Improper Cloud Permission Management as a Primary Cause of Data Breaches

Mariusz Burdach

11-2024

Agenda

1. Issues Caused by Improper Permission Management
2. Consequences of Improper Permission Management
3. Basic Information About Service Accounts
4. The Principle of Least Privileges
5. Identification of privileged accounts

Issues Caused by Improper Permission Management (1)

In many attacks, obtaining access to accounts with administrative rights becomes necessary.

Return on mitigation: Targeting investment to increase resilience

During Microsoft Incident Response engagements, customer environments have been found to lack mitigations that range from the simple to the more complex.

While the goal of all mitigations is to make environments more resilient to cyberattacks, customers may not always have the resources to implement all of them, and a return on mitigation framework is helpful for prioritization. Generally speaking, the lower the resources and effort involved, the higher the return on mitigation (ROM). As an example of a high return, consider a simple solution to implement context-based MFA protection. This solution is highly effective in preventing initial access (high security value) but very simple to implement (low effort). When implemented, this solution effectively prevents initial access by providing more context around the authentication attempt, such as geographic location and the application used. The additional context can be combined with requiring the user to enter a number (number matching) to complete MFA to further improve sign-in security.

Return on mitigation: Targeting investment to increase resilience continued

Higher ROM H | Medium ROM M | Lower ROM L

Return on mitigation by MITRE ATT&CK technique

MITRE ATT&CK techniques observed	Reconnaissance	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Insecure Active Directory configuration	H	L	H	H	H		L		H				
Unsecure legacy authentication is still leveraged		H	L		H							L	
Lack of detection controls		H		H			H						
Lack of EDR coverage		H			H		H		H				
Missing context-based multifactor protection mechanisms		H			H		H						
No advanced MFA protection mechanisms enabled		H			H		H						
No advanced password protection enabled		H					H		H				
Poor user lifecycle management		H					H						
Insecure operating system configuration		H					H						
Resource exposed to public access		H											
Missing or inconsistent update management		M	M				M						
Legacy and unsecure protocols		M			M		M						
Insufficient device security controls		L	M	H	H					L			M
Legacy or unsupported operating systems		L	L										
Weak email protection against common threats		L							L			L	
No MFA or MFA not mandatory for privileged accounts		L											
No vulnerability management			L	L		L					L		
Missing cloud application management and monitoring				L						L		L	
Missing security barrier between the cloud and on-premise					H				H				
No privilege separation					L		L		L				
No hardened device used for high privileged accounts					L				L				
No privileged identity management solution					L				L				
Not adhering the Least Privilege Principle					L								
Insufficient protections for local accounts									H				
Missing data classification and sharing configuration										L		L	

Source: Microsoft Digital Defense Report, October 2023

Issues Caused by Improper Permission Management (2)

Return on mitigation: Targeting investment to increase resilience continued

ROM	Issues found	% of customers with the issue
Higher		
15	No advanced MFA protection mechanisms enabled	37%
15	Poor user lifecycle management	21%
15	Lack of EDR coverage	13%
15	Lack of detection controls	10%
13	Resource exposed to public access	2%
12	Insufficient protections for local accounts	60%
12	Missing security barrier between cloud and on-premise	54%
12	Insecure Active Directory configuration	43%
12	Insufficient device security controls	8%
11	Legacy cloud authentication is still used	47%
11	No advanced password protection enabled	37%
11	Missing content based MFA protection mechanisms	24%
11	Insecure operating system configuration	3%
Medium		
8	Legacy and unsecure protocols	18%
7	Missing or inconsistent update management	43%
6	Missing cloud application management and monitoring	21%
6	No privileged identity management solution	8%
6	No MFA, or MFA not mandatory for privileged accounts	21%
6	Weak email protection against common threats	16%
6	Legacy or unsupported operating systems	14%
Lower		
4	No privilege separation	41%
4	No hardened workstations used for administration	23%
4	Missing data classification and sharing restrictions	5%
3	No vulnerability management	30%
2	No adherence to the Least Privilege Principle	63%

