AUTOMATED, CONTINUOUS & VISIBLE: BUILDING A SOLID SECURITY CULTURE AT SPEED

Aaron McKeown
Head of Security Engineering and Architecture
Xero
Key Learning Objectives

OBJECTIVE 1
Security accessibility for developers

OBJECTIVE 2
Challenges are everywhere

OBJECTIVE 3
Sticking to your principles
Global small business platform

Connecting people with the right numbers anytime, anywhere, on any device
1.2m Subscribers

1,800+ Staff globally

$1.4t Transactions recorded in the year to 31 March 2017
Supporting our next wave of growth: a migration to the public cloud

<table>
<thead>
<tr>
<th>Weeks</th>
<th>Minutes</th>
<th>Seconds</th>
<th>Milliseconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-premises hardware</td>
<td>EC2 VM</td>
<td>Containers app</td>
<td>Lambda function</td>
</tr>
</tbody>
</table>
Maintaining and improving security

- No breaches or accidental information disclosures
- Support recognized information security standards
- Automated and agile infrastructure stack, with developer initiated security components

Design principles: Apply security at all layers, Encrypt at rest and in transit, Technical and logical segregation
Agile growth: AWS accounts & services
Product based teams
Key principles for Security Engineering

1. Repeatable and automated build and management of security systems
2. Accelerated pace of security innovation
3. On-demand security infrastructure that works at any scale
Challenges are everywhere

- Configuration Drift
- Context
- Speed of Innovation
- Shared Responsibility
Configuration drift
Speed

km/h
Shared responsibility

Xero Applications & Content

Network Security
Identity & Access Control
Inventory & Config
Data Encryption

AWS Foundation Services

Compute
Storage
Database
Networking

AWS Global Infrastructure

Availability Zones
Edge Locations
Regions

Security IN the cloud
Security OF the cloud

Xero + security partners

Identity & Access Control
Inventory & Config
If I “need” to encrypt at rest, it should be easy. Providing access to KMS makes it easy.

Self-service trumps having to request things from another team, which trumps having to raise a ticket. If it’s too hard, I’ll do it “later”. PACMAN is a simple thing, but it’s actually changed my life. Resetting MANAGE password was a PITA before.

If I “need” to patch my instances for vulnerabilities, Halo reporting makes that easy.

Luke Ryan, Principal Engineer, Xero
Welcome to Platform Access Manager. From here you can see the status of all your platform identities. Since you manage a team you can also view the access status for your team members.

**Identities**

- **Manage Domain**
  - Status: Active
  - Password has expired

- **AWS**
  - Status: Active
  - MFA: Active
  - 8 days until disabled

- **GitHub**
  - Username: [redacted]

- **Duo**
  - Status: Active
  - Push devices: 1
I try to detect what you would like from keywords in your messages. You can ask me about things like your Manage Domain account, AWS account, MFA devices, GitHub username and whether you have signed the DAA.

**Manage Domain:**
- Username: *auron.mckeown*
- Status: Active
- GitHub Username: *admin.aaron*

**AWS Amazon Web Services:**
- Username: *auron.mckeown.aws.com*
- Status: Disabled

Here is all the info I can find for you.

**Duo:**
- Username: *auron.mckeown*
- Status: Active
- Number of push devices: 1
Nau mai Aaron

Welcome to Platform Access Manager. From here you can see the status and access status for your team members.

Identities

Manage Domain

AWS

Status: Active
MFA: Active
8 days until disabled

Request KMS Key

You can use this form to request a KMS key be created in the xero-ps-paas-sec account. This ensures secure storage of the keys used to encrypt data in your application. Key creation requires approval from a KMS administrator.

