# Closing the Cloud Security Gap with Privileged Access Governance

ISSA-LA Chapter Dinner Meeting

Presenter: Art Poghosyan, CEO, Britive Inc.

### 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:









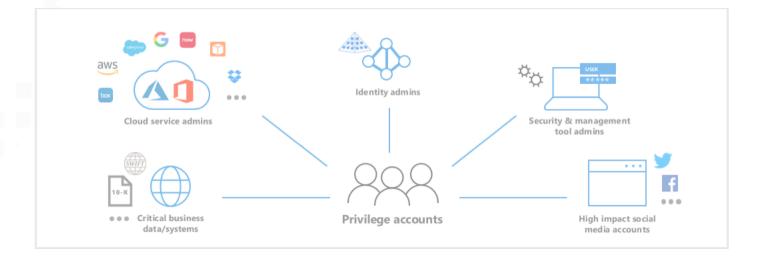
# Privileged Access is More than "r00t" Accounts

# 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



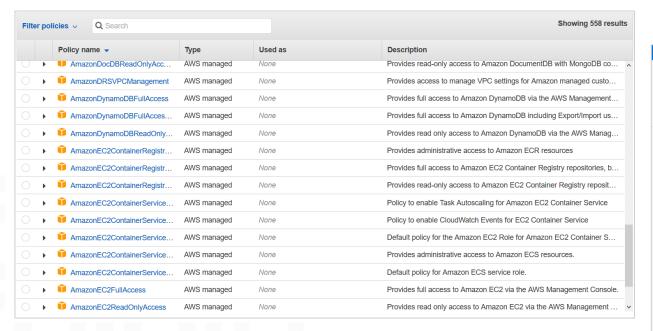
### Privileged Access is a Major Challenge

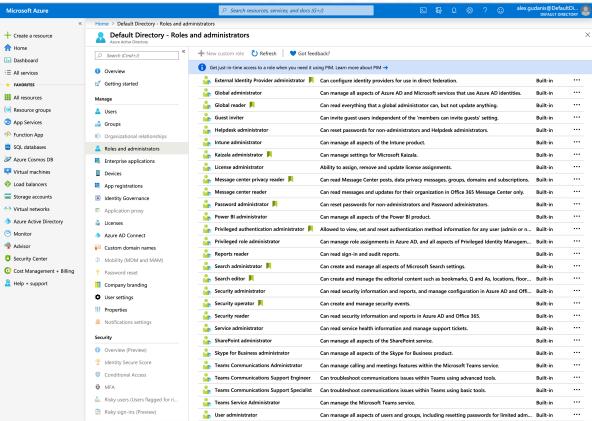
- Difficult to get visibility into existing access
  - who has
  - what access
  - in **what** platform



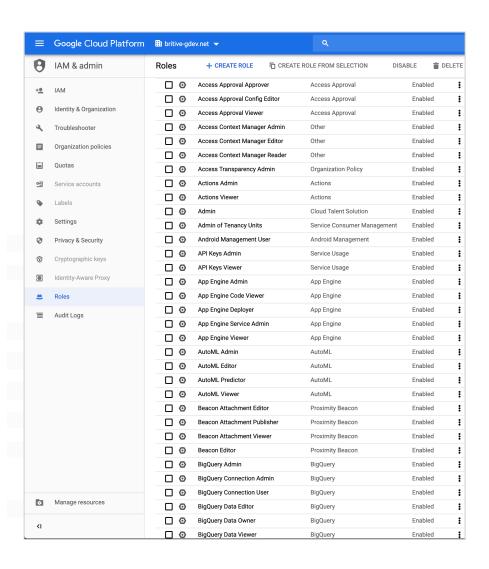
Each cloud platform has it's proprietary privilege model

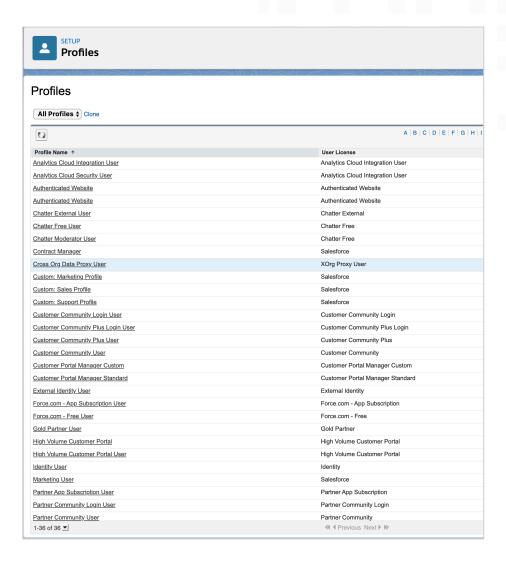
# Cloud Privileges - Granular But Complex