Source: Microsoft Digital Defense Report, October 2023

2024		2022
Misconfiguration & Inadequate Change Control	1	Identity & Access Mgmt (IAM)
Identity & Access Mgmt (IAM)	2	Insecure Interfaces and APIs
Insecure Interfaces and APIs	3	Misconfiguration & Inadequate Change Control
Inadequate Selection/ Implementation of Cloud Security Strategy	4	Inadequate Selection/ Implementation of Cloud Security Strategy
Insecure Third-Pary Resources	5	Insecure Software Development
Insecure Software Development	6	Insecure Third-Party Resources
Accidental Cloud Disclosure	7	System Vulnerabilities
System Vulnerabilities	8	Accidental Cloud Disclosure
Limited Cloud Visibility/ Observability	9	Misconfiguration & Exploitation of Serverless & Container Workloads*
Unauthenticated Resource Sharing	10	Advanced Persistent Threats
Advanced Persistent Threats	11	Cloud Storage Data Exfiltration*

**Security issues not in the top 11 for 2024*

Source: 2024 Top Threats - Cloud Security Alliance

Consequences of Improper Permission Management in Public Cloud

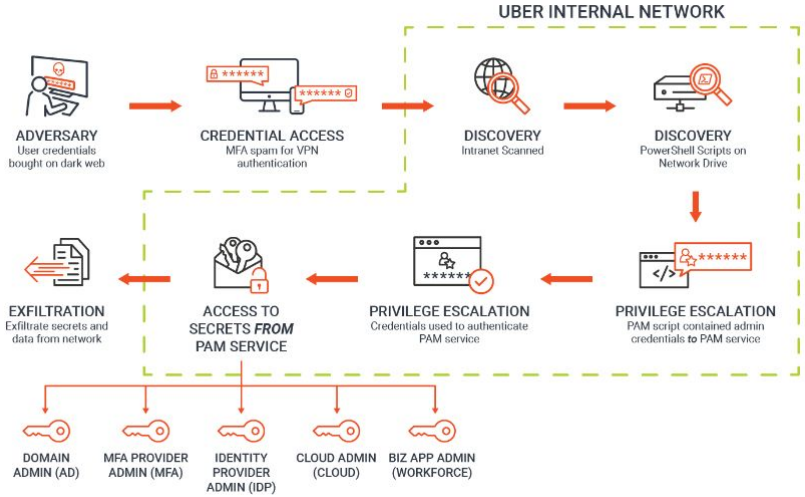
Key reasons of successful attack:

- incorrect bucket permissions
- credentials “embedded” in code/scripts
- excessive account permissions
- lack of permission separation
- incorrect resource configuration
- unused service accounts

Examples on confirmed incidents:

- Capital One - 2019 (credentials with extensive permissions in metadata)
- Shanghai Police - 2022 (incorrect configuration of public cloud resources)
- Toyota - 2023 (incorrect configuration of public cloud resources)
- Uber - …
- Microsoft - 2023

Uber 2022



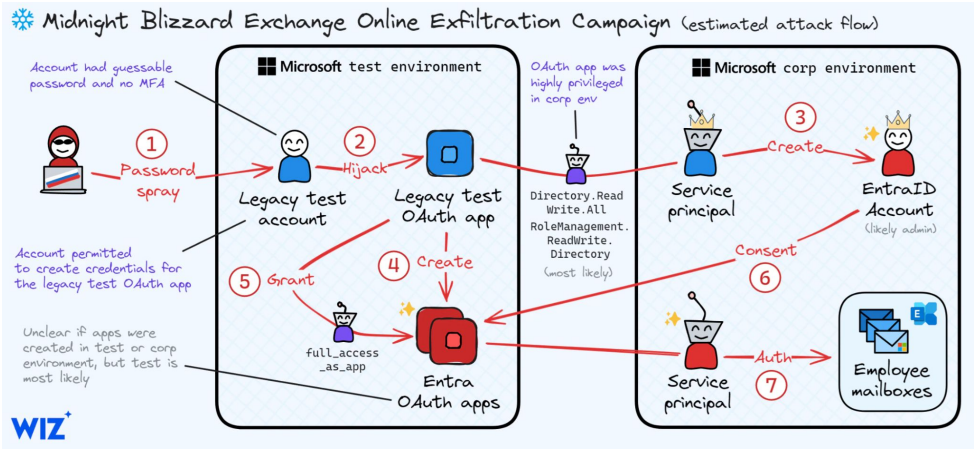
Phase 2: Discovery. Most likely, this contractor did not have special or elevated privileges to sensitive resources but did have access to a network share, as did other Uber workers. This network share was either open or misconfigured to allow broad read ACL. Within the network share, the attacker discovered a PowerShell script containing hard-coded privileged credentials to Uber's PAM solution.

