AWS Security Monitoring & Compliance Validation From Adobe

Scott Pack

Security Engineer, Adobe

Presenter

- Scott Pack
  - Security Engineer @ Adobe
  - SLC, UT
  - 4 Year Splunker
  - Proudly DQd at 3 Pinewood Derbies

- Agenda
  - Background
  - AWS Security Data Sources
  - Aggregation & Ingest
  - Bit of Analysis

The Background

- Digital Marketing & Analytics
- 55k hosts across 30 sites
- Collection of ~20 admin teams.
  - Different tech stacks, but mostly *nix
- Monitoring Toolset:
  - Netflow, FPC, IDS, Network Transaction

Security Operations At Adobe

- Splunk as a Core Service
  - Used for all logs: application, network, host, etc
- Security Engineering: Own the data sources
  - Set up systems that feed Splunk
- Security Operations: SplunkES Analysis & Investigation
  - Consume the data

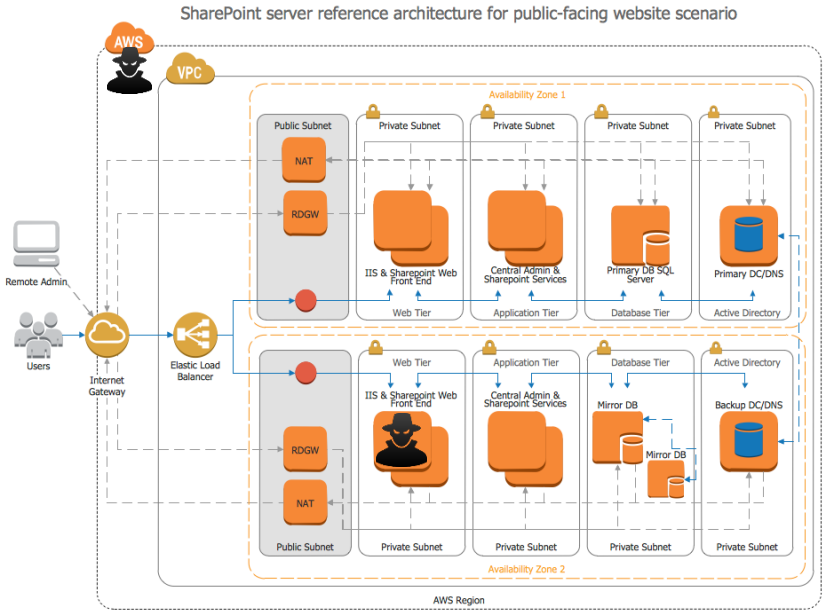
Shifting To AWS

- Lots of accounts … > 200
- Dozens of teams, thousands of instances
- Missing data to:
  - Detect/respond to incidents
  - Making assurances to Compliance
- We received a mandate: Fix this
  - Get whatever visibility you can
  - Minimize risk of operations impact
  - Be cost sensitive

AWS Security Incidents? Wut?

- AWS Account Compromise:
  - Baddie interacts w/ AWS as an authenticated user

- Host compromise
  - Baddie has some control of a host



Make Our Lives Easier:

- Follow the same model: Data -> Splunk ES -> SOC
- Don’t juggle hundreds of AWS API keys
- Out-of-band monitoring
- Quick setup
- Reduce future need to redeploy
- Keep it to AWS Native data sources

Data Sources

CloudTrail
API Usage & Logging

Trusted Advisor
Security Practice Checks

Config
Account Configuration & Inventory

VPC FlowLogs
Virtual Interface Connectivity

Identity & Access Management
Credential Report

ELB Access Logs
Load Balancer Logging

Data Examples

CloudTrail


```
{ [-]
   awsRegion:  sa-east-1
   eventID:  51da0e8a-8dd3-4e3a-a54e-bfaa0433ad49
   eventName:  RunInstances
   eventSource:  ec2.amazonaws.com
   eventTime:  2016-08-19T00:52:37Z
   eventType:  AwsApiCall
   eventVersion:  1.04
   recipientAccountId:  555555555555
   requestID:  76c65b6a-32be-4a0f-8362-313d0c2a7ef9
   requestParameters: { [+]
  }
   responseElements: { [+]
  }
   sourceIPAddress:  1.2.3.4
   userAgent:  terraform/0.7.0
   userIdentity: { [+]
  }
}
```


