Logging, Monitoring, and Alerting in AWS

The TL;DR

Jonathon Poling
Managing Principal Consultant
Secureworks
About Why Me?

• A little bit about me… blah blah blah

• Why Me? this though
  • Spent WAY TOO MUCH TIME…
    • Reading AWS Docs + Building AWS Cheat Sheets + Hoarding Community Tools
  • AWS SME for Secureworks
  • Developed Secureworks’ AWS IR Service Line
  • Help SMB through Fortune 10 Customers…
    • Intelligently Configure/Instrument Their Environments
    • Protect Their Infrastructure
    • Effectively Respond to Incidents
Agenda

• The AWS Security Model
• Logging in AWS
  • Main Log Sources
  • Setup/Configuration
• Monitoring
  • Generalities and Specifics
  • Tooling (Native + 3rd-Party)
• Alerting
  • Instrumenting for Success
  • Tooling (Native + 3rd-Party)
• Basking in Softball Questions
“Why We Moved To AWS”

Need a break?

AWS Glue
Does It All
Creates
ETL code
for you. !!!

Stock Picture Source: https://twitter.com/awscloud/status/968273839389061120
FALSE
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
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 – Main Sources

<table>
<thead>
<tr>
<th></th>
<th>What</th>
<th>Delay</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudTrail</td>
<td>API calls</td>
<td>15 minutes</td>
<td>S3</td>
</tr>
<tr>
<td>CloudWatch Events</td>
<td>Subset of API calls for changes</td>
<td>Real-time</td>
<td>CloudWatch Events</td>
</tr>
<tr>
<td>VPC</td>
<td>Network flows</td>
<td>15 minutes</td>
<td>CloudWatch Logs</td>
</tr>
<tr>
<td>S3</td>
<td>Bucket access</td>
<td>Hourly</td>
<td>S3</td>
</tr>
<tr>
<td>ELB</td>
<td>Web requests</td>
<td>5 minutes</td>
<td>S3</td>
</tr>
<tr>
<td>CloudFront</td>
<td>Cache requests</td>
<td>Up to 24 hours</td>
<td>S3</td>
</tr>
</tbody>
</table>
Logging in AWS – Setup/Config

CloudTrail

- Your account’s syslog on steroids
- Enabled by Default for 90 days of retention BUT...
  - Each region’s logs are kept ONLY in that region’s bucket (ROYAL PAIN for response)
  - Only “Global” (IAM/STS) service events will be logged across all regions/buckets
- Configure Global/Central Logging to one bucket
  - “Apply Trail to all Regions” config option
- Disable global logging on all other buckets (otherwise you’ll be dealing with dupes)
Logging in AWS – Setup/Config

CloudWatch

• System performance metrics
  • Enabled by default (metrics sent every 15 minutes)
  • Enabling “Detailed Monitoring” will send metrics every 1 minute

• OS/Application Logs
  • Send to CloudWatch via EC2 Systems Manager (SSM) and/or CloudWatch Logs Agent
    • Both require installation of additional agent on each Instance

• Additional stuff you’re also sending (CT, VPC, etc.)
Logging in AWS – Setup/Config

**Config**

- Track Resource “Compliance” against a set of rules
- Easy setup via Console or CLI
- Deliver config logs to SNS Topic and/or S3

**Config Rules**

- Enable various default Config Rules to monitor/alert on configuration changes as they occur or on a schedule
- Create custom rules according to your environment and policies
- AWS Managed Rules provided/enabled by default

**Now with Multi-Account Multi-Region Data Aggregation**
Logging in AWS – Setup/Config

Config (Cont.)

• (BONUS) Software Monitoring
  • Monitor/record software inventory/changes
  • Requires Instances to be configured as “Managed Instances”
Logging in AWS – Setup/Config

S3

- Enable MFA Delete
  - Protect against attacker (and simple accidental) log deletion
- Bucket-Level (aka Management Event) Logging
  - Enabled by Default
- Object-Level (aka Data Event) Logging
  - Configure by enabling “Data Events” in a Trail for a given Bucket
- Server Access Logs (Apache log-ish)
  - Enable via Bucket Configuration options
Logging in AWS – Setup/Config

VPC Flow Logs (i.e. Netflow logs)

• Can be enabled for VPC, VPC Subnet, or Elastic Network Interface (ENI)
  • Enable for anything of which you might even remotely care about the incoming/outgoing traffic
  • Start with (Critical) VPC(s) housing critical server(s)?

