Cloud Security Automation: From Infrastructure to App
Introduction

Frank Kim

- ThinkSec
  - Founder
- SANS Institute
  - Former CISO
  - Curriculum Lead
    - DevSecOps and Management
  - Author & Instructor
    - SEC540, MGT512, MGT514

- Contact
  - frank.kim@thinksec.com
  - @fykim
Infrastructure Security Automation
Cloud Custodian

• Open source tool to manage cloud environments
  – Policies for security and cost management
  – Written in Python with policy rules in YAML
  – Supports AWS, Azure, and GCP

• Example policies
  – Detect root logins and logins from invalid IPs
  – Block resources in non-standard regions
  – Configure ELB TLS ciphers and protocols
  – Configure settings and block block public S3 object ACLs
  – Detect and remediate SecurityGroup violations
policies:
  - name: high-risk-security-groups-remediate
    resource: security-group
    description: Remove any rule from a security group that allows 0.0.0.0/0 or ::/0 (IPv6) ingress and notify the user who added the violating rule.
    mode:
      type: cloudtrail
      events:
        - source: ec2.amazonaws.com
          event: AuthorizeSecurityGroupIngress
          ids: "requestParameters.groupId"
        - source: ec2.amazonaws.com
          event: AuthorizeSecurityGroupEgress
          ids: "requestParameters.groupId"
        - source: ec2.amazonaws.com
          event: RevokeSecurityGroupEgress
          ids: "requestParameters.groupId"
        - source: ec2.amazonaws.com
          event: RevokeSecurityGroupIngress
          ids: "requestParameters.groupId"
    filters:
      - or:
        - type: ingress
          Cidr:
            value: "0.0.0.0/0"
        - type: ingress
          CidrV6:
            value: "::/0"
    actions:
      - type: remove-permissions
        ingress: matched
Remediation Workflow

AWS CloudTrail

AWS activities generate CloudTrail logs

Amazon CloudWatch

CloudWatch Event Rule and Target trigger response

AWS Lambda

Lambda function triggered to remediate violations
CloudTrail Log

```json
"eventTime":"2019-04-09T16:53:39Z",
"eventSource":"ec2.amazonaws.com",
"eventName":"AuthorizeSecurityGroupIngress",
"awsRegion":"us-west-2",
"sourceIPAddress":"67.169.115.247",
"userAgent":"signin.amazonaws.com",
"requestParameters":{
  "groupId":"sg-099675a2740d77207",
  "ipPermissions":{
    "items":[
      {
        "ipProtocol":"tcp",
        "fromPort":0,
        "toPort":65535,
        "groups":{
        },
        "ipRanges":{
          "items":[
            {
              "cidrIp":"0.0.0.0/0"
            }
          ]
        }
      }
    ]
  }
}
```
CloudWatch Event Rule

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

- Event Pattern
- Schedule

Build event pattern to match events by service

- Service Name: EC2
- Event Type: AWS API Call via CloudTrail

For AWS API call events, CloudWatch Events supports the same read/write APIs as CloudTrail does. Read-only APIs, such as those that begin with List, Get, or Describe are not supported by CloudWatch Events. See more details about which services are supported by CloudTrail.

- Any operation
- Specific operation(s)

- AuthorizeSecurityGroupIngress
- AuthorizeSecurityGroupEgress
- RevokeSecurityGroupEgress
- RevokeSecurityGroupIngress
CloudWatch Event Log

- 16:54:05 START RequestId: c8414088-083f-49cf-a3b6-bf4c47ca4534 Version: $LATEST
- 16:54:05 [INFO] 2019-04-09T16:54:05.928Z c8414088-083f-49cf-a3b6-bf4c47ca4534 Processing event
- 16:54:05 "eventTime": "2019-04-09T16:53:39Z",
- 16:54:05 "eventName": "AuthorizeSecurityGroupIngress",
- 16:54:05 "awsRegion": "us-west-2",
- 16:54:05 "sourceIPAddress": "67.169.115.247",
- 16:54:05 "userAgent": "signin.amazonaws.com",
- 16:54:05 "requestParameters": {},
- 16:54:05 "groupId": "sg-099675a2740d77207",
- 16:54:05 "ipPermissions": [
  
  "ipProtocol": "tcp",
  "fromPort": 0,
  "toPort": 65535,
  "groups": [],
  "ipRanges": [
    "cidrIp": "0.0.0.0/0"
  ]
]

CloudWatch triggers less than a minute after the bad event

Event name is AuthorizeSecurityGroupIngress

SecurityGroup that was updated

Allows all IPs

Event invokes a RemovePermission action
CloudWatch Event Target

 Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function

Function* custodian-high-risk-security-groups-remediate

- Configure version/alias
- Configure input

Add target*
#2

DevSecOps Automation
DevOps cycles through five key phases:

- **PRE-COMMIT**: Activities before code is checked in to version control
- **COMMIT**: Automated build and Continuous Integration (CI) steps
- **ACCEPTANCE**: Automated acceptance and functional testing with Continuous Delivery (CD)
- **PRODUCTION**: Steps before, during, and after code is deployed to production
- **OPERATIONS**: Continuous monitoring, testing, audit, and compliance checks
Secure DevOps Practices

Learn to build, deliver, and deploy modern applications using secure DevOps and cloud principles, practices, and tools.