Config


```
{ [-]
   ARN:  arn:aws:ec2:sa-east-1:55555555555:instance/i-ff99042e
   availabilityZone:  sa-east-1c
   awsAccountId:  5555555555555
   awsRegion:  sa-east-1
   configuration: { [+]
  }
   configurationItemCaptureTime:  2016-08-19T00:58:46.489Z
   configurationItemStatus:  ResourceDiscovered
   configurationItemVersion:  1.2
   configurationStateId:  1
   configurationStateMd5Hash:  f353b859017bc719249b9cd102fba0d5
   relatedEvents: [ [+]
  ]
   relationships: [ [+]
  ]
   resourceCreationTime:  2016-08-19T00:52:10.000Z
   resourceId:  i-ff99042e
   resourceType:  AWS::EC2::Instance
   snapshot_id:  69e3a29c-a67a-4b6d-b22f-bce6404bd620
   snapshot_time:  1471626526.634382
   supplementaryConfiguration: {
  }
   tags: { [+]
  }
}
```


Credential Report


```
{ [-]
   access_key_1_active:  false
   access_key_1_last_rotated:  N/A
   access_key_1_last_used_date:  N/A
   access_key_1_last_used_region:  N/A
   access_key_1_last_used_service:  N/A
   access_key_2_active:  false
   access_key_2_last_rotated:  N/A
   access_key_2_last_used_date:  N/A
   access_key_2_last_used_region:  N/A
   access_key_2_last_used_service:  N/A
   arn:  arn:aws:iam::555555555555:root
   awsAccountId:  555555555555
   cert_1_active:  true
   cert_1_last_rotated:  2013-08-02T18:09:34+00:00
   cert_2_active:  false
   cert_2_last_rotated:  N/A
   mfa_active:  false
   password_enabled:  not_supported
   password_last_changed:  not_supported
   password_last_used:  2016-07-27T18:39:13+00:00
   password_next_rotation:  not_supported
   user:  <root_account>
   user_creation_time:  2013-05-14T18:43:39+00:00
}
```


ELB Access Logs

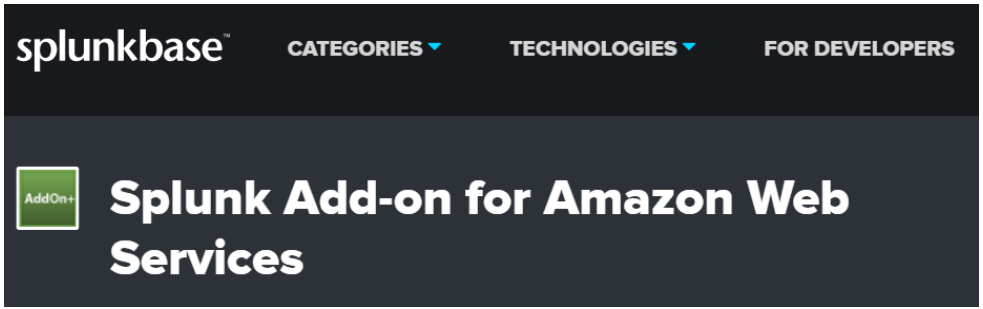

```
2016-08-19T17:00:27.196195Z DCO-ATS-00 1.2.3.4:60073 10.92.48.57:8080 0.000048 0.001157 0.000018 200 200 0 7
"GET http://yur_website_stuff.com:80/index.php HTTP/1.1" "Mozilla/5.0 (iPhone; CPU iPhone OS 9_3_4)"
```


VPC Flows


```
2 555555555555 eni-81a8c2ea 172.31.11.13 1.2.3.4 56139 8081 6 2 120 1471539837 1471539956 ACCEPT OK
```

OK, So This?

