Chapitre 3
Auditer l'Active Directory

1. Méthodologie

Avant de plonger dans les aspects techniques de l'audit d'un Active Directory, il est essentiel de replacer la démarche dans un cadre méthodologique clair. Un audit sérieux ne se résume pas à l'exécution de quelques outils ou à la recherche opportuniste de vulnérabilités ; il s'agit d'un processus structuré, pensé pour être à la fois exhaustif, reproductible et compréhensible par toutes les parties prenantes, qu'il s'agisse d'équipes techniques ou de directions stratégiques.

C'est pourquoi nous allons, dans un premier temps, présenter les principales méthodologies qui servent de fil conducteur aux tests d'intrusion et aux audits de sécurité. Ces cadres donnent une logique à l'évaluation, d'en tracer les étapes et d'établir un langage commun entre auditeurs et audités.

1.1 Le PTES : un standard de référence

Le PTES (*Penetration Testing Execution Standard*) est sans doute l'un des cadres les plus connus dans le domaine du *pentest*. Il définit un ensemble d'étapes qui structurent la réalisation d'un audit de sécurité, de la préparation à la restitution.

Le PTES s'articule en plusieurs phases :

- **Pré-engagement** : définition du périmètre, des objectifs et des règles de l'audit.
- **Collecte d'informations** : phase de reconnaissance où l'on rassemble le maximum de données disponibles sur l'environnement cible (cartographie réseau, inventaire des machines, découverte de l'Active Directory).
- **Modélisation des menaces** : identification des vecteurs d'attaque les plus pertinents en fonction du contexte.
- **Analyse des vulnérabilités** : repérage des faiblesses techniques ou organisationnelles exploitables.
- **Exploitation** : tentative de mise à profit des vulnérabilités découvertes pour obtenir un accès ou une élévation de privilèges.
- **Post-exploitation** : consolidation des accès, recherche de mouvements latéraux, extraction d'informations sensibles.
- **Reporting** : restitution claire et documentée des constats, avec recommandations adaptées.

Dans le contexte d'un Active Directory, le PTES trouve une application particulièrement pertinente. La phase de collecte d'informations comprend notamment l'énumération des comptes, des groupes, des relations d'approbation et des stratégies de sécurité. La modélisation des menaces portera sur les abus possibles de configurations inadaptées, de failles dans la gestion des mots de passe ou de délégations mal maîtrisées. La phase de post-exploitation consiste alors à mettre en œuvre des techniques comme le Pass-the-Hash, l'extraction de tickets Kerberos ou l'abus de Service Principal Names (SPN) afin de progresser dans l'environnement.

1.2 La Cyber Kill Chain : penser comme un attaquant

Développée par Lockheed Martin, la Cyber Kill Chain est un modèle qui décrit les étapes d'une attaque ciblée. Contrairement au PTES, qui est conçu comme un guide de bonnes pratiques pour l'auditeur, la Kill Chain se place davantage du point de vue de l'attaquant.

Elle comprend sept étapes successives :

- **Reconnaissance** : identification des cibles et collecte d'informations.
- **Weaponization** : préparation de l'attaque (exploitation, payload).
- **Delivery** : livraison de la charge malveillante (phishing, exploitation réseau, etc.).
- **Exploitation** : exécution du code ou exploitation d'une vulnérabilité.
- **Installation** : mise en place de portes dérobées ou d'outils de persistance.
- **Command & Control (C2)** : établissement d'un canal de communication entre l'attaquant et l'environnement compromis.
- **Actions on Objectives** : réalisation des objectifs (vol de données, élévation de privilèges, sabotage).

Dans le cadre d'un audit Red Team, la Cyber Kill Chain se révèle particulièrement utile, car elle permet de simuler pas à pas les tactiques adverses réelles et d'évaluer la capacité de détection et de réaction de l'organisation face à une attaque complète.