Source: https://www.cyberark.com/resources/blog/unpacking-the-uber-breach

Microsoft 2023

The statement "legacy test OAuth application that had elevated access to the Microsoft corporate environment" indicates that the test application has been previously granted consent by a highly privileged Entra ID user in the corporate tenant — allowing it high privileges in the corporate environment. To understand what privileges were previously granted by a legitimate admin, we'll take note of the following statement: "They created a new user account to grant consent in the Microsoft corporate environment." In other words, either the MS Graph app permission of `Directory.ReadWrite.All` or `User.ReadWrite.All` was previously granted consent to the corporate tenant, since these are the only MS Graph permissions that allow creation of new Entra ID users. For the demonstration's purpose, let's assume that the `Directory.ReadWrite.All` permission was granted consent.

The report later states that the adversary used the Office 365 Exchange Online app permission of `full_access_as_app` to access corporate mailboxes. Since it requires admin consent, we can conclude that the newly created user in the corporate environment was assigned admin privileges. Thus, we can infer that the MS Graph app permission of `RoleManagement.ReadWrite.Directory` was also previously granted access to the corporate environment (prior to the incident).



Source: https://www.wiz.io/blog/midnight-blizzard-microsoft-breach-analysis-and-best-practices

Identification of privileged accounts

- Principal - **who**?
- Roles and permissions - **what**?
- Scope - **where**?

Principles

- Users
- Groups
- Service Accounts / Service Principal / Managed Identity

Introduction: Tacklin and shadow IT for a

Threat actors prey on unaddressed technical debt, outdated security controls, and shadow IT.

If there is a weak point in your system, threat actors are going to find it. You may be using the latest security tools to fortify your core environment, but if you still have old infrastructure, unpatched systems, outdated configurations, and apps granted too many permissions by departments you aren't even aware of, then you may be unwittingly leaving security holes for threat actors to exploit.

Quote from Microsoft Digital Defense Report 2024

Service account attached to VM

1

```
identity_email_com@vm:~$ gcloud auth list
                Credentialed Accounts
ACTIVE  ACCOUNT
        accountname@<projectname>.iam.gserviceaccount.com
*       identity@email.com
To set the active account, run:
    $ gcloud config set account `ACCOUNT`
```

2

```
identity_emai_com@vm:~$ gcloud config set account accountname@<PROJECT>.iam.gserviceaccount.com
Updated property [core/account].
```

3

```
identity_email_com@vm:~$ gcloud secrets list

identity_email_com@vm:~$ gcloud storage cp gs://trainingbucket/folder001/sourcefile destinationfile --project=example_of_project
```

Alternatywna metoda:

```
$curl -H "Metadata-Flavor: Google" "http://IP/computeMetadata/v1/instance/service-accounts/default/token"

$curl -H "Authorization: Bearer $ACCESS_TOKEN" “https://storage.googleapis.com/storage/v1/b?project=example_of_project”
```

Roles and permissions in public cloud

Azure

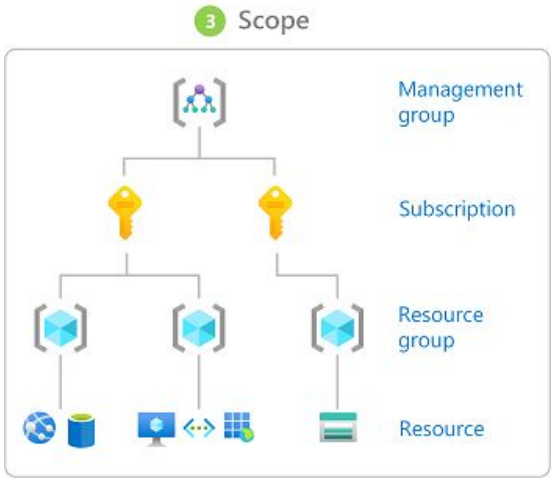
- build-in roles: 120+
- number of permissions: 20 234