- Has input types for:
  - Config Snapshots
  - Config Rules
  - CloudTrail
  - CloudWatch Logs
  - ELB Access Logs
  - S3 Buckets



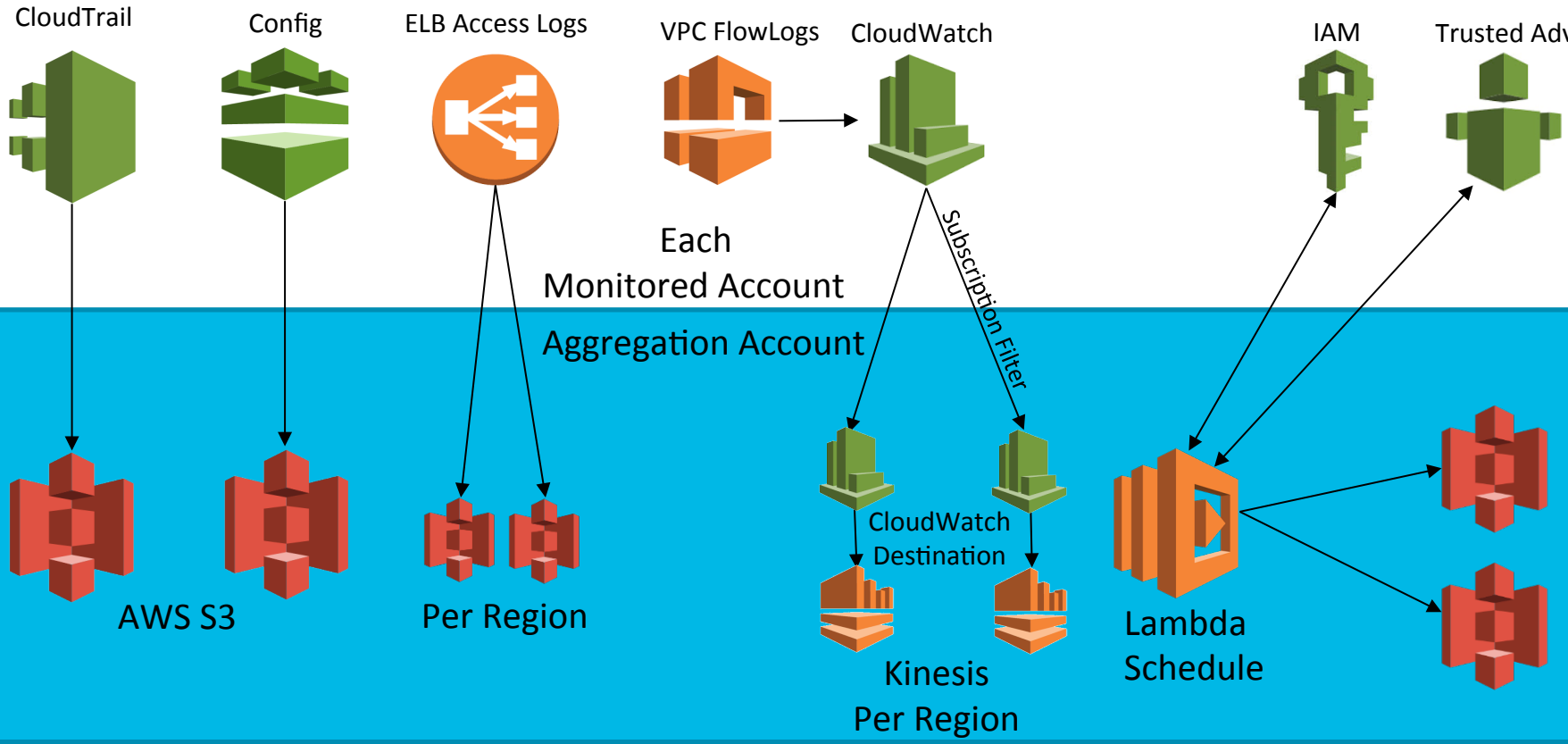
- But…
  - Input Stanza Explosion
    - Account x sourcetype x (region)
    - ~ 28 Inputs per account
  - API Keys for each account

Cross-Account Authentication

- IAM Users
  - Use API Keys directly
- Roles
  - AWS Security Token Service
  - Can be “Assumed” by a specified Principal
    - Principal: AWS User, Account, Service, Other Role
  - Authenticate to an Aggregation Account user
    - Assume the cross-account role
    - Retrieve temporary access keys
    - Make calls with temporary keys

- http://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_cross-account-with-roles.html