# Cloud Privileges - Granular But Complex





### Managing Complexity is Not Easy



### Especially with spreadsheets

#### Example shown is from GCP

Number of Permissions: 2065	Number of Permissions: 1858	Number of Permissions: 850	Number of Permissions: 533
Owner	Editor	Viewer	Security Admin
accessapproval.requests.approve			
accessapproval.requests.dismiss			
accessapproval.requests.get	accessapproval.requests.get	accessapproval.requests.get	
accessapproval.requests.list	accessapproval.requests.list	accessapproval.requests.list	accessapproval.requests.list
accessapproval.settings.get	accessapproval.settings.get	accessapproval.settings.get	
accessapproval.settings.update			
accesscontextmanager.accessLevels.create	accesscontextmanager.accessLevels.create		
accesscontextmanager.accessLevels.delete	accesscontextmanager.accessLevels.delete		
accesscontextmanager.accessLevels.get	accesscontextmanager.accessLevels.get	accesscontextmanager.accessLevels.get	
accesscontextmanager.accessLevels.list	accesscontextmanager.accessLevels.list	accesscontextmanager.accessLevels.list	accesscontextmanager.accessLevels.list
accesscontextmanager.accessLevels.update	accesscontextmanager.accessLevels.update		
accesscontextmanager.accessPolicies.create	accesscontextmanager.accessPolicies.create		
accesscontextmanager.accessPolicies.delete	accesscontextmanager.accessPolicies.delete		
accesscontextmanager.accessPolicies.get	accesscontextmanager.accessPolicies.get	accesscontextmanager.accessPolicies.get	
accesscontextmanager.accessPolicies.getlamPolicy	accesscontextmanager.accessPolicies.getlamPolicy	accesscontextmanager.accessPolicies.getlamPolicy	accesscontextmanager.accessPolicies.getlamPolicy
accesscontextmanager.accessPolicies.list	accesscontextmanager.accessPolicies.list	accesscontextmanager.accessPolicies.list	accesscontextmanager.accessPolicies.list
accesscontextmanager.accessPolicies.setlamPolicy			accesscontextmanager.accessPolicies.setlamPolicy
accesscontextmanager.accessPolicies.update	accesscontextmanager.accessPolicies.update		-
accesscontextmanager.accessZones.create	accesscontextmanager.accessZones.create		

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

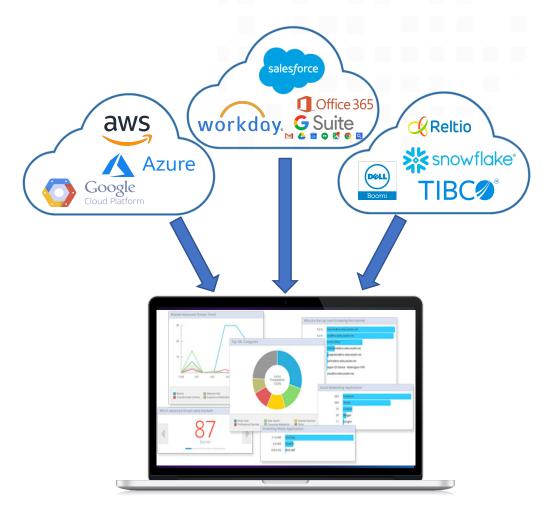


The result is exponentially higher risk of access breach that is mostly unmanaged!

# 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

Who	What	Where	When	Why
DevOps	Console, CLI, API, Containers, Servers	AWS, Azure	Continuous	Normal work
Security	Security Functions, Containers, Servers	AWS, SalesForce	Continuous	Normal work
Business	Admin Functions	Workday, SalesForce	Periodic	Delegated admin
Service Desk	Configuration/ Admin Functions	ServiceNow, SalesForce	Continuous	Normal work
Project Staff	Configuration/ Admin Functions	Workday	Occasional	Project needs

### Building the Privilege Governance Framework

Basic PAM first - credential vaulting

break-glass accounts, cloud "root", keys, etc.



### Cloud Privilege Management Best Practices







### Cloud Privilege Management Best Practices



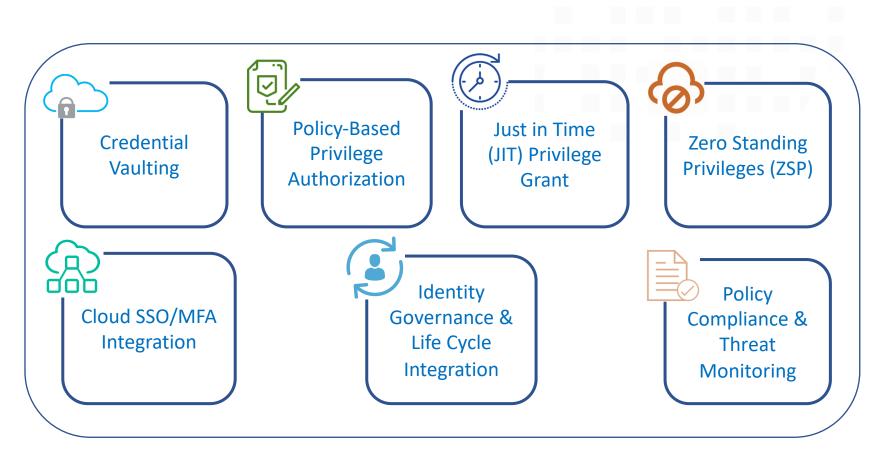




### Cloud Privilege Access Governance Framework

#### Prioritize implementation

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



### Thank You!

Art Poghosyan
CEO, Britive Inc.
<a href="mailto:art@britive.com">art@britive.com</a>

