AYI NEDJIMI Consultants

Cloud Pentest Azure : Exploitation Misconfiguration

Catégorie : Cloud Security Lecture : 8 min Publié le : 07/03/2026 Auteur : Ayi NEDJIMI

Techniques avancées de pentest Azure : exploitation des misconfigurations Entra ID, RBAC Azure, Storage Accounts, Key Vault et mouvement latéral.

Résumé exécutif

Ce guide technique couvre les méthodologies de pentest Azure : exploitation d'Entra ID, RBAC Azure, Storage Accounts et Key Vault. Il détaille les outils ROADtools, AzureHound et MicroBurst pour l'énumération et l'exploitation des misconfigurations.
Azure est souvent perçu comme plus sécurisé par défaut qu'AWS grâce à son intégration native avec Active Directory et ses guardrails Entra ID. Cette perception est trompeuse et dangereuse. L'écosystème Azure présente des surfaces d'attaque spécifiques que les pentesters formés uniquement sur AWS ne connaissent pas, et les misconfigurations les plus critiques se cachent dans l'interaction entre Entra ID et les ressources Azure, dans les assignations de rôles RBAC héritées, et dans les permissions des applications et service principals que les équipes déploient sans comprendre leur portée réelle. Après avoir mené des dizaines de pentests Azure pour des entreprises européennes largement investies dans l'écosystème Microsoft, je partage les techniques d'exploitation les plus efficaces, les outils spécialisés indispensables et les misconfigurations récurrentes qui offrent systématiquement des chemins d'escalade exploitables vers les données les plus sensibles des organisations ciblées.

Comment énumérer un tenant Azure efficacement ?

L'énumération Azure commence par la reconnaissance du **tenant Entra ID**. L'outil **AADInternals** permet de découvrir des informations sur le tenant à partir d'un simple nom de domaine : existence du tenant, services configurés (Exchange Online, SharePoint, Teams), politiques d'authentification. **ROADtools** (développé par Dirk-jan Mollema) est l'outil de référence pour l'énumération approfondie d'Entra ID : utilisateurs, groupes, applications, service principals, rôles d'annuaire, et surtout les permissions déléguées et applicatives de chaque application.

La commande `roadrecon gather` collecte l'intégralité du graphe Entra ID via l'API Microsoft Graph, et `roadrecon gui` offre une interface web pour explorer les données. **AzureHound** (l'extension Azure de BloodHound) va plus loin en cartographiant les chemins d'escalade entre les objets Entra ID et les ressources Azure, visualisant les relations de type "cet utilisateur est Owner de ce groupe, qui est Contributor sur cette subscription, qui contient un Key Vault avec des secrets de production".

Pour les techniques d'escalade IAM cross-cloud, notre article sur escalade de privilèges IAM cloud fournit des parallèles utiles entre les modèles AWS et Azure.

Outil	Fonction	Prérequis	Phase
AADInternals	Tenant recon	Aucun	Reconnaissance
ROADtools	Entra ID enum	Credentials utilisateur	Énumération
AzureHound	Attack path mapping	Credentials utilisateur	Énumération
MicroBurst	Azure resource enum	Az module	Exploitation
PowerZure	Azure exploitation	Az PowerShell	Exploitation
TokenTactics	Token manipulation	Token initial	Latéral movement

Mon avis : Le pentest Azure est fondamentalement un pentest Active Directory étendu au cloud. Si votre pentester ne maîtrise pas les concepts d'Entra ID (application registrations, service principals, managed identities, OAuth scopes), il passera à côté de 70% des chemins d'escalade exploitables. Les compétences AD on-premises seules ne suffisent plus.

Quelles misconfigurations Entra ID sont exploitables ?

Les misconfigurations Entra ID les plus critiques incluent : **Application Registrations avec permissions excessives** (une app avec Directory.ReadWrite.All ou RoleManagement.ReadWrite.Directory permet de s'attribuer des rôles Global Admin), **consentement admin accordé sans revue** (des applications tierces avec des permissions sur les mails, fichiers et calendriers de tous les utilisateurs), **absence de Conditional Access** (pas de MFA, pas de restriction géographique, pas de vérification de conformité du device), et **users autorisés à créer des applications** (paramètre par défaut qui permet à tout utilisateur de créer des app registrations potentiellement dangereuses).