GCP

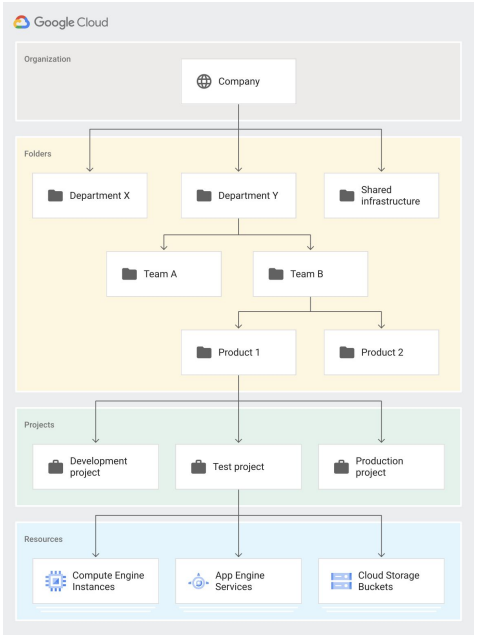
- build-in roles: 1000+
- number of permissions: 10 366

Scope

Azure



GCP



Principle of Least Privilege

- limited number of permissions in roles - create your own role
- remove admin permissions from “regular” accounts
- limit the scope to a minimum (= at the resource level)
- rules for service accounts
  - do not export keys (do not generate keys)
  - use Workload Identity / Managed Identity features
- other rules
  - limit number of admin accounts
  - use Just-In-Time Access
  - use secure environment for performing admin tasks / running scripts with admin permissions

How to identify high privilege accounts?

Focus on:

- Administrative permissions
- Permissions allowing for escalation of permissions - so-called “impersonation’

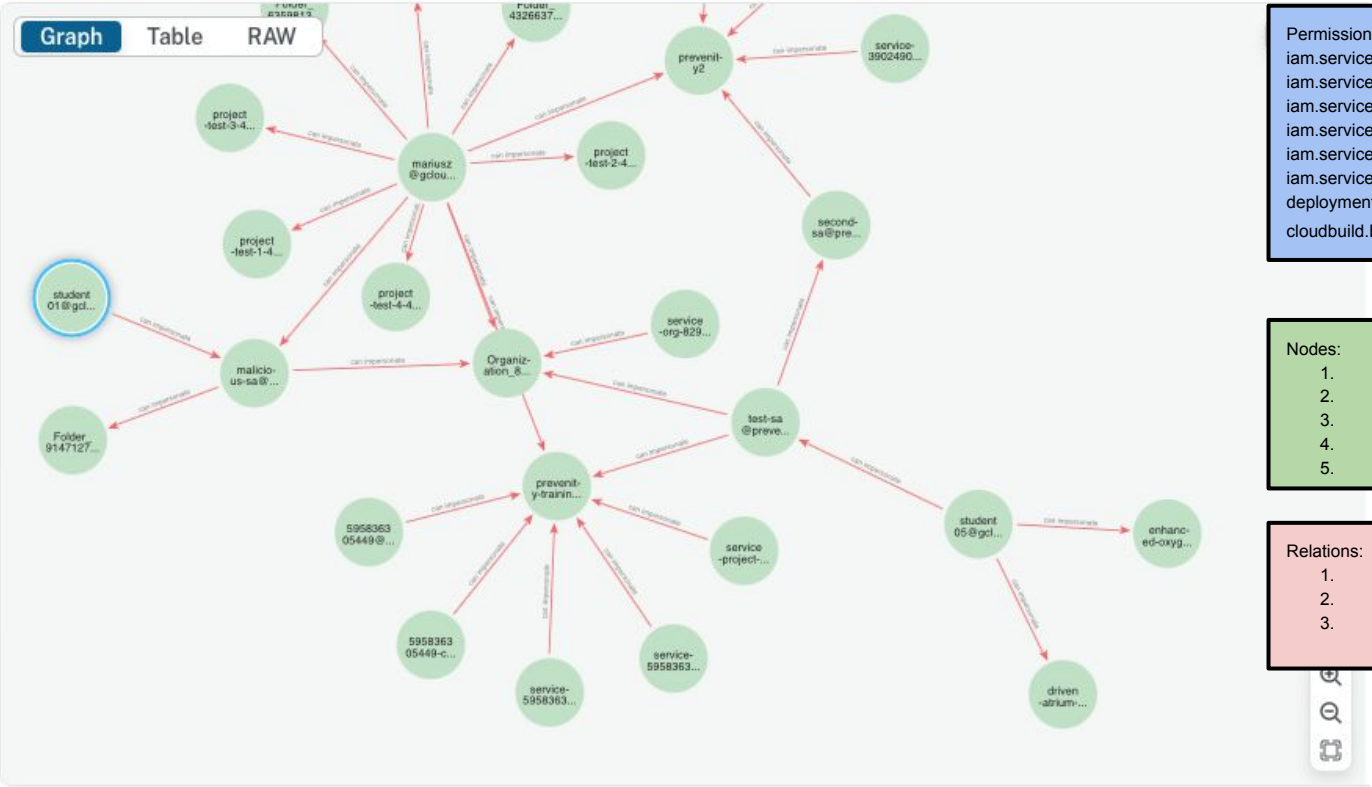
What to do:

