**CUSTOMER**
- RESPONSIBLE FOR SECURITY "IN" THE CLOUD
  - CUSTOMER DATA
  - PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
  - OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
    - CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
    - SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
    - NETWORK TRAFFIC PROTECTION (ENCRYPTION/INTEGRITY/IDENTITY)

**AWS**
- RESPONSIBLE FOR SECURITY "OF" THE CLOUD
  - COMPUTE
  - STORAGE
  - DATABASE
  - NETWORKING
    - AWS GLOBAL INFRASTRUCTURE
    - REGIONS
    - AVAILABILITY ZONES
    - EDGE LOCATIONS
Why Should We Care?

A snippet from the recent OneLogin attack...

Method of attack

Our review has shown that a threat actor obtained access to a set of AWS keys and used them to access the AWS API from an intermediate host with another, smaller service provider in the US. Evidence shows the attack started on May 31, 2017 around 2 am PST. Through the AWS API, the actor created several instances in our infrastructure to do reconnaissance. OneLogin staff was alerted of unusual database activity around 9 am PST and within minutes shut down the affected instance as well as the AWS keys that were used to create it.

https://www.onelogin.com/blog/may-31-2017-security-incident
Repeat After Me...

Security is MY responsibility in AWS.
Security is MY responsibility in AWS.
Security is MY responsibility in AWS.
Security is MY responsibility in AWS.
Security is MY responsibility in AWS.
Security is MY responsibility in AWS.
Logging in AWS – Log Sources

- CloudTrail
  - API Activity Logs

- CloudWatch
  - System Performance Monitoring/Alerting Logs
  - System (OS)/Application Log Monitoring/Alerting

- Config
  - Resource Configuration Logs

- S3
  - Bucket Access Logs
  - Data/Object Access Logs
  - Server Access Logs
Logging in AWS – CloudTrail

• Logs AWS API calls (actions taken in/on your account)
• Provides visibility into your user/resource/env activity
• These logs (help to) identify…
  • Which users and accounts performed various AWS calls/functions/actions
  • When/where the calls occurred (date/time, region)
  • Who called/made them (region/location, source IP, user agent, user name, etc.)
  • Etc.
Logging in AWS – CloudWatch

• System Performance Metrics
  • Real-time performance metrics for search, graphing, and alarms
    • EC2 Instance, Volume, and RDS DB enabled by default
  • Sent every 15 minutes (default)
  • Various high-level metrics for monitoring system performance
    • CPU, Disk I/O, Network I/O, Status Checks

• Log Monitoring
  • System/Application logs + CloudTrail logs
Logging in AWS – Config

- Logs a multitude of resource configurations
  - EC2 Instances, VPC, EBS Volumes, Security Groups, etc.

- Captures periodic resource configuration snapshots

- Allows you to...
  - Capture snapshots of resources in time
  - Compare (diff) snapshots
  - Retrieve configuration history of resources
    - When was a resource created/modified/changed?
    - Who changed it and how?
Logging in AWS – S3

- **Bucket-Level (aka Management Event) Logging**
  - Logs all Delete/Get/Put Bucket activities

- **Additional Logging**
  - **Object-Level (aka Data Event) Logging**
    - Logs object-specific activities like GetObject, PutObject, DeleteObject with useful date/time, AWS account, role, source IP, etc. info

- **Server Access Logs**
  - Resembles and provides more “HTTP-like” access logs for those who prefer
Monitoring / Alerting - General

• What Should I Monitor in General?
  • General Activities
    • Environment Enumeration/Recon
      • General API’s: Get*/List*
    • Resource/Data/Event Collection
      • General API’s: Get*/Describe*/List*/Lookup*
    • Resource Creation/Modification/Deletion
      • General API’s: Delete*/Disable*/Remove*
  • Log Tampering/Modification
    • General API’s: Stop*/Delete*/Update*/Set*
Monitoring / Alerting - Specifics

• Quadfecta of Monitoring
  • Config + CloudTrail + CloudWatch + Lambda

• What Specifically Should I Monitor and Alert on?
  • Lots of services, API types/calls, and nuances

So, here’s my best advice...
1. Know your business, services, systems, and network.
2. Read the AWS docs.
3. Think/play/test like an attacker.
4. Try to catch yourself.
Environment Preparation

• Isolation Environment
  • Create Isolation VPC
    • Set up VPC with route only to internet
      • Ensure zero communication is allowed betwixt this isolation VPC and others*
        • *Unless you have a REALLY GOOD reason
  • Set Up Network Logging
    • Enable VPC Flow Logs
Environment Preparation

- DFIR Setup
  - Create “Responder” Account(s)
  - Create IAM “Responder” Role
    - Can be Read-Only Admin or a crafted policy
      - Depends what you’d like to do from the account
  - Create Dedicated S3 Bucket(s) for Collected Data
    - Setup Tip: Create different “Folders” for Logs, Disk, and Memory
    - Lockdown Tip: Apply Bucket-level permissions to only allow access to Responder Role
Environment Preparation