Le vecteur le plus puissant est l'*illicit consent grant* : un attaquant crée une application malveillante et trompe un admin pour qu'il accorde le consentement admin. L'application obtient alors un accès permanent aux données du tenant via l'API Graph, indépendamment des changements de mot de passe de l'admin. La documentation de Azure Defender for Cloud détaille les bonnes pratiques de sécurisation d'Entra ID, et l'ANSSI propose des recommandations complémentaires sur la sécurisation des annuaires cloud.

Comment exploiter le RBAC Azure pour escalader ?

Le **RBAC Azure** fonctionne différemment de l'IAM AWS. Les rôles sont assignés à des *scopes* hiérarchiques : Management Group, Subscription, Resource Group, Resource. Un rôle Contributor au niveau Subscription s'hérite sur tous les Resource Groups et ressources en dessous. Les chemins d'escalade classiques incluent : **Contributor sur un Automation Account** (exécuter des runbooks avec une Managed Identity privilégiée), **Contributor sur une VM** (exécuter des commandes via Run Command avec la Managed Identity de la VM), **Contributor sur un Logic App** (modifier le workflow pour accéder aux connecteurs configurés avec des identités privilégiées).

L'outil **MicroBurst** de NetSPI automatise la découverte de ces chemins avec des modules comme `Get-AzPasswords` (extraction de credentials depuis les Automation Accounts, App Services, et Key Vaults accessibles) et `Get-AzDomainInfo` (énumération complète des ressources et permissions). Pour les parallèles avec AWS, notre article sur les escalades de privilèges AWS montre des techniques similaires adaptées à l'écosystème Amazon.

Lors d'un pentest Azure pour une banque européenne, nous avons obtenu un accès initial via un compte développeur avec le rôle Reader sur une subscription. Via AzureHound, nous avons identifié que ce compte était membre d'un groupe Entra ID avec le rôle Contributor sur un Automation Account. Ce compte d'automatisation disposait d'une Managed Identity avec le rôle Owner au niveau Management Group. En créant un runbook PowerShell exécuté avec cette identité, nous avons obtenu le contrôle total de l'ensemble des 23 subscriptions Azure de la banque en trois heures depuis un simple accès Reader initial.

Pourquoi les Storage Accounts sont des cibles prioritaires ?

Les **Azure Storage Accounts** concentrent souvent les données les plus sensibles : blobs contenant des backups de bases de données, file shares avec des documents internes, table storage avec des logs applicatifs. Les misconfigurations courantes incluent : **accès public activé** (anonymous access sur les containers), **Shared Access Signatures (SAS) trop larges** (SAS au niveau du compte avec tous les services et permissions, sans restriction IP, avec une expiration de plusieurs années), **Storage Account keys non rotées** (les deux clés d'accès donnent un accès total et ne sont jamais rotées), et **absence de Private Endpoints** (accès via l'IP publique au lieu du réseau privé).

L'exfiltration depuis un Storage Account compromis se fait via `azcopy` ou l'API REST avec un SAS token. Pour les pentests CI/CD sur Azure DevOps, notre article sur attaques CI/CD GitOps couvre les techniques d'exploitation des pipelines qui stockent souvent des SAS tokens en variables de pipeline insuffisamment protégées.

Comment exploiter les Managed Identities ?

Les **Managed Identities** (System-assigned et User-assigned) sont le mécanisme Azure pour éliminer les credentials dans le code. Une VM, une Function App, un App Service ou un AKS pod avec une Managed Identity peut obtenir un token d'accès via l'endpoint IMDS (169.254.169.254). En pentest, si vous compromettez une ressource avec une Managed Identity, vous héritez de toutes ses permissions RBAC. Les modules Pacu et MicroBurst automatisent la récupération de ces tokens et leur utilisation pour pivoter vers d'autres ressources Azure.

La gestion des secrets dans les configurations IaC est couverte dans notre article sur secrets sprawl et collecte, et les techniques d'exploitation via audit Terraform compliance complètent cette analyse avec les vecteurs applicatifs.

À retenir : Le pentest Azure se distingue par l'importance centrale d'Entra ID et du RBAC hiérarchique. Les outils ROADtools et AzureHound sont indispensables pour cartographier les chemins d'escalade. Les Managed Identities et les Automation Accounts sont les pivots les plus fréquemment exploitables pour escalader depuis un accès limité vers le contrôle total du tenant.

Faut-il tester la séparation tenant et les B2B trusts ?