1.3 Au-delà des modèles : vers une démarche structurée

Si le PTES et la Cyber Kill Chain constituent des cadres de référence solides, ils peuvent être utilement complétés par d'autres approches. Le NIST SP 800-115 propose un guide méthodique pour la réalisation de tests de sécurité, tandis que la base de connaissances MITRE ATT&CK dresse une cartographie des tactiques et techniques employées par les attaquants, particulièrement pertinente pour anticiper et illustrer des scénarios d'intrusion dans un Active Directory. Au-delà de leur contenu, ces référentiels apportent une structure et une vision d'ensemble qui renforcent la portée de l'audit. L'adoption d'une telle démarche méthodologique ne doit pas être perçue comme une contrainte ; elle constitue au contraire une garantie de rigueur et de cohérence. Elle assure une couverture complète des tests, évite une approche morcelée limitée aux aspects purement techniques et facilite la communication avec les décideurs, en transformant des constats opérationnels en éléments compréhensibles sur le plan stratégique.

L'audit d'un Active Directory dépasse ainsi le simple usage d'outils pour s'inscrire dans une évaluation globale et structurée de la sécurité de l'organisation.

2. Collecte d'informations

Dans toute démarche d'audit, la reconnaissance constitue la première étape, et sans doute l'une des plus décisives. Elle est au cœur des méthodologies présentées précédemment : c'est la phase initiale du PTES et de la Cyber Kill Chain. À ce stade, l'auditeur ne cherche pas encore à exploiter de vulnérabilités, mais à collecter le maximum d'informations sur l'environnement cible. Ces données brutes, souvent obtenues de manière passive ou anonyme, serviront de base à toutes les étapes ultérieures. Plus la reconnaissance est rigoureuse et complète, plus l'audit gagne en efficacité.

2.1 Découverte du réseau et identification des services

2.1.1 Découverte

La première tâche consiste généralement à **cartographier le réseau** afin d'identifier les systèmes en ligne et les services exposés. Des outils open source tels que **Nmap**, **Masscan** ou encore **Rustscan** permettent de scanner les plages d'adresses IP fournies dans le périmètre d'audit. L'objectif est de détecter les serveurs critiques, et en particulier les **contrôleurs de domaine Active Directory**, reconnaissables par la présence de services caractéristiques comme **LDAP** (389/TCP, 636/TCP), **Kerberos** (88/TCP, 88/UDP), **DNS** (53/TCP, 53/UDP), **Global Catalog** (3268/TCP, 3269/TCP) ou encore **ADWS** (9389/TCP).

Ces outils peuvent également aider à repérer des services inattendus ou mal protégés, qui constituent souvent des points d'entrée intéressants.

Parmi les nombreux outils de cartographie réseau, **Nmap** s'impose comme une référence incontournable. Sa popularité ne tient pas seulement à son ancienneté, mais à la richesse de ses fonctionnalités, sa portabilité (Linux, Windows, macOS) et sa communauté active. Il permet à la fois de découvrir des hôtes, d'identifier les services en écoute, de reconnaître les versions logicielles, voire d'exécuter des scripts d'audit automatisés.

Parmi ses nombreuses options, certaines méritent une attention particulière car elles couvrent la majorité des besoins en reconnaissance :

`-n` : désactive la résolution DNS afin d'accélérer le scan et éviter des requêtes parasites.

`-sn` : exécute uniquement une détection d'hôtes (ping scan), sans scanner les ports.

`-Pn` : considère l'hôte comme actif sans envoi de ping préalable, utile lorsque l'ICMP est filtré.

`-T5` : règle le niveau d'agressivité du scan (T3 est le comportement par défaut ; `T5` est beaucoup plus rapide mais plus détectable).

Pour le scan des ports :

`-sS` : effectue un SYN scan furtif, rapide et discret, très utilisé en audit.

`-p-` : scanne l'ensemble des 65 535 ports TCP, pour une couverture complète.

`--top-ports <N>` : se limite aux N ports les plus fréquemment utilisés (par défaut 1 000).

`--open` : affiche uniquement les ports ouverts, simplifiant la lecture des résultats.