• DFIR Setup (Cont.)
  • Create Limited-Access Security Group for Instance(s)
    • Restrict access by service/port
      • Allow only SSH / RDP
    • Restrict access by source IP
      • Use specific IP(s) from which the analyst(s) will be connecting
Environment Preparation

• DFIR Setup (Cont.)
  • Create Instance(s)
    • Assign “Responder” Role
  • Create 2 Volumes
    • Root (min size for OS files/data)
    • Data (min size needed for local data analysis)
  • Instrument with DFIR Tools
    • Add all your fave DFIR + awscli tools
      • #ProTip: FOSS like a BOSS
      • Commercial tools (licenses) can get tricky
Targeted Response

• Collect Memory/Disk Image
  • Collect Memory Image
    • Collection methodology same as physical collection (LiME, FTK, etc… womp, womp)
    • Can be automated with Managed Instances / Run Command
  • #ProTip: Image directly to S3 Bucket
• Collect Disk Image
  • Snapshot all pertinent Volume(s) on affected Instance(s)
Targeted Response

• Isolate / Monitor Instance(s)
  • Move affected Instance(s) to Isolation VPC
    • Immediate containment and network logging
  • Monitor/collect network IOC’s
  • Create CloudWatch Alerts across enterprise for IOC’s

* Swap the order of the disk/memory collection and system isolation based on the situation and assessed risk(s)
Log Collection

• Super Simple Collection
  • CloudTrail is essentially the single log source for API Calls, Config Logs, CloudWatch Logs, etc.
  • If setup correctly, all will be logged to single S3 Bucket in consistent format (typically JSON)
  • Simply copy all relevant logs from the S3 Bucket to local volume or read directly from S3
Analysis – Disk/Memory

- Disk
  - Create Volume(s) from Snapshot
  - Attach Volume(s) to Analysis Instance
  - Analyze as you would any physical attached disk
  - #ProTip: Volumes created from Snapshots may perform at less than 50% of expected performance level until initialized

- Memory
  - Read image directly from S3 or copy to local analysis volume
  - Analyze as you would any other memory image
Analysis – Logs

• Log Analysis Options
  • Manual (Command-Line) Analysis
    • For the CL gluttons in some of us…
    • Mass <gunzip,zcat,sift,jq> * with your CL kung-fu
  • #ProTip: Use *jq* - it is your best friend for JSON logs
• References:
  • Download/Usage: [https://stedolan.github.io/jq/](https://stedolan.github.io/jq/)
  • Try it Out Live: [https://jqplay.org/](https://jqplay.org/)
Analysis – Logs

JQ Examples

$ cat log.json | jq '.Records[].userIdentity.userName'
  >> "Alice"

$ cat log.json | jq '.Records[0].sourceIPAddress'
  >> "205.251.233.176"
Analysis – Logs

• Log Analysis Options (Cont.)
  • Assisted Analysis (courtesy* of AWS)
    • Extract into ES for analysis
      • Use Kinesis to send logs to your ES cluster
  • Use Athena to SQL Query logs
    • Create Table + Data Structure
    • Point it at specific S3 Bucket or Logs folder(s)
    • SQL Query to your heart’s (dis)content
  • Currently limited to US N.VA, Ohio, and Oregon regions
Analysis – Logs

- What To Do/Look For (General Overview)
  - CloudTrail
    - Identify/extract all API calls associated with known compromised Instances, Roles, and Users
  - CloudWatch
    - Review any specific alerts/alarms
    - Identify Resource Anomalies/Spikes
    - Review additional logs collected by CloudWatch agent (if instrumented)
Analysis – Logs

- What To Do/Look For (General Overview)
  - Config
    - Compare auto (and manually) generated Snapshots
    - Review configuration history for known involved resources (Instances, IAM Accounts, Security Groups, ...)
  - Identify anomalous environment alterations
    - Created/Modified/Deleted Users, Roles, VPC’s, Instances, ...
  - S3
    - Review Bucket and Object accesses
      - GET*/PUT*/DELETE* API Calls
What Did We Learn?

• AWS provides a LOT of native and awesome capabilities to leverage for DFIR
  • I’ll take a typical default AWS deployment any day over a typical default on-premises deployment. Yeah, I said it.

• There’s a non-trivial ramp-up to operating with scalpel-like precision in the Cloud (AWS), BUT...

• At a certain point, DFIR methodologies for AWS and on-premises converge
  • Being great at the latter is a great start for the former
  • Why not be awesome at both?
What Did We (Maybe) Learn?

• A couple new words*...
  • Quadfecta
    • “A set of four wins at related events; a set of four events.”
  • Betwixt
    • “(archaic) another word for between”

• This guy has a REALLY hard time consolidating so much material into a 40 minute session
  • Keep an eye out for the next presentations/posts on targeted log configuration and analysis tips/tricks

*Wiktionary says so
$ for i in claps; do success(); done

>> Segmentation Fault

Jonathon Poling
SecureWorks
@JPoForenso
http://ponderthebits.com
✓ DFIR+Linux+Mac+AWS Content