• Logged to CloudWatch Logs as a new Log Group with a Stream for each associated ENI
  • Create CW Metric Filters/Alarms for traffic you care about
    • Brute force RDP/SSH attempts
    • Inbound/Outbound traffic to known bad IP’s
    • Large outbound transfers (data exfil alert anyone?)
Logging in AWS – TL;DR

• How Do I Enable These Logs?
  • Basic CloudTrail + CloudWatch enabled by default
  • These + others can (and should) be configured by you

• How Do I Centralize Them?
  • (Easy) Send to/store in dedicated S3 Bucket(s)
  • (Convoluted) CW → Kinesis streams → AWS Lambda fanout

• How Should I Configure Them?
  • Understand your business, system, and network needs
  • Read the AWS docs
  • Configure as appropriate
  • Baseline recommendations on the following slides...
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*/Update*/Set*
Monitoring / Alerting - Specifics

• What Services/Logs Should I Monitor, Specifically?
  • CloudTrail
  • CloudWatch
  • Config
  • EC2
  • IAM (+ Access Advisor)
  • GuardDuty
  • S3
  • STS
  • Trusted Advisor (Basics)
Monitoring / Alerting – Native Tools

• What Tools Can I Use to Monitor/Alert?
  • Native
    • Quadfecta of Monitoring/Alerting
      • Config + CloudTrail + CloudWatch + Lambda
    • Supporting tools
      • Trusted Advisor
      • CloudWatch Agent(s)
      • GuardDuty (New-ish)
        • Automatically monitors VPC Flow + CT + DNS logs
        • Region-specific/limited (because… of course)
        • Only analyzes CT mgmt events, not log data events
Monitoring / Alerting – Native Tools

• Example Native Monitoring/Alerting Infrastructure…

Source: https://aws.amazon.com/answers/account-management/real-time-insights-account-activity/
Monitoring / Alerting - Examples

**CloudWatch Alarms**
- Send notifications in response to certain logs / metrics
- Billing
  - Alarm when billing threshold near/reached
- System Logs
  - Monitor for lateral movement
    - Create Metric + Alarm on RDP connections, PSExec run on system, ...
- EC2
  - Stop, Terminate, Reboot, or Recover an Instance (using CloudWatch Alarm Actions)
- VPC Flow Logs
  - Monitor for network traffic spikes/C2 activity
    - Create Metric + Alarm on known IOC’s - IP/port/protocol/bytes/packets
Monitoring / Alerting - Examples

CloudWatch Events
- Perform actions in response to environment / resource changes (aka Events)
- CloudTrail (with logs sent to CloudWatch)
  - StopLogging or DeleteTrail API call made ->
    - Send notification to security team with caller identity and info
- EC2
  - Unsupported Instance type created ->
    - Lambda function to stop/isolate an Instance
  - Instance Terminated ->
    - Extract info / instance metadata / logs before shutdown
- GuardDuty
  - Configure all Findings to be sent/converted to CW Events
  - Monitor for (and respond to) Findings
Monitoring / Alerting - Examples

CloudWatch Events

- IAM
  - New User created ->
    - Lambda function to disable account until reviewed
    - Gratuitous failed logins, brute force attempts ->
      - Send email to security team, Lambda function to block IP
  - S3
    - Log file(s) deleted ->
      - Send notification/email with associated username/account
- STS
  - Sensitive role assumed ->
    - Lambda function to disable account that assumed role / disable role
Monitoring / Alerting - Examples

CloudWatch Events
- Trusted Advisor
  - Un(der)utilized EC2 Instances ->
    - Send email / notification and Stop Instances
  - Access Key Publicly Exposed ->
    - Delete Access Key and send email notification / summary
- VPC Flow Logs
  - Known malicious IP seen ->
    - Run command to capture memory, lambda function to isolate instance
  - Authentication brute-forcing / password spraying attempts ->
    - Alarm with offending IP(s) and traffic summary
  - Large outbound data transfers ->
    - Alarm on anomalous outbound traffic (data exfil?)
Monitoring / Alerting - Examples

**Config Rules**

- Send SQS notification in response to resource/config changes
  -- AND/OR --
- Trigger rule (Lambda function) to run:
  - Periodically
    - Run IAM MFA validation rule every 24 hours
    - Run check for unused Elastic IP’s every 3 days
  - In response to certain resource changes
    - IAM User Created ->
      - Trigger rule for password policy check
    - EC2 Instance Stopped ->
      - Trigger rule to tag Instance as “Unused”
    - S3 Bucket Created ->
      - Trigger rule to check that bucket logging/versioning is enabled
Monitoring / Alerting – 3rd Party Tools

• What Tools Can I Use to Monitor/Alert?
  • 3rd-Party
    • Cloud-Custodian
    • Cloud-Inquisitor
    • CloudTracker
      • Requires CT logs be loaded into ElasticSearch via Mozilla’s HindSight
    • Security Monkey
    • StreamAlert (ElastAlert)
    • Wazuh
    • ...and more
Monitoring / Alerting – TL;DR

• What Should I Be Monitoring?
  • Everything from (high-level) account changes down to (low-level) API calls from all your existing/used services

• What Tools Should I Use to Monitor?
  • Native tools aplenty
  • 3rd-Party tools aplenty-er
**Monitoring / Alerting – TL;DR**

- **In What Order Should I Set Up Monitoring/Alerting?**
  - Same way you’d build an *actually useful* TI program...
  1. Clearly define your (monitoring) goal(s)
  2. Understand your existing valuable data(sets)/information
  3. Implement and exhaust your native tools and capabilities
     a. If you’re kinda sure about #3, proceed to #4.
  5. Implement additional/3rd-party tools as needed to augment monitoring and reach your intended goal(s)

**Understand ➔ Build ➔ Exhaust ➔ Augment**
The End

Twitter: @JPoForenso
Blog: https://www.ponderthebits.com