Aggregation



Collection Plumbing: S3

- S3 Buckets:
  - ELB (1 per region)
    - Permit PutObject from AWS ELB IAM Roles
  - Config
    - Permit PutObject from config.amazonaws.com
  - Config Parsed
  - CloudTrail
    - Permit PutObject from cloudtrail.amazonaws.com
  - Trusted Advisor Results
    - Permit PutObject from Lambda Execution IAM role

AWS ELB Account IDs for Log Delivery: http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/enable-access-logs.html#attach-bucket-policy

Collection Plumbing: Rest of it

- Aggregation AWS Account
- Kinesis Stream:
  - 1 Per region
- CloudWatch LogDestinations
  - 1 Per region
  - Directs to region-local kinesis stream

Registration

Registration

CloudFormation

Inputs:
Description
Jira Queue

Resources:
Config Role
FlowLogs Role
SecEng Role

Role's Done!

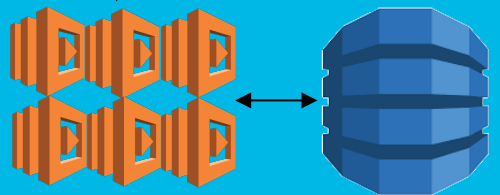
SNS Notification

Registration Lambda

Registration DynamoDB

Setup & Retrieval

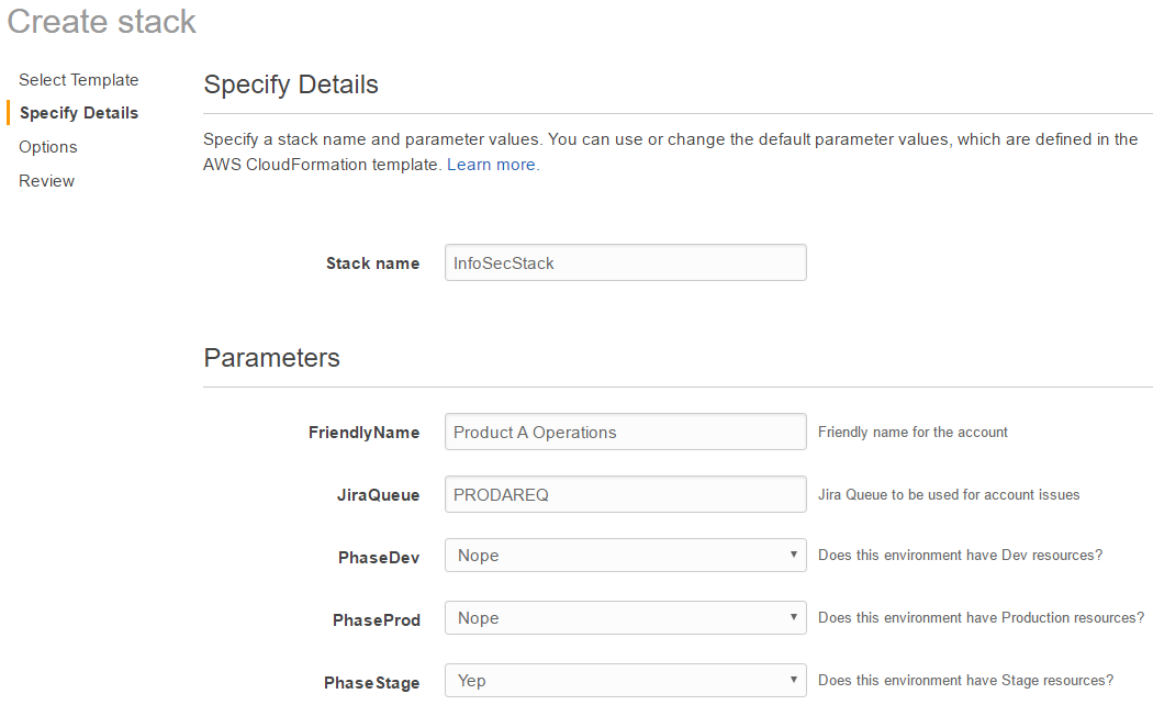
Assumed Role



Daily Setup Functions

Registration DynamoDB

Enrollment Via Web UI

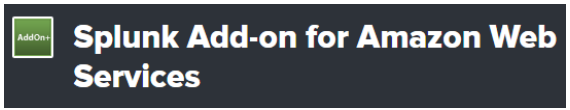


Collection

+

Analysis

Splunk App



- Input Methods: S3
- Input Sourcetypes: CloudTrail, VPC Flows, ELB Access Logs

- Parsing Handler: GZIPMessageHandler (Thanks Damien!)

