Going Serverless: Security Outside the Box
Has cost, staffing, time, or money prevented you from deploying a security tool you needed?
Traditional Approach: Servers

- Application
- Scaling
- Runtime
- Operating System
- Virtualization
- Hardware
Traditional Approach: VMs
You don’t need servers anymore
Serverless: Thinking Outside The Box
@rule(logs=['cloudwatch:events'],
    matchers=[],
    outputs=['aws-s3:sample-bucket',
             'pagerduty:sample-integration',
             'slack:sample-channel'],
    req_subkeys={'detail': ['userIdentity', 'eventType']})

def cloudtrail_root_account_usage(rec):
    ...
    author: airbnb_csirt
    description: Root AWS credentials are being used;
                 This is against best practice and may be an attacker
    ...

    return (rec['detail']['userIdentity']['type'] == 'Root' and
            rec['detail']['userIdentity'].get('invokedBy') is None and
            rec['detail']['eventType'] != 'AwsServiceEvent')
rule hacktool_macos_keylogger_caseyscarborough
{
  meta:
  description = "A simple and easy to use keylogger for macOS."
  reference = "https://github.com/caseyscarborough/keylogger"
  author = "@mimeframe"
strings:
  $a1 = "/var/log/keystroke.log" wide ascii
  $a2 = "ERROR: Unable to create event tap." wide ascii
  $a3 = "Keylogging has begun." wide ascii
  $a4 = "ERROR: Unable to open log file [...]" wide ascii
condition:
  2 of ($a*)
}
1B+ logs analyzed/day
1TB+ data/day

50K+ files scanned/day
2TB+ files analyzed/week
Writing a Serverless App

Scaling
Securing
Deploying
Monitoring
Sharing
Verizon partner data breach exposes millions of customer records
Accessed through an unprotected Amazon S3 storage server

Time Warner Cable vendor leaks 4 million records containing customer data

Security
Leaky S3 bucket sloshes deets of thousands with US security clearance
Bunch of resumés citing secret government work exposed
**AWS Lambda: FaaS**

**Step 1:** Write code (Python, Java, C#, NodeJS)

**Step 2:** Upload to AWS Lambda

**Step 3:** Invoke the function
import boto3

s3_client = boto3.client('s3')

def lambda_handler(event, context):
    # Get the s3 bucket/object info from the event
    for record in event['Records']:
        bucket = record['s3']['bucket']['name']
        key = record['s3']['object']['key']

        # Define the list of insecure ACLs
        insecure_acl_list = [
            'http://acs.amazonaws.com/groups/global/AuthenticatedUsers',
            'http://acs.amazonaws.com/groups/global/AllUsers']

        # Get the object ACL
        object_acl = s3_client.get_object_acl(Bucket=bucket, Key=key)

        # Analyze the ACL to find global objects
        for grant in object_acl['Grants']:
            grantee = grant.get('Grantee', [])
            if ('URI' in grantee and
                grantee['URI'] in insecure_acl_list):

                # Print Object information to the console
                print('Warning, insecure S3 object detected!')
                print('Bucket: {} 
Key: {}' .format(bucket, key))
A Basic Example

Detect Misconfigured AWS S3 Buckets

START RequestId: 42b9d96c-b9d8-11e7-a5b5-ab2c2f892fa Version: $LATEST

Warning, insecure S3 object detected!

Bucket: jack-insecure-bucket-test

Key: logo.png

END RequestId: 42b9d96c-b9d8-11e7-a5b5-ab2c2f892fa

REPORT RequestId: 42b9d96c-b9d8-11e7-a5b5-ab2c2f892fa Duration: 204.94 ms
AWS Lambda: Process S3 Events
Writing

Scaling a Serverless App

Securing
Deploying
Monitoring
Sharing
Linking Serverless Applications
Linking Services
AWS Lambda to AWS Lambda

S3 Analyzer
Invoke
Notify
Fix ACL
S3 Remediator
Linking Services

**AWS SQS:** Serverless Queue

- **S3 Analyzer**
- SQS Queue
- **S3 Remediator**

- **Notify**
- **Fix ACL**
- **Send**
- **Receive**
Real-Time Analysis
Scaling Serverless