SEC566: Cloud and DevOps Security Automation

www.sans.org/SEC566

Serverless Security Top Ten

1. Function as a Service (FaaS) & Serverless
2. Network State-ful Security
3. Service Authentication
4. Identity and Access Management
5. Code Sanitization
6. Secret and Key Management
7. Runtime Security
8. Cloud Workload and Application Security
9. Security Compliance and Compliance
10. Security Incident and Event Management

Securing Cloud Services

- AWS
- Google Cloud
- Microsoft Azure

Secure DevOps Toolchain

1. Pre-Commit
   - Secure Code Analysis
   - Infrastructure as Code
   - Security and Privacy
   - Access Controls
2. Build
   - Dependency Management
   - Build Security
   - Continuous Integration
3. Test
   - Automated Security Testing
   - Security Configuration
4. Release
   - Secure Configuration Management
   - Secure Deployment
   - Continuous Delivery
5. Run
   - Security Monitoring
   - Security Monitoring
   - Security Monitoring

Continuous Security Monitoring, Testing, and Compliance Checks

- Security Configuration Checks
- Security Web Application
- Security Container
- Security Compliance
- Security Monitoring
- Security Monitoring
- Security Monitoring

Building a DevSecOps Program (CALMSS)

- Culture
  - Develop a culture of security across your organization.
- Automate
  - Automate security tests and controls.
- Build
  - Build secure software by default.
- Test
  - Test security throughout the development process.
- Educate
  - Educate your team on security best practices.

Serverless Security Top Ten

1. Function as a Service (FaaS) & Serverless
   - AWS Lambda
   - Google Cloud Functions
   - Microsoft Azure Functions
2. Network State-ful Security
   - VPC
   - Subnet
   - Security Group
3. Service Authentication
   - JWT
   - OAuth
   - SAML
4. Identity and Access Management
   - IAM
   - RBAC
   - Policy
5. Code Sanitization
   - Static Code Analysis
   - Dynamic Code Analysis
6. Secret and Key Management
   - Secrets Manager
   - Key Vault
7. Runtime Security
   - Containers
   - Container Security
8. Cloud Workload and Application Security
   - API
   - Microservices
   - Infrastructure as Code
9. Security Compliance and Compliance
   - PCI-DSS
   - HIPAA
   - GLBA
10. Security Incident and Event Management
    - SIEM
    - Threat Intelligence
    - Incident Response

Continuous Integration (CI)

- Build automation
- Code quality checks
- Dependency validation
- Security testing

Continuous Deployment (CD)

- Automation tools
- Continuous integration
- Automated testing

Continuous Monitoring (CM)

- Security monitoring
- Performance monitoring
- Availability monitoring

Continuous Feedback (CF)

- Security feedback
- Performance feedback
- Availability feedback

Continuous Learning (CL)

- Security training
- Performance training
- Availability training

Continuous Improvement (CI)

- Security improvement
- Performance improvement
- Availability improvement

Serverless Security Top Ten

1. Function as a Service (FaaS) & Serverless
   - AWS Lambda
   - Google Cloud Functions
   - Microsoft Azure Functions
2. Network State-ful Security
   - VPC
   - Subnet
   - Security Group
3. Service Authentication
   - JWT
   - OAuth
   - SAML
4. Identity and Access Management
   - IAM
   - RBAC
   - Policy
5. Code Sanitization
   - Static Code Analysis
   - Dynamic Code Analysis
6. Secret and Key Management
   - Secrets Manager
   - Key Vault
7. Runtime Security
   - Containers
   - Container Security
8. Cloud Workload and Application Security
   - API
   - Microservices
   - Infrastructure as Code
9. Security Compliance and Compliance
   - PCI-DSS
   - HIPAA
   - GLBA
10. Security Incident and Event Management
    - SIEM
    - Threat Intelligence
    - Incident Response

Continuous Integration (CI)

- Build automation
- Code quality checks
- Dependency validation
- Security testing

Continuous Deployment (CD)

- Automation tools
- Continuous integration
- Automated testing

Continuous Monitoring (CM)

- Security monitoring
- Performance monitoring
- Availability monitoring

Continuous Feedback (CF)

- Security feedback
- Performance feedback
- Availability feedback

Continuous Learning (CL)

- Security training
- Performance training
- Availability training

Continuous Improvement (CI)

- Security improvement
- Performance improvement
- Availability improvement
DevOps Security Tools and Processes