- Identify and outline the following permissions-related information
  - build-in roles oraz custom roles
  - scope
  - users/groups/service accounts/principals

Examples created using my tools and Neo4j visualizations.

Impersonation in GCP

```
neo4js MATCH p=()-[:`can_impersonate`]->() RETURN p;
```

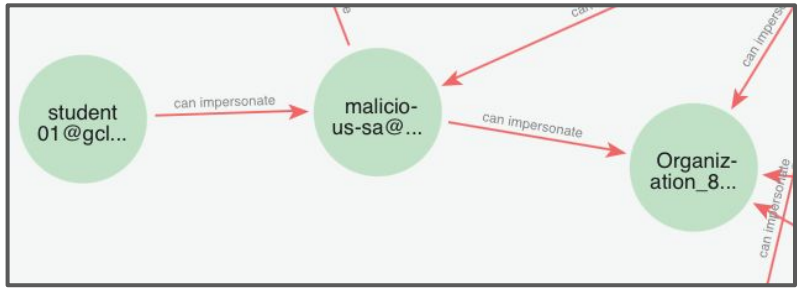


Permissions:
iam.serviceAccounts.signJwt
iam.serviceAccounts.getAccessToken
iam.serviceAccounts.implicitDelegation
iam.serviceAccounts.signBlob
iam.serviceAccounts.getOpenIdToken
iam.serviceAccounts.actAs
deploymentmanager.deployments.create
cloudbuild.builds.create

Nodes:
1. SA1
2. OrgA
3. FolderB
4. ResourceC
5. User1

Relations:
1. SA1 -> FolderB
2. SA2 -> Resource C
3. User1 -> SA1

Impersonation in GCP



1

```
gcloud auth list
        Credentialed Accounts
ACTIVE  ACCOUNT
*       student01@gcloud.domain.com
To set the active account, run:
    $ gcloud config set account `ACCOUNT`
```

