AYI NEDJIMI Consultants

Cloud Pentest AWS avec Pacu et CloudFox : Le Guide

Catégorie : Cloud Security Lecture : 8 min Publié le : 06/03/2026 Auteur : Ayi NEDJIMI

Méthodologie de pentest cloud AWS avancée avec Pacu et CloudFox : énumération IAM, escalade de privilèges, mouvement latéral et exfiltration.

Résumé exécutif

Ce guide technique détaille la méthodologie de pentest AWS utilisant Pacu et CloudFox, couvrant la reconnaissance, l'énumération IAM, l'escalade de privilèges, le mouvement latéral et l'exfiltration de données dans les environnements cloud Amazon.
Le pentest cloud AWS est une discipline radicalement différente du pentest infrastructure classique. Pas de Nmap, pas de Metasploit sur des ports ouverts, pas d'exploitation de services réseau vulnérables. L'attaquant cloud opère principalement via les API AWS, exploitant des misconfigurations IAM, des permissions excessives et des interactions entre services pour escalader ses privilèges et atteindre les données critiques. Après avoir conduit plus de cinquante pentests AWS pour des organisations de toutes tailles, je partage dans ce guide la méthodologie structurée que nous utilisons en engagement réel, articulée autour de deux outils majeurs — Pacu pour l'exploitation automatisée et CloudFox pour la reconnaissance contextuelle — complétée par des techniques manuelles et des scripts custom qui couvrent les angles morts de chaque outil utilisé en conditions opérationnelles de pentest cloud avancé.

Comment structurer un pentest AWS en phases ?

Notre méthodologie de pentest AWS suit cinq phases calquées sur le framework MITRE ATT&CK Cloud. **Phase 1 — Reconnaissance** : identification des assets AWS exposés (S3 buckets publics, API Gateway endpoints, CloudFront distributions, IP ranges). **Phase 2 — Initial Access** : exploitation de credentials fuités, SSRF vers le metadata service IMDS, ou utilisation de credentials fournis (assumed breach scenario). **Phase 3 — Enumeration** : cartographie complète de l'environnement via les API AWS (IAM, EC2, S3, Lambda, RDS, etc.). **Phase 4 — Privilege Escalation** : exploitation des chemins d'escalade IAM pour obtenir des permissions supérieures. **Phase 5 — Lateral Movement & Data Access** : pivot entre services et comptes, accès aux données sensibles.

Les vecteurs d'escalade de privilèges AWS sont exhaustivement documentés dans notre article sur escalades de privilèges AWS. Cette phase est souvent la plus critique du pentest et nécessite une compréhension approfondie du modèle IAM AWS.

Phase	Outil principal	Techniques clés	Durée typique
Reconnaissance	CloudFox, Prowler	Asset discovery, DNS enum	1-2 jours
Initial Access	Manuel, truffleHog	Cred hunting, SSRF	1 jour
Enumeration	CloudFox, Pacu	IAM enum, service mapping	2-3 jours
Privilege Escalation	Pacu, PMAPPER	IAM paths, assume role	2-3 jours
Lateral Movement	Pacu, AWS CLI	Cross-account, Lambda pivot	2-3 jours

Mon avis : Le scénario "assumed breach" où le pentester reçoit des credentials IAM avec des permissions limitées est de loin le plus réaliste et le plus utile. Tester uniquement depuis l'extérieur sans credentials rate 80% de la surface d'attaque réelle, car la majorité des compromissions cloud démarrent avec des credentials fuités ou une SSRF.

Quelles techniques de reconnaissance CloudFox utiliser ?

CloudFox est un outil de reconnaissance contextuelle développé par Bishop Fox qui aide à identifier les chemins d'attaque dans les environnements AWS. Contrairement à Pacu qui est orienté exploitation, CloudFox se concentre sur la cartographie et l'identification des points d'intérêt. Les modules les plus utiles incluent : `cloudfox aws permissions` (liste les permissions effectives de chaque principal), `cloudfox aws access-keys` (identifie les clés d'accès et leur âge), `cloudfox aws endpoints` (découvre les services exposés), `cloudfox aws secrets` (liste les secrets dans Secrets Manager et SSM), et `cloudfox aws role-trusts` (cartographie les relations de confiance entre rôles).

Le module `cloudfox aws iam-simulator` est particulièrement puissant : il utilise l'*IAM Policy Simulator* d'AWS pour tester les permissions effectives d'un principal contre toutes les actions et ressources. Cela permet d'identifier des permissions cachées non évidentes dans les policies, notamment celles héritées des policies de groupe ou des **permission boundaries**. La documentation sur AWS Security décrit le modèle d'évaluation des policies IAM que CloudFox exploite pour ses analyses.

Comment exploiter Pacu pour l'escalade de privilèges ?