Alias

Alias:
- Environment
- Enter a service, eg s3-xii
- Portfolio

Account

Search

Region

- us-east-1
- us-west-2
- ap-southeast-2

Tags

Owner

rod.drury@xero.com

Description

E.g. Test S3 bucket named xero-ci for Platform Services

Cancel
Submit
<table>
<thead>
<tr>
<th>OS Type</th>
<th>Server (Display Name)</th>
<th>Agent Version</th>
<th>Agent State</th>
<th>Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>ip-10-98-0-109</td>
<td>4.1.1</td>
<td>Active</td>
<td>15</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-10-98-0-120</td>
<td>4.1.1</td>
<td>Active</td>
<td>20</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-10-98-0-174</td>
<td>4.1.1</td>
<td>Active</td>
<td>15</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-10-98-0-186</td>
<td>4.1.1</td>
<td>Active</td>
<td>17</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-10-98-0-245</td>
<td>4.1.1</td>
<td>Active</td>
<td>17</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-10-98-0-252</td>
<td>4.1.1</td>
<td>Active</td>
<td>17</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-10-98-0-38</td>
<td>4.1.1</td>
<td>Active</td>
<td>17</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-10-98-0-45</td>
<td>4.1.1</td>
<td>Active</td>
<td>17</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-172-31-22-26</td>
<td>4.1.3</td>
<td>Active</td>
<td>1</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-172-31-27-204</td>
<td>4.1.3</td>
<td>Active</td>
<td>1</td>
</tr>
<tr>
<td>Linux</td>
<td>ecommerce-connector</td>
<td>4.1.3</td>
<td>Active</td>
<td>1</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-172-31-29-150</td>
<td>4.1.3</td>
<td>Active</td>
<td>1</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-172-31-29-250</td>
<td>4.1.3</td>
<td>Active</td>
<td>1</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-172-31-31-218</td>
<td>4.1.3</td>
<td>Active</td>
<td>1</td>
</tr>
<tr>
<td>Linux</td>
<td>ip-172-31-6-236</td>
<td>4.1.3</td>
<td>Active</td>
<td>1</td>
</tr>
</tbody>
</table>
import cloudpassage
import os
token = os.getenv("HALO_API_KEY")
secret = os.getenv("HALO_API_SECRET_KEY")
session = cloudpassage.HaloSession(token, secret)
servers = cloudpassage.Server(session)
issues = cloudpassage.Issue(session)
for s in servers.list_all():
    issue_count = len(issues.list_all(agent_id=s.id))
    print("Group: %s\nName: %s\nIssue count: %d % (s.group_name, s.hostname, issue_count))
<table>
<thead>
<tr>
<th>Policy Name</th>
<th>Rule ID</th>
<th>Rule Name</th>
<th>Pass</th>
<th>Total</th>
<th>Pass %</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIS AWS Foundations Benchmark v1.1</td>
<td>CIS: 1.1</td>
<td>Avoid the use of the &quot;root&quot; account</td>
<td>1</td>
<td>3</td>
<td>33%</td>
</tr>
<tr>
<td></td>
<td>CIS: 1.2</td>
<td>Ensure multi-factor authentication (MFA) is enabled for all IAM users that</td>
<td>32</td>
<td>60</td>
<td>53%</td>
</tr>
</tbody>
</table>
|                                                 | CIS: 1.3| have a console 
...                                           | 35   | 50    | 70%    |
|                                                 | CIS: 1.4| Ensure access keys are rotated every 90 days or less                       | 4    | 5     | 80%    |
|                                                 | CIS: 1.5| Ensure IAM password policy requires at least one uppercase letter          | 9    | 11    | 82%    |
|                                                 | CIS: 1.6| Ensure IAM password policy requires at least one lowercase letter          | 3    | 6     | 50%    |
|                                                 | CIS: 1.7| Ensure IAM password policy require at least one symbol                     | 1    | 25    | 4%     |
|                                                 | CIS: 1.8| Ensure IAM password policy require at least one number                     | 25   | 100   | 25%    |
|                                                 | CIS: 1.9| Ensure IAM password policy requires minimum length of 14 or greater        | 3    | 9     | 33%    |
|                                                 | CIS: 1.10| Ensure access keys are rotated every 90 days or less                      | 0    | 0     | -      |
|                                                 | CIS: 1.11| Ensure a log metric filter and alarm exist for AWS Config configuration  | 35   | 50    | 70%    |
|                                                 | CIS: 1.12| Ensure IAM password policy require at least one lowercase letter          | 3    | 6     | 50%    |
|                                                 | CIS: 1.13| Ensure IAM password policy require at least one symbol                     | 1    | 25    | 4%     |
|                                                 | CIS: 1.14| Ensure IAM password policy require at least one number                     | 25   | 100   | 25%    |
|                                                 | CIS: 1.15| Ensure IAM password policy requires minimum length of 14 or greater        | 3    | 9     | 33%    |
|                                                 | CIS: 1.16| Ensure access keys are rotated every 90 days or less                      | 0    | 0     | -      |
Security Culture

SECURITY CULTURE NEEDS TO BE AGILE

- Short frequent cycles
- Security as a service
- Cross-functional teams
- Communication
Security Retrospective

“Security is a journey, not a destination”

Make your product teams and vendors your partners

Strategic thinking is critical, develop and share your plans

Evolve your architecture, evolve your skills and understanding
Key principles for Security Engineering

1. Repeatable and automated build and management of security systems
2. Accelerated pace of security innovation
3. On-demand security infrastructure that works at any scale
Evolve your principles: automated, continuous & visible

“If you don't put a metric on it, how do you know what normal is?”
FAST & SECURE
Apply What You Have Learned Today

- Foster an actual security culture
- "Let Go"

- Responsibility shift
- Develop a roadmap
- DevSecOps Gap Analysis

- Continuous improvement cycles ("90 day plans")