Aggregation reduces amount of Splunk inputs: 26 Total Inputs

- S3: 14
- Kinesis Inputs: 10
- Additional Logging: 2

Currently running on a dedicated Heavy Forwarder.

- If needed, split regions to different forwarders.

Sourcetypes, Lookups, And Other Fun

- Sourcetypes: Cheated off the Splunk App for AWS.
  - Set json KV format and check line-breaks
- Use HTTP Event Collector to dump DynamoDB account registrations
  - Scheduled lookup-generating search
  - Every event has the account ID somewhere in it (Almost).
- Tagging into Enterprise Security data models
  - ELB Access Logs & VPC Flows right out of the box

Getting Dashboard Approval

- GETTING DASHBOARD APPROVAL

“Gotchas”

Data Frequency/Latency

Config	• Daily Snapshots
Trusted Advisor	• Daily Snapshots
CloudTrail	• 5-8 minute latency
ELB Access Logs	• 5-10 minute latency
VPC Flow Logs	• 5-10 minute latency

Splunk Gotchas:

- Kinesis Modular Input
  - Can chew up memory
  - Increase what it gets:
    - /opt/splunk/etc/apps/kinesis_ta/bin

    java_args = [JAVA_EXECUTABLE, "-classpath",CLASSPATH**,"-Xms512m","-Xmx512m",**
    "-Dsplunk.securetransport.protocol="+SECURE_TRANSPORT,JAVA_MAIN_CLASS]

- Config Snapshots are jsonormous
  - Use lambda to split up the resources

AWS Gotchas:

- Still no packet-level visibility
- ELB Permission Granularity Restrictions
  - ModifyAttributes
- Keep an eye on capacity. Watch:
  - DynamoDB Reads
  - Kinesis Shard Usage

Where We're At Right Now

- 40 AWS accounts currently enrolled
- 500-800 GB/day
- Haven't broken any accounts yet!
- Finding more data sources
  - Config Rules
  - Inspector
- Automated our AWS security policy audit
- Written a handful of Splunk Enterprise correlation rules
  - Actioned by SOC
- Automated Jira ticketing for remediation

Questions?

Contact:
scottjpack@gmail.com
github.com/scottjpack
Twitter: @scottjpack

THANK YOU

splunk>

Permissions

```
Role: SecEngRole
Permits assume from arn:aws:iam::            :role/SecEngCrossAccountRole
Managed Policies:
    ReadOnlyAccess
    AWSSupportAccess
ELB:
    elasticloadbalancing.DescribeLoadBalancerAttributes
    elasticloadbalancing.DescribeLoadBalancers
    elasticloadbalancing.ModifyLoadBalancerAttributes
iam.PassRole (ConfigIamRole)
EC2:
    ec2.DescribeRegions
    ec2.CreateFlowLogs
    ec2.DeleteFlowLogs
    ec2.DescribeFlowLogs
    ec2.DescribeVpcs
CloudWatch Logs:
    logs.CreateLogGroup
    logs.DeleteSubscriptionFilter
    logs.PutSubscriptionFilter
    logs.PutRetentionPolicy
    logs.DeleteRetentionPolicy
inspector.*
config.*
```

DynamoDB: Account Registration Item

{ [-]
  DevPhaseOutput: Yep
  InspectorRoleARN: arn:aws:iam::555555555555:role/InfoSec-InspectorIamRole-1WPVBFHJ3CQM1
  ProdPhaseOutput: Yep
  StagePhaseOutput: Yep
  account_id: 55555555555
  config_pull_enable: true
  config_role_arn: arn:aws:iam::555555555555:role/InfoSec-ConfigIamRole-1CCXRZ8SN2IL5
  description: CampaignOps
  elb_access_log_enable: true
  flowlogs_role_arn: arn:aws:iam::555555555555:role/InfoSec-FlowLogsIamRole-7R1QLDHRXS1F
  jira_queue: CPGNTEAM
  role_arn: arn:aws:iam::555555555555:role/InfoSec-SecEngRole-9W6HAJ8SNOEK
  trusted_advisor_collect: true
  vpc_flow_logs: true
}