2

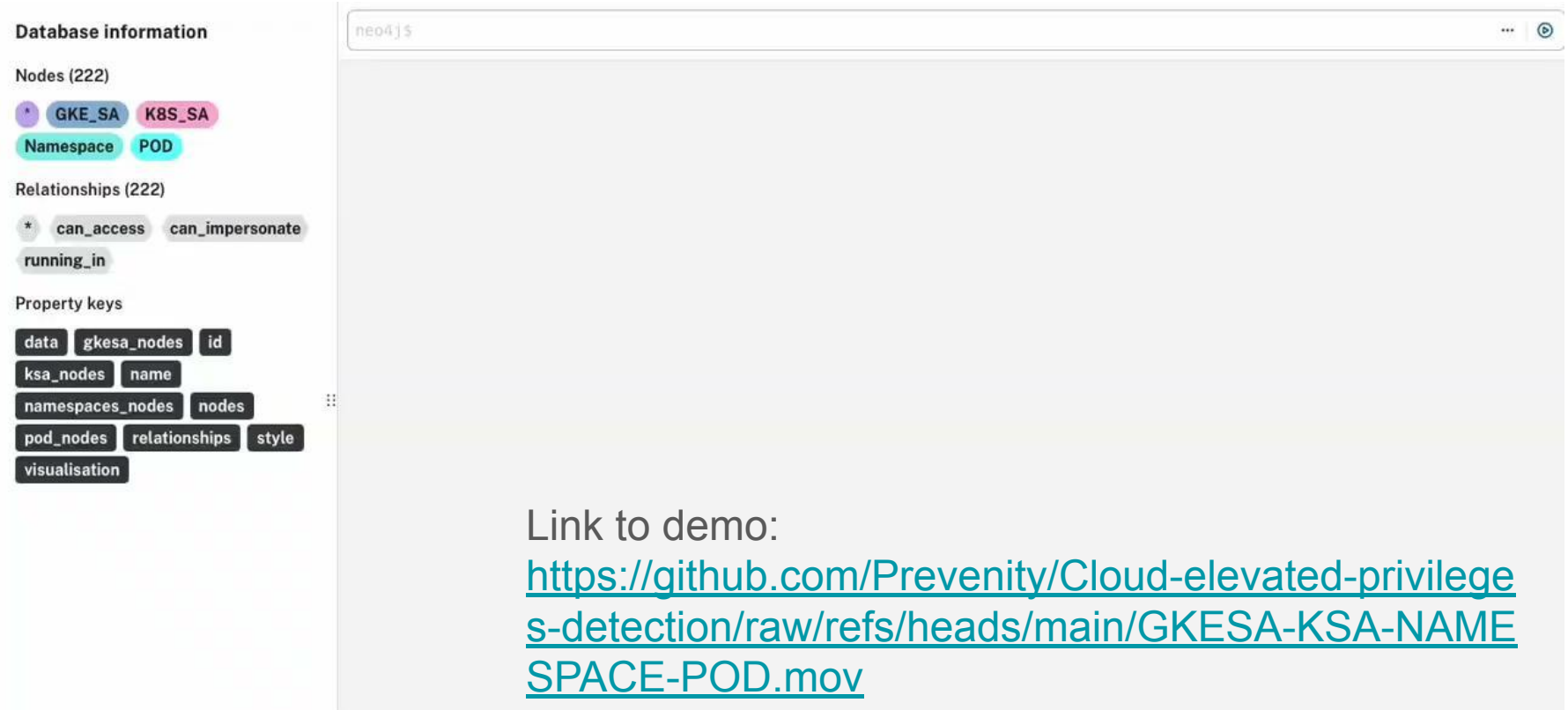
```
gcloud storage gs://trainingbucket/folder001/sourcefile destinationfile --project=example_of_project
  Completed files 0 | 0B
ERROR: (gcloud.storage.cp) HTTPError 403: student01@gcloud.domain.com does not have storage.objects.get access to the Google Cloud Storage
object. Permission 'storage.objects.get' denied on resource (or it may not exist).
```

3

```
gcloud storage gs://trainingbucket/folder001/sourcefile destinationfile --project=example_of_project
--impersonate-service-account=malicious-sa@domain.iam.gserviceaccount.com

WARNING: This command is using service account impersonation. All API calls will be executed as [malicious-sa@domain.iam.gserviceaccount.com.
WARNING: This command is using service account impersonation. All API calls will be executed as [malicious-sa@domain.iam.gserviceaccount.com].
Copying gs://trainingbucket/folder001/source to file://destinationfile
WARNING: This command is using service account impersonation. All API calls will be executed as [malicious-sa@domain.iam.gserviceaccount.com].
WARNING: This command is using service account impersonation. All API calls will be executed as [malicious-sa@domain.iam.gserviceaccount.com].
WARNING: This command is using service account impersonation. All API calls will be executed as [malicious-sa@domain.iam.gserviceaccount.com].
WARNING: This command is using service account impersonation. All API calls will be executed as [malicious-sa@domain.iam.gserviceaccount.com].
  Completed files 1/1 | 2.6kiB/2.6kiB
```