Un pentest Azure complet doit évaluer la sécurité des frontières du tenant. Les **B2B Guest Users** invités depuis d'autres tenants conservent souvent des permissions excessives dans le tenant hôte. Les **Cross-tenant access settings** configurés trop largement peuvent permettre à des utilisateurs d'un tenant partenaire d'accéder à des ressources sensibles. Testez également la séparation entre les environnements Azure et Microsoft 365 : un accès à SharePoint Online peut révéler des documents contenant des credentials Azure, et inversement un accès Azure peut permettre de lire les mails via l'API Graph si les permissions de la Managed Identity le permettent.

Les **Conditional Access Policies** Azure sont souvent la première barrière que le pentester doit évaluer et potentiellement contourner. Une politique de Conditional Access bien configurée peut bloquer de nombreux vecteurs d'attaque : imposer le MFA élimine l'exploitation de credentials volés par phishing simple, restreindre les connexions par localisation bloque les attaquants distants, et exiger la conformité du device empêche l'utilisation de machines non gérées. Cependant, les exclusions de Conditional Access sont fréquentes et exploitables : les Service Principals sont souvent exemptés du MFA, les comptes break-glass sont parfois mal protégés, et les applications legacy qui ne supportent pas l'authentification moderne sont autorisées en basic auth. Le pentester doit cartographier toutes les politiques Conditional Access et leurs exclusions pour identifier les chemins de contournement, en utilisant ROADtools pour énumérer les politiques configurées via l'API Graph sans disposer de privilèges d'administrateur.

L'utilisation de **TokenTactics** permet de manipuler les tokens OAuth2 Azure pour le mouvement latéral entre les applications. Un token d'accès obtenu pour une ressource peut parfois être échangé contre un token pour une autre ressource si les scopes le permettent, ou un refresh token peut être utilisé pour obtenir de nouveaux access tokens avec des scopes différents. Cette technique est particulièrement efficace contre les configurations qui n'imposent pas le binding de token au device.

Votre dernier pentest Azure a-t-il réellement couvert la surface d'attaque Entra ID, ou s'est-il limité aux ressources Azure Resource Manager sans explorer les chemins d'escalade via les applications et service principals ?

Comment exploiter les Azure Functions et Logic Apps ?

Les **Azure Functions** et **Logic Apps** sont des vecteurs d'exploitation souvent négligés lors des pentests Azure. Une Azure Function avec une Managed Identity peut être modifiée pour exécuter du code arbitraire dans le contexte de sécurité de cette identité. Si la Function dispose d'un rôle Contributor sur la subscription, le pentester obtient un accès large en modifiant simplement le code de la fonction via le portail ou l'API. Les Logic Apps sont encore plus intéressantes car elles stockent souvent des connecteurs pré-authentifiés vers des services externes : Office 365, SharePoint, SQL, Key Vault. Modifier le workflow d'une Logic App permet d'exploiter ces connecteurs sans connaître les credentials sous-jacents.

Les **App Services** Azure présentent leurs propres vulnérabilités. Le *Kudu console* (accessible via {nom-app}.scm.azurewebsites.net) fournit un accès shell au conteneur de l'App Service. Si vous obtenez les **deployment credentials** (disponibles dans le profil de publication), vous pouvez déployer du code malveillant directement. Les variables d'environnement de l'App Service contiennent souvent des connection strings vers les bases de données et les Storage Accounts en clair. Utilisez PowerZure pour automatiser l'extraction de ces credentials et le pivot vers les ressources connectées. Cette technique est particulièrement dévastatrice dans les architectures microservices où chaque App Service a accès à des ressources backend différentes.

L integration de l intelligence artificielle dans les outils de pentest Azure accelere la reconnaissance et l identification des chemins d escalade. Des outils comme GraphRunner explorent automatiquement les relations entre les objets Entra ID pour identifier des chemins d escalade complexes impliquant des chaines de trois a cinq etapes que les analystes humains manqueraient lors d une investigation manuelle. La combinaison de ces outils automatises avec l expertise humaine produit des resultats superieurs a ce que chaque approche accomplit seule.

Sources et références : CISA · Cloud Security Alliance

Conclusion : structurer un rapport de pentest Azure

Le rapport de pentest Azure doit documenter la kill chain complète depuis l'accès initial jusqu'aux données sensibles atteintes. Cartographiez visuellement les chemins d'escalade avec des exports AzureHound. Détaillez chaque misconfiguration avec la commande Azure CLI ou PowerShell pour la reproduire et la corriger. Priorisez les remédiations par impact et facilité de correction : les Conditional Access policies sont rapides à déployer et bloquent de nombreux vecteurs, tandis que la refonte complète du RBAC peut prendre des mois. Proposez un plan de remédiation en trois vagues avec des quick wins immédiats et des chantiers structurants à moyen terme.

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

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