Real-Time Analysis
Scaling Serverless

Real-Time Analysis
AWS Kinesis: Serverless Real-Time Streaming

Clients → Kinesis Stream → ShardId-001 → Lambda
               ShardId-002 → Lambda
               ShardId-003 → Lambda
               ShardId-004 → Lambda
               ShardId-005 → Lambda
Taking Action with Serverless
Taking Action

**AWS SNS: Serverless Messaging**
Taking Action

**AWS SNS:** Serverless Messaging

Unauthorized access detected

- **Severity:** Critical
- **Opened:** 5:15 PM (6h ago)
- **Current Status:** Triggered
- **Integration:** API

**Details**

**Summary**

Unauthorized access detected

**Custom Details**

<table>
<thead>
<tr>
<th>record</th>
<th>username</th>
<th>src_ip</th>
<th>dst_ip</th>
<th>host</th>
<th>role</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>alice</td>
<td>10.3.15.1</td>
<td>10.6.0.2</td>
<td>web-01</td>
<td>web-server</td>
<td>1508877426</td>
</tr>
</tbody>
</table>
## Serverless Component Summary

<table>
<thead>
<tr>
<th>Kinesis Streams</th>
<th>Real-time streaming data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lambda</td>
<td>Compute</td>
</tr>
<tr>
<td>S3</td>
<td>Object storage</td>
</tr>
<tr>
<td>SNS</td>
<td>Pub/sub messaging</td>
</tr>
<tr>
<td>SQS</td>
<td>Message queueing</td>
</tr>
<tr>
<td>Athena</td>
<td>SQL queries over S3</td>
</tr>
<tr>
<td>DyanmoDB</td>
<td>NoSQL database</td>
</tr>
</tbody>
</table>
Lambda Containers
Service 1 Policy