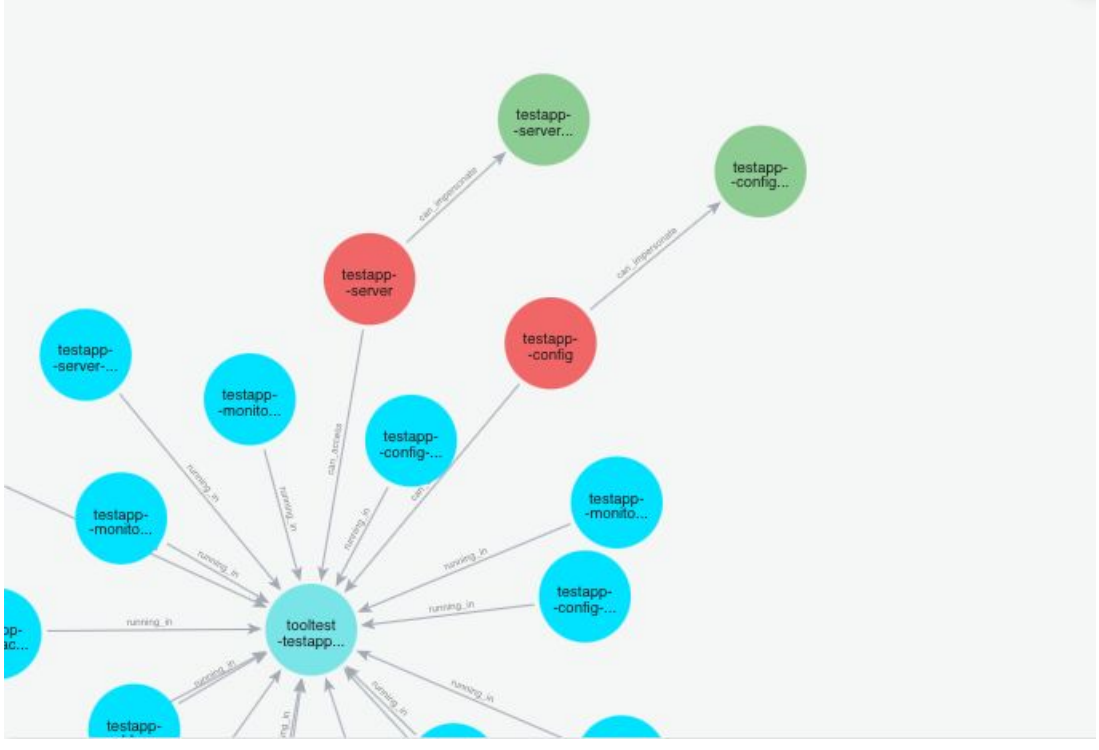
Service accounts in Kubernetes and related service accounts in GCP