- Pre-commit: Threat Modeling, IDE Security Plugins, Pre-commit Hooks, Peer Code Review
- Commit: Static Code Analysis, Security Unit Tests, Dependency Management, Container Security
- Production: Security Smoke Tests, Security Configuration, Secrets Management, Server Hardening
- Operations: Blameless Postmortems, Continuous Monitoring, Penetration Testing, Threat Intelligence
Infrastructure as Code

Defining infrastructure configuration in code:

- Use high-level languages and templates to provision systems, install and configure packages, and manage users, groups, storage, firewalls....
- Take advantage of platform APIs and community/vendor modules
- Check changes into version control; review and test in advance
- Deploy through automated build pipeline (Continuous Delivery)

Programmable Infrastructure lets you:

- Treat runtimes like cattle, not pets
- Standardize within/across environments
- Create environments that are easy and cheap to set up, tear down
Blue/Green Deployment

Divert traffic from one environment to another
  • Each running a different version of the application

Benefits of blue/green deployments
  • Reduced downtime
  • Improved ability to rollback
  • Faster deployment of features and bug fixes

Use blue/green deploys when you have:
  • Immutable infrastructure
  • Well defined environment boundary
  • Ability to automate changes
Where is the blast radius in this example?
LoadBalancerListener **Ssl**:
Type: AWS::ElasticLoadBalancingV2::Listener  
Condition: CreateHttpsResources  
Properties:  
  LoadBalancerArn: !Ref LoadBalancer  
  Port: 443  
  Protocol: HTTPS  
  Certificates:  
    - CertificateArn: !Ref CertificateArn  
DefaultActions:  
  - Type: forward  
    TargetGroupArn: !Ref TargetGroup **Ssl**

LoadBalancerListener **SslGreen**:
Type: AWS::ElasticLoadBalancingV2::Listener  
Condition: CreateHttpsResources  
Properties:  
  LoadBalancerArn: !Ref LoadBalancer  
  Port: 444  
  Protocol: HTTPS  
  SslPolicy: ELBSecurityPolicy-2016-08  
  Certificates:  
    - CertificateArn: !Ref CertificateArn  
DefaultActions:  
  - Type: forward  
    TargetGroupArn: !Ref TargetGroup **SslGreen**
#3 Application Security Automation
AWS WAF Security Automations

Preconfigured protections for AWS WAF
• Block common SQL Injection and XSS attacks
• Stop scanners, HTTP floods, and bad bots
• Prevent access from known bad IP addresses

Easy to deploy CloudFormation template
• Provided by AWS Labs
• Configurable to enable/disable certain features
• Source code available on GitHub
**AWS WAF Security Automations Architecture**

- **AWS WAF**
  - SQL Injection and XSS protection
  - Bad bot and scraper protection
  - Scanner and HTTP flood protection
  - Known attacker/bad IP protection
  - IP whitelist/blacklist

- **Amazon CloudFront**
- **Application Load Balancer**

- **Access Logs**
- **Requests to honeypot endpoint**

- **S3 Log Bucket**
- **Amazon API Gateway**
- **AWS Lambda Access Handler**
- **AWS Lambda Log Parser**
- **AWS Lambda IP Lists Parser**
- **CloudWatch Event**

- **Third-party IP reputation lists**

WAF Infrastructure Code

WafAcceptance:
  Type: AWS::CloudFormation::Stack
  DependsOn: [ Buckets ]
  Properties:
    TemplateURL: !Join ["/", [!FindInMap [S3BaseUrl, !Ref "AWS::Region" , url], !Ref TemplateBucket, "templates/waf/deployment/dist/aws-waf-security-automations.template"]]
  Parameters:
    TemplateBucket: !Ref TemplateBucket
    SqlInjectionProtectionParam: "yes"
    CrossSiteScriptingProtectionParam: "yes"
    ActivateHttpFloodProtectionParam: "yes"
    ActivateScannersProbesProtectionParam: "yes"
    ActivateReputationListsProtectionParam: "no"
    ActivateBadBotProtectionParam: "yes"
Infrastructure Deployment via Jenkins

- **Deploy step calls:**
  - `aws cloudformation deploy --stack-name DM-Infrastructure --template-file master.yaml`
• Infrastructure Security
  – Identify and remediate security issues with Cloud Custodian

• DevSecOps Pipeline
  – Inject security checks into the Continuous Delivery pipeline

• Application Security
  – Leverage Infrastructure as Code to automate deployment and blocking

In Summary
What’s new in SEC540?

• Updated title
  – SEC540: Cloud Security and DevOps Automation
  – More accurately reflects course content and student base

• NetWars bonus challenges
  – Days 1-4 from 5pm - 7pm

• Microsoft Azure content
  – Azure with Terraform, Kubernetes, and blue/green

• New SEC540 challenge coin
  – Participants receive a SEC540 sticker
Cloud & DevOps Security
Summit & Training
Denver, CO
SUMMIT: Nov 4-5
TRAINING: Nov 6-10
sans.org/CloudDevOps
Material based on
SEC540: Cloud Security and DevOps Automation