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["s3:GetObjectAcl"],
            "Resource": "arn:aws:s3::<bucket-name>/*"
        },
        {
            "Effect": "Allow",
            "Action": ["sqs:SendMessage"],
            "Resource": "arn:aws:sqs::<region>::<acct>::<queue>"
        }
    ]
}
```
Key Management Service (KMS)

Systems Manager Parameter Store

Secrets Management

Encrypted Parameter Store → Get API Token → Lambda → Make Request → 3rd-party API
Deploying a Serverless App
Infrastructure As Code
Terraform / CloudFormation

S3 -> Notification -> Lambda
resource "aws_s3_bucket" "data_store" {
  bucket = "my_bucket.${var.region}"
}

resource "aws_lambda_function" "analyzer" {
  function_name = "data_store_analyzer"
  role          = "${aws_iam_role.analyzer_role.arn}"
  handler       = "main.analyze"
  runtime       = "python3.6"
  filename      = "analyzer.zip"
}

# Configure S3 bucket to notify analyzer Lambda for new objects
resource "aws_s3_bucket_notification" "analyze_new_objects" {
  bucket = "${aws_s3_bucket.data_store.id}"
  lambda_function {
    lambda_function_arn = "${aws_lambda_function.analyzer.arn}"
    events              = ["s3:ObjectCreated:*"]
  }
}
resource "aws_s3_bucket" "data_store" {
  bucket = "my_bucket.${var.region}"
}

resource "aws_lambda_function" "analyzer" {
  function_name = "data_store_analyzer"
  role          = "${aws_iam_role.analyzer_role.arn}"
  handler       = "main.analyze"
  runtime       = "python3.6"
  filename      = "analyzer.zip"
}
resource "aws_s3_bucket" "data_store" {
  bucket = "my_bucket.${var.region}"
}

resource "aws_lambda_function" "analyzer" {
  function_name = "data_store_analyzer"
  role          = "${aws_iam_role.analyzer_role.arn}"
  handler       = "main.analyze"
  runtime       = "python3.6"
  filename      = "analyzer.zip"
}

# Configure S3 bucket to notify analyzer Lambda for new objects
resource "aws_s3_bucket_notification" "analyze_new_objects" {
  bucket = "${aws_s3_bucket.data_store.id}"
  lambda_function {
    lambda_function_arn = "${aws_lambda_function.analyzer.arn}"
    events              = ["s3:ObjectCreated:*"]
  }
}
Benefits
of Infrastructure as Code

Protected by code reviews

Complex infra expressed as small repeatable parts

The entire application can be shared with others

Easily create (or destroy) the entire infrastructure
$ terraform apply -auto-approve=false

...
Versioning and Aliasing

- Test
- Stage
- Prod

- $LATEST
- v2
- v1
Writing
Scaling
Securing
Deploying

Monitoring a Serverless App
Sharing
Types of Errors

- Insufficient Permissions
- Timeouts
- Out of Memory
- Exceptions
Error Handling

2 Different Invocation Models

Asynchronous - Errors Are Automatically Retried

Dead Letter Queue

Synchronous - AWS Step Functions Allow Explicit Control Flow
Debugging

CloudWatch Logs

START RequestId: f37cc213-9091-4cf7-afdb-18991b93d468 Version: 1

[INFO] 2017-11-01T15:40:01.369Z f37cc213-9091-4cf7-afdb-18991b93d468
{}  
  "metric.name": "TotalRecords",
  "metric.value": 1

[INFO] 2017-11-01T15:40:01.546Z f37cc213-9091-4cf7-afdb-18991b93d468
Rule [cloudtrail_root_account_usage] triggered an alert on log type [cloudwatch]

[INFO] 2017-11-01T15:40:01.825Z f37cc213-9091-4cf7-afdb-18991b93d468
Sent alert to 'jactest_prod_streamalert_alert_processor' with lambda request

[INFO] 2017-11-01T15:40:01.826Z f37cc213-9091-4cf7-afdb-18991b93d468
{}
  "metric.name": "TotalProcessedSize",
  "metric.value": 1054

[INFO] 2017-11-01T15:40:01.827Z f37cc213-9091-4cf7-afdb-18991b93d468
{}
  "metric.name": "FailedParses",
  "metric.value": 0

[INFO] 2017-11-01T15:40:01.827Z f37cc213-9091-4cf7-afdb-18991b93d468
{}
  "metric.name": "TriggeredAlerts",
  "metric.value": 1

[INFO] 2017-11-01T15:40:01.827Z f37cc213-9091-4cf7-afdb-18991b93d468
[Firehose] Sending 1 records to streamalert_data_cloudwatch_events

[INFO] 2017-11-01T15:40:02.34Z f37cc213-9091-4cf7-afdb-18991b93d468
{}
  "metric.name": "FirehoseRecordsSent",
  "metric.value": 1

[INFO] 2017-11-01T15:40:02.35Z f37cc213-9091-4cf7-afdb-18991b93d468
[Firehose] Successfully sent 1 messages to streamalert_data_cloudwatch_events

END RequestId: f37cc213-9091-4cf7-afdb-18991b93d468
Monitoring

Metrics: Alarms and Dashboards
Writing
Scaling
Securing
Deploying
Monitoring

Sharing a Serverless App
Incoming Data

Laptops, Workstations, Servers

SaaS Applications

StreamAlert Apps

Kinesis

S3

SNS

Kinesis Firehose

S3

Athena

Real-Time Data Analysis

Classify and Process Rules

Alert Processing

Outgoing Alerts

APIs

Lambda

PagerDuty

Slack

S3

Lambda
Serverless is Service-Oriented, Secure by Default, Flexible, Scalable, Reliable, Cost-Effective.
Thank You O’Reilly Security!

Jack Naglieri | @jack_naglieri
StreamAlert Architect

Austin Byers | @austinbbyers
BinaryAlert Architect

@mimeframe
BinaryAlert Concept, YARA Rules
StreamAlert Concept

@ryandeivert
Core StreamAlert Contributor

@fusionrace
BinaryAlert Contributor, YARA Rules

@emma_C_Lin
StreamAlert Contributor

@javutin
StreamAlert Contributor
Questions?

binaryalert.io
streamalert.io

@jack_naglieri
@austinbbyers