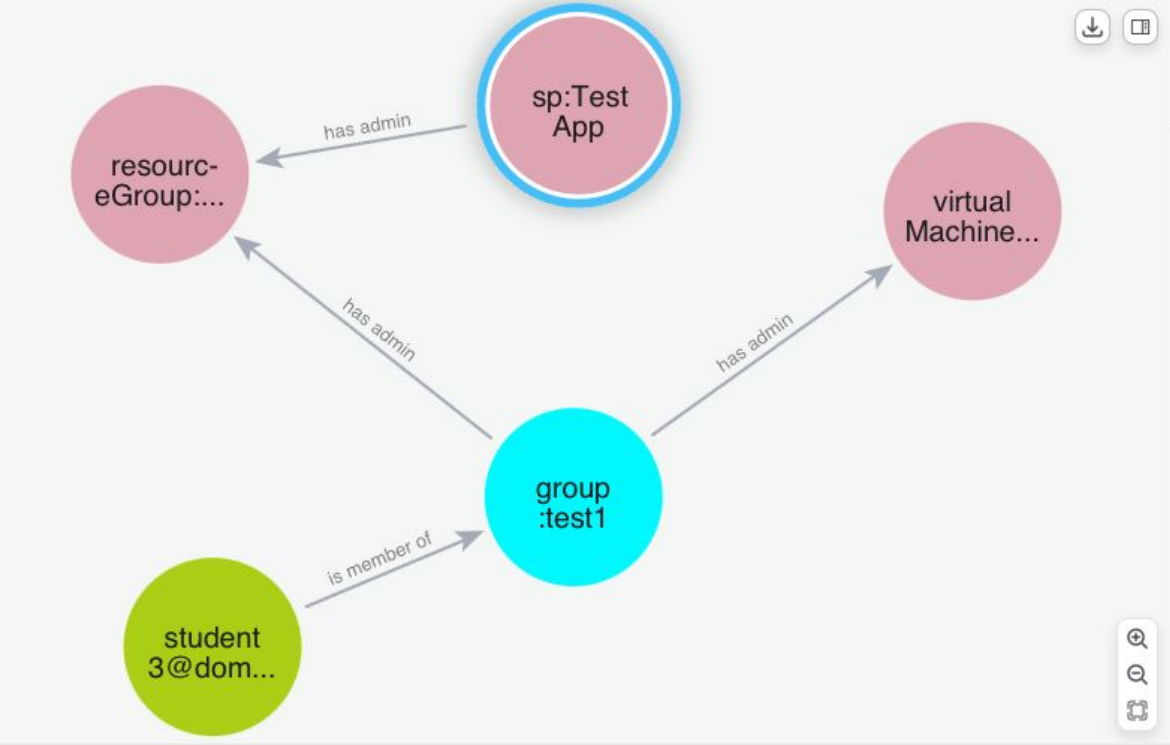
Link to demo:
https://github.com/Prevenity/Cloud-elevated-privileges-detection/raw/refs/heads/main/GKESA-KSA-NAMESPACE-POD.mov

Service accounts cont. (backup - instead of demo)

Relations: GKE SA <-> K8S SA <-> Namespaces <-> Workloads



Azure - accounts with high privileges



Node details

nodes

Key	Value
<id>	4:00911d22-f1ed-43a2-a320-3599ae703046:18
nodes	" sp:TestApp"

Roles Azure:

- Owner
- Contributor
- User Access Administrator
- Role Based Access Control Administrator

Nodes:

1. User1
2. Group1
3. Service Principle A
4. Resource Group B
5. Subscription A
6. Resource D

Recommendations / Summary

- “Pre-production” stage - security by design
  - Policies and Standards
  - Trainings for devops
  - Security requirements
  - Security architecture analysis
  - Threat modeling
  - Risk assessments
  - Security tests and audits
- “Going to production” stage - “last chance to detect”
  - CI/CD pipelines i Security Gates
  - Constraints Policies
- “Production” stage
  - Recommender API / Access Reviews - ID Governance
  - periodic security tests and audits
  - Security monitoring and review
    - Monitoring Owner/Editor for Principles
    - Monitoring “Excessive permissions”
    - Disabling unused Service Accounts
    - Locking default Service Accounts
    - Rotating keys for Service Accounts - if any

Thank you.

Sources

[1] https://www.microsoft.com/en-us/security/security-insider/microsoft-digital-defense-report-2023

[2] https://www.microsoft.com/en-us/security/security-insider/intelligence-reports/microsoft-digital-defense-report-2024

[3] https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-2024

[4] https://www.wiz.io/blog/midnight-blizzard-microsoft-breach-analysis-and-best-practices

[5] https://www.clouddefense.ai/breaches/2024

[6] https://www.theregister.com/2022/07/18/apac_tech_news_roundup/

[7] https://www.cyberark.com/resources/blog/unpacking-the-uber-breach

[8] https://github.com/Prevenity/Cloud-elevated-privileges-detection [scripts used in this presentation]

[9] https://learn.microsoft.com/en-us/azure/role-based-access-control/best-practices

[10] https://cloud.google.com/iam/docs/best-practices-service-accounts

[11] https://neo4j.com/