Closing the Cloud Security Gap with Privileged Access Governance

ISSA-LA Chapter Dinner Meeting
Presenter: Art Poghosyan, CEO, Britive Inc.

October 16, 2019
Why is Cloud So Difficult to Secure?

- It’s dynamic and high-volume
- There is a learning curve
- You don’t have a complete control over it
- Cloud-native security tools are still maturing
- Wrong approach (and tools) is used
Bad Guys Relentlessly Target Privileged Access

As evidenced by research surveys:

- 2019 Verizon DBIR
- Forrester & Gartner Reports
- McAfee Report

And data breaches:

- Capital One: 106 million (2019)
- Uber: 57 million (2016)
- J.P. Morgan: 76 million (2014)
Privileged Access is More than “r00t” Accounts

Cloud components that require privileged access:
- Management Consoles
- CLIs
- APIs
- Resources
- Workloads
- Administrative functions
- Data level access

How is privileged access defined:
- Roles
- Policies
- Groups
- Profiles
- Permissions
- Keys
Privileged Access is a Major Challenge

Difficult to get visibility into existing access

- who has
- what access
- in what platform

Each cloud platform has its proprietary privilege model
Cloud Privileges - Granular But Complex
Cloud Privileges - Granular But Complex
Managing Complexity is Not Easy

Especially with spreadsheets

Example shown is from GCP

<table>
<thead>
<tr>
<th>Owner</th>
<th>Editor</th>
<th>Viewer</th>
<th>Security Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>accessapproval</td>
<td>requests.approve</td>
<td>accessapproval.requests.get</td>
<td>accessapproval.requests.get</td>
</tr>
<tr>
<td>accessapproval</td>
<td>requests.get</td>
<td>accessapproval.requests.get</td>
<td>accessapproval.requests.get</td>
</tr>
<tr>
<td>accessapproval</td>
<td>requests.list</td>
<td>accessapproval.requests.list</td>
<td>accessapproval.requests.list</td>
</tr>
<tr>
<td>accessapproval</td>
<td>settings.get</td>
<td>accessapproval.settings.get</td>
<td>accessapproval.settings.get</td>
</tr>
<tr>
<td>accessapproval</td>
<td>settings.update</td>
<td>accessapproval.settings.update</td>
<td>accessapproval.settings.update</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>access,levels.create</td>
<td>accesscontextmanager.access,levels.create</td>
<td>accesscontextmanager.access,levels.create</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>access,levels.delete</td>
<td>accesscontextmanager.access,levels.delete</td>
<td>accesscontextmanager.access,levels.delete</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>access,levels.get</td>
<td>accesscontextmanager.access,levels.get</td>
<td>accesscontextmanager.access,levels.get</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>access,levels.list</td>
<td>accesscontextmanager.access,levels.list</td>
<td>accesscontextmanager.access,levels.list</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>access,levels.update</td>
<td>accesscontextmanager.access,levels.update</td>
<td>accesscontextmanager.access,levels.update</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>accessPolicies.create</td>
<td>accesscontextmanager.accessPolicies.create</td>
<td>accesscontextmanager.accessPolicies.create</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>accessPolicies.delete</td>
<td>accesscontextmanager.accessPolicies.delete</td>
<td>accesscontextmanager.accessPolicies.delete</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>accessPolicies.getPolicy</td>
<td>accesscontextmanager.accessPolicies.getPolicy</td>
<td>accesscontextmanager.accessPolicies.getPolicy</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>accessPolicies.list</td>
<td>accesscontextmanager.accessPolicies.list</td>
<td>accesscontextmanager.accessPolicies.list</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>accessPolicies.setiamPolicy</td>
<td>accesscontextmanager.accessPolicies.setiamPolicy</td>
<td>accesscontextmanager.accessPolicies.setiamPolicy</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>accessPolicies.update</td>
<td>accesscontextmanager.accessPolicies.update</td>
<td>accesscontextmanager.accessPolicies.update</td>
</tr>
<tr>
<td>accesscontextmanager</td>
<td>accessZones.create</td>
<td>accesscontextmanager.accessZones.create</td>
<td>accesscontextmanager.accessZones.create</td>
</tr>
</tbody>
</table>

It’s even worse when there is no process and consistency!
Complexity Results in Excessive Privilege

- Privileges are granted too broadly
- Privilege “creep” is common
- Privilege right-sizing is not a regular process

*Excessive Privileges = High Risk of Access Breach*
Standing Privileges Multiply the Risk

- Standing privileges, i.e. “always on”
- Excessive privileges
- “Un-vaulted” privileged credentials

The result is exponentially higher risk of access breach that is mostly unmanaged!
Transforming Cloud Privileged Access from Weakness to Strength!

You Can’t Control What you Can’t See

- Discover and inventory
  - Accounts & creds
  - Privileges & permissions
## Define Privilege Use Cases

<table>
<thead>
<tr>
<th>Who</th>
<th>What</th>
<th>Where</th>
<th>When</th>
<th>Why</th>
</tr>
</thead>
<tbody>
<tr>
<td>DevOps</td>
<td>Console, CLI, API, Containers, Servers</td>
<td>AWS, Azure</td>
<td>Continuous</td>
<td>Normal work</td>
</tr>
<tr>
<td>Security</td>
<td>Security Functions, Containers, Servers</td>
<td>AWS, SalesForce</td>
<td>Continuous</td>
<td>Normal work</td>
</tr>
<tr>
<td>Business</td>
<td>Admin Functions</td>
<td>Workday, SalesForce</td>
<td>Periodic</td>
<td>Delegated admin</td>
</tr>
<tr>
<td>Service Desk</td>
<td>Configuration/Admin Functions</td>
<td>ServiceNow, SalesForce</td>
<td>Continuous</td>
<td>Normal work</td>
</tr>
<tr>
<td>Project Staff</td>
<td>Configuration/Admin Functions</td>
<td>Workday</td>
<td>Occasional</td>
<td>Project needs</td>
</tr>
</tbody>
</table>
Building the Privilege Governance Framework

Basic PAM first - credential vaulting

break-glass accounts, cloud “root”, keys, etc.
Cloud Privilege Management Best Practices

- Just in Time (JIT) Privilege Grant
- Policy-Based Privilege Authorization
- Zero Standing Privileges (ZSP)
Cloud Privilege Management Best Practices

- Cloud SSO/MFA Integration
- Identity Governance & Life Cycle Integration
- Policy Compliance & Threat Monitoring
Cloud Privilege Access Governance Framework

Prioritize implementation
- Use cases
- Risk exposure
- Cloud capabilities
- Resource skills

- Credential Vaulting
- Policy-Based Privilege Authorization
- Just in Time (JIT) Privilege Grant
- Zero Standing Privileges (ZSP)
- Cloud SSO/MFA Integration
- Identity Governance & Life Cycle Integration
- Policy Compliance & Threat Monitoring
Thank You!

Art Poghosyan
CEO, Britive Inc.
art@britive.com

www.britive.com