Pacu est le framework d'exploitation AWS open-source de référence, développé par Rhino Security Labs. Il fonctionne comme un Metasploit pour AWS avec des modules organisés par phase d'attaque. Pour l'*escalade de privilèges*, les modules clés sont : `iam__privesc_scan` (identifie automatiquement les 21+ chemins d'escalade connus), `iam__enum_permissions` (énumère les permissions via brute-force API), et `lambda__backdoor_new_roles` (exploite la permission iam:PassRole pour créer des rôles avec des policies admin).

Les chemins d'escalade les plus fréquemment exploitables en pentest incluent : **iam:CreatePolicyVersion** (créer une nouvelle version de policy avec des permissions admin), **iam:AttachUserPolicy/AttachRolePolicy** (s'attacher une policy admin existante), **iam:PassRole + lambda:CreateFunction + lambda:InvokeFunction** (créer une Lambda avec un rôle admin),

sts:AssumeRole (assumer un rôle plus privilégié via une trust policy permissive), et **ec2:RunInstances + iam:PassRole** (lancer une instance avec un rôle admin et y accéder via SSM). Notre article sur les escalade de privilèges IAM cloud détaille ces techniques avec des exemples de commandes.

Lors d'un pentest AWS pour un groupe média, nous avons reçu des credentials IAM avec uniquement les permissions de lecture sur S3. En utilisant CloudFox pour cartographier les rôles IAM et leurs trust policies, nous avons identifié un rôle Lambda avec la permission iam:PassRole sans condition de ressource. Via la chaîne lambda:CreateFunction + iam:PassRole + lambda:InvokeFunction, nous avons créé une Lambda associée au rôle admin du compte. En 4 heures, nous sommes passés de read-only S3 à admin complet du compte contenant les données de 12 millions d'utilisateurs.

Quelles sont les techniques de mouvement latéral AWS ?

Le mouvement latéral dans AWS exploite principalement trois vecteurs : les **trust policies cross-account** (AssumeRole vers d'autres comptes AWS de l'organisation), les **resource policies permissives** (accès à des buckets S3, queues SQS ou topics SNS d'autres comptes), et le **pivot via les services** (utiliser Lambda, CodeBuild ou EC2 avec des rôles dans d'autres comptes). Pacu facilite ce mouvement avec les modules `iam__enum_roles` (discovery des rôles assumables cross-account) et `organizations__enum` (cartographie de l'organisation AWS).

Une technique avancée consiste à exploiter les **Lambda Layers** partagées entre comptes pour injecter du code malveillant qui s'exécutera dans le contexte IAM de la fonction cible. De même, les **AMI partagées** peuvent être backdoorées pour compromettre les instances lancées dans d'autres comptes. Le pentest des pipelines CI/CD via secrets sprawl et collecte révèle souvent des credentials de comptes de production stockés dans des environnements de développement moins protégés.

La cartographie des chemins d'attaque RBAC Kubernetes sur EKS, couverte dans attaques RBAC Kubernetes, ajoute une dimension supplémentaire au mouvement latéral dans les environnements AWS qui utilisent des clusters EKS. La vérification des configurations Terraform via audit Terraform compliance peut révéler des misconfigurations exploitables avant même le déploiement.

Comment détecter et exfiltrer les données sensibles ?

L'accès aux données est l'objectif final du pentest. Sur AWS, les données sensibles se trouvent principalement dans : **S3 buckets** (documents, backups, exports), **RDS/DynamoDB** (bases de données applicatives), **Secrets Manager/SSM Parameter Store** (credentials et configurations), **CloudWatch Logs** (logs applicatifs contenant des PII), et **EBS Snapshots** (snapshots de volumes qui peuvent contenir des données en clair). Pacu propose des modules pour chaque source : `s3__download_bucket`, `rds__explore_snapshots`, `ssm__download_parameters`.

L'exfiltration simulée (dans le cadre du pentest) doit démontrer la faisabilité sans réellement exfiltrer des données sensibles. Documentez les chemins d'accès, le volume de données accessibles et la classification des données trouvées. L'ANSSI fournit des recommandations sur les tests d'intrusion respectant le cadre réglementaire français et les limites à ne pas dépasser.

À retenir : Un pentest AWS efficace combine reconnaissance contextuelle (CloudFox), exploitation automatisée (Pacu) et techniques manuelles. L'escalade de privilèges IAM est presque toujours le pivot central du test. Documentez chaque étape avec des preuves reproductibles et fournissez des recommandations de remédiation priorisées par exploitabilité plutôt que par sévérité théorique CVSS.

Faut-il tester les guardrails et la détection ?

