Securing Microservices

Containerized Security in AWS

Mike Gillespie, Solutions Architect, Amazon Web Services
Splitting Monoliths
Ten Years Ago
Splitting Monoliths Ten Years Ago
Splitting Monoliths Five Years Ago
Splitting Monoliths

Five Years Ago

REST JSON
Fast binary encodings
Microservices
Evolution of Business Logic

- Monolith
- Microservices
- Functions
Competing Forces

- Business
  - Build it faster
  - Make it secure
  - Keep it stable

- Development
- Security
- Operations
### Networking
- Virtual Private Cloud: Isolated cloud resources
- Web Application Firewall: Filter Malicious Web Traffic
- Shield: DDoS protection
- Certificate Manager: Provision, manage, and deploy SSL/TSL certificates

### Encryption
- Key Management Service: Manage creation and control of encryption keys
- CloudHSM: Hardware-based key storage
- Server-Side Encryption: Flexible data encryption options

### Identity & Management
- IAM: Manage user access and encryption keys
- SAML Federation: SAML 2.0 support to allow on-prem identity integration
- Directory Service: Host and manage Microsoft Active Directory
- Organizations: Manage settings for multiple accounts

### Compliance
- Service Catalog: Create and use standardized products
- Config: Track resource inventory and changes
- CloudTrail: Track user activity and API usage
- CloudWatch: Monitor resources and applications
- Inspector: Analyze application security
- Macie: Discover, Classify & Protect data

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
With AWS, Security Is a Shared Responsibility

Customers are responsible for security ‘in’ the Cloud

AWS is responsible for security ‘of’ 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 Global Infrastructure
- Regions
- Availability Zones
- Edge Locations

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
VPC Security

Instance Level Firewalls – Security Groups
Subnet Network Rules – NACLs
Intelligent Threat Protection – GuardDuty
Inline Network Security – 3rd Party Marketplace

Select tools that enable automation!
Host-Based Agents

Amazon Inspector
AWS Simple Server Manager
3rd Party Agents
   Anti-virus
   IPS
   DLP

Again - Select tools that enable automation!
Amazon Machine Image Builds

ECS Optimized AMI

- ECS Optimised Amazon Linux
- RHEL
- Ubuntu
- Container Centric OS

Foundational AMI

- Security best practices
- Provisioners
- Loggers
- Config, and so on

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
API Gateway

Acts as a front door to the microservices and provides:
  Authentication
  Rate Throttling
  Monitoring
  Versioning

Select an API Gateway that support Automation
Web Application Firewall

WAFs provide Layer 7 protection for CVEs and OWASP Top 10.

AWS Web Application Firewall

AWS Marketplace
  SaaS WAF Offerings
  Virtual Appliances
Best practices

• Define your resource limits **up front**
• It’s not just **memory** and **CPU**.
• **Monitor** usage
• Leverage **Auto Scaling**
• **Amazon Shield**
• **Infrastructure as Code**
BUILDING AN ECOSYSTEM

AWS Lambda

ECS

Fargate

ECR

EKS

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
Amazon ECS—Task & Service

Internet

EC2 INSTANCES

ECS Service

LOAD BALANCER

TASK

Container

EC2 INSTANCES

ECS AGENT

LOAD BALANCER

TASK

Container

ECS AGENT

Amazon ECS

AGENT COMMUNICATION SERVICE

API

CLUSTER MANAGEMENT ENGINE

KEY/VALUE STORE

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
PRODUCTION WORKLOADS ON AWS

- AWS VPC networking mode
- Advanced task placement
- Deep integration with AWS services
- ECS CLI

- Global footprint
- Powerful scheduling engines
- Auto scaling
- CloudWatch metrics
- Load balancers
Amazon EKS

Highly available

Upstream

Production workloads

Integrated with AWS Services
1. Pre ENI Attachment: The primary ENI (eth0) is in the default namespace.

2. ENI Attached: The new ENI (eth1) is in the default namespace.

3. ENI Provisioned: The ECS Agent invokes CNI plugins to move the new ENI into a new namespace and configure it with the addresses and routes.
Best practices

• Signing container images (Docker content trust)

• Set filesystems to be read-only (readonlyRootFilesystem)

• Remove setuid/setgid binaries from images (defang)

• Set containers to run as non-root user

• Run Vulnerability Analysis on Container/VM Build in pipeline
Storing secrets in **environment variables**

```json
"environment" : [ 
  { "name" : "DB_USERNAME", "value" : "admin" },
  { "name" : "DB_PASSWORD", "value" : "Pa$$word123" } 
]
```

- Suggested by 12-factor apps (III. Config)
- Environment variables can be seen in too many places
  - Linked containers
  - ECS API calls
  - Docker inspect
- Can’t be deleted

https://12factor.net/
Protecting secrets using IAM roles for tasks

Benefits

• Simplify usage of AWS SDKs in containers
• Credential isolation between tasks/container
• Authorization per task/container
• Auditability in Amazon CloudTrail with taskArn
VPC flow logs and Task ENI

StopTask: {
  task: "52c..."
}

$ docker diff / inspect
## AWS Partner Community

<table>
<thead>
<tr>
<th>Foundation</th>
<th>DevOps</th>
<th>Monitoring and Logging</th>
<th>Security</th>
<th>Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoreOS</td>
<td>shippable</td>
<td>circleci</td>
<td>aqua</td>
<td>linkerd</td>
</tr>
<tr>
<td>docker</td>
<td>CloudBees</td>
<td>Datadog</td>
<td>Twistlock</td>
<td>TiGERA</td>
</tr>
<tr>
<td>MESOSPHERE</td>
<td>GitLab</td>
<td>sysdig</td>
<td>New Relic</td>
<td>weaveWorks</td>
</tr>
<tr>
<td>ATLASSIAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.