Un pentest AWS complet inclut l'évaluation des contrôles défensifs. Testez si **GuardDuty** détecte vos activités (credential exfiltration, port scanning depuis EC2, accès S3 depuis des IP suspectes). Vérifiez si les **SCP** bloquent effectivement les actions interdites. Évaluez le temps de détection et de réponse de l'équipe SOC. Ce volet "purple team" est souvent plus utile que l'exploitation elle-même, car il révèle les lacunes défensives concrètes. Documentez chaque contrôle testé avec son résultat (détecté/non détecté, temps de détection, réponse apportée).

La documentation des findings de pentest AWS nécessite une rigueur particulière pour la reproductibilité. Chaque finding doit inclure : la commande Pacu ou CloudFox exacte utilisée avec les paramètres, le contexte IAM du principal exploité (ARN, permissions effectives), les étapes de reproduction numérotées, une capture d'écran ou un extrait de log prouvant l'exploitation, l'impact business évalué dans le contexte du client, et une recommandation de remédiation avec la commande AWS CLI ou le snippet Terraform pour corriger la misconfiguration. Pour les chemins d'escalade multi-étapes, documentez chaque étape intermédiaire avec son propre niveau de preuve, car le client devra corriger chaque maillon de la chaîne indépendamment. Les findings doivent être priorisés par exploitabilité réelle sur l'environnement du client plutôt que par sévérité CVSS théorique, en mettant en avant les chemins d'attaque complets qui mènent aux données critiques identifiées lors de la phase de cadrage du pentest avec les parties prenantes business.

Les outils complémentaires comme **ScoutSuite** de NCC Group fournissent un audit de conformité complet exportable en HTML qui sert de base au rapport d'état des lieux de la configuration de sécurité AWS. Combinez-le avec les findings d'exploitation Pacu pour un rapport qui couvre à la fois les misconfigurations théoriques et les chemins d'attaque effectivement exploitables, donnant au client une vision complète de sa posture et des priorités de remédiation.

Votre dernier pentest AWS a-t-il réellement testé les chemins d'escalade IAM, ou s'est-il limité à scanner les misconfigurations S3 publiques que n'importe quel outil CSPM détecte automatiquement ?

Comment automatiser la reconnaissance avec des scripts custom ?

Au-delà de Pacu et CloudFox, les pentesters AWS efficaces développent des scripts custom pour couvrir les angles morts des outils standards. Un script d'énumération des **resource policies** parcourt tous les buckets S3, queues SQS, topics SNS, clés KMS et Lambda functions pour identifier les policies qui autorisent des accès cross-account ou publics. Un script de découverte des **trust relationships** map l'ensemble des rôles IAM avec leurs trust policies pour construire un graphe de confiance visualisable dans Neo4j ou Bloodhound. Un script d'extraction des **user-data** EC2 récupère les scripts de démarrage de toutes les instances accessibles, souvent truffés de credentials en clair pour la configuration initiale.

L'outil **Prowler** complète l'arsenal avec un audit de conformité automatisé contre les CIS Benchmarks AWS, détectant les misconfigurations sans les exploiter — utile pour le reporting exhaustif du pentest. Pour l'analyse des permissions effectives, **PMAPPER** (Principal Mapper) construit un graphe des chemins d'escalade IAM plus exhaustif que le module Pacu car il simule toutes les combinaisons possibles de permissions via l'IAM Policy Simulator. En combinant ces outils avec des scripts custom adaptés au contexte spécifique du client, le pentester couvre une surface d'attaque bien plus large que ce qu'un outil unique peut offrir, et produit des findings plus pertinents car contextualisés à l'architecture réelle de l'environnement testé.

L integration du pentest AWS dans un programme de securite continu transforme un exercice ponctuel en une amelioration mesurable. Planifiez des pentests trimestriels focalises sur des perimetres differents : IAM et permissions un trimestre, configurations reseau le suivant, donnees et chiffrement le troisieme, pipelines CI/CD le quatrieme. Cette rotation garantit une couverture complete annuelle et permet de mesurer la progression de la remediation entre chaque test d intrusion realise sur votre environnement cloud.

Sources et références : CISA · Cloud Security Alliance

Conclusion : livrer un rapport de pentest AWS actionnable

Le rapport de pentest AWS doit aller au-delà de la liste de vulnérabilités. Structurez-le autour de la kill chain exploitée : initial access, escalade, mouvement latéral, objectif atteint. Pour chaque étape, détaillez la technique utilisée, les commandes exactes (Pacu/CloudFox), la preuve d'exploitation et la recommandation de remédiation. Classez les findings par exploitabilité et impact business plutôt que par sévérité CVSS générique. Incluez un executive summary qui traduit les findings techniques en risques business compréhensibles par le COMEX, et une roadmap de remédiation priorisée sur 30, 60 et 90 jours.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

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