Pour l'identification des services :

`-sV` : active la détection de versions logicielles.

`--version-intensity <0-9>` : ajuste la profondeur de cette détection (7 par défaut, 9 étant le plus agressif).

`-sC` : lance les scripts par défaut de Nmap (NSE), qui testent des configurations ou vulnérabilités courantes.

`--script <nom>` : permet d'exécuter des scripts spécifiques (par ex. liés à SMB, LDAP ou DNS).

Enfin, pour les rapports :

`-oG <fichier>` : exporte les résultats au format grepable, un format pratique pour être réutilisé par d'autres outils.

`-oX <fichier>` : exporte les résultats en XML, un format pratique pour être intégré dans un rapport.

Ces options, combinées judicieusement, permettent d'adapter **Nmap** à la majorité des contextes rencontrés en audit : de la découverte initiale rapide à l'analyse approfondie de services critiques.

Nmap accepte une grande variété de cibles : une adresse IP unique, plusieurs adresses séparées par des espaces, une plage d'IP (ex. 192.168.1.1-50), une CIDR (ex. 192.168.1.0/24) ou encore un fichier contenant une liste d'hôtes (-iL <fichier>).

– Découverte d'hôte actif :

```
nmap -sn <ip>
```

– Découverte automatisée et export des hôtes :

```
nmap -sn <subnet> -oG - | awk '/Up$/{print $2}' > hosts.txt
```

– Découverte de l'Active Directory :

```
nmap -sS -Pn -p 53,88,389,636,3268,3269,9389 --open -iL hosts.txt
```

– Identification rapide des services exposés :

```
nmap -sS -Pn -sV --top-ports 50 --open -iL hosts.txt
```

– Scan complet TCP :

```
nmap -sS -Pn -sC -sV -p- --open -iL hosts.txt
```

– Scan UDP ciblé :

```
nmap -sU -Pn -sV --top-ports 50 --open -iL hosts.txt
```

La sortie brute de Nmap, bien qu'extrêmement riche, peut parfois être difficile à lire lorsqu'elle affiche un grand nombre d'hôtes et de services. Pour améliorer la lisibilité, il est possible d'utiliser **grc** (*Generic Colouriser*), un outil simple qui ajoute de la couleur et rend l'interprétation des résultats bien plus agréable. Ce programme n'est pas limité à Nmap : il peut également être appliqué à de nombreux autres utilitaires en ligne de commande, rendant l'expérience d'audit plus confortable.

```
apt install grc
grc nmap -sn <ip>
```

2.1.2 Analyse DNS et recherche d'informations associées

Dans un environnement Active Directory, le DNS occupe une place centrale puisqu'il assure la résolution des noms de machines et des services liés au domaine. L'auditeur doit donc s'appuyer sur ce service pour en extraire des informations complémentaires. Après avoir identifié les contrôleurs de domaine au moyen de scans et d'analyses de services, il est essentiel de configurer correctement la résolution DNS sur la machine d'audit. Sur une distribution comme Kali Linux, cela revient généralement à ajouter l'adresse IP d'un contrôleur de domaine dans le fichier /etc/resolv.conf. Cette configuration permet de résoudre les noms internes du domaine et d'interagir de façon fiable avec les différents services Active Directory, condition indispensable pour la suite de l'évaluation.

Une fois cette étape réalisée, l'auditeur peut approfondir son analyse en vérifiant la configuration du service DNS. Dans certains cas, une erreur peut autoriser un **transfert de zone**, offrant alors une visibilité quasi complète sur les enregistrements disponibles et donc sur la structure interne du domaine. Bien que cette mauvaise configuration soit rare dans des environnements correctement sécurisés, elle reste un exemple parlant de ce qu'une reconnaissance attentive peut révéler. Le test s'effectue aisément à l'aide de l'outil **dig**, par exemple avec la commande :

```
grc dig axfr <domaine> @<serveur_dns>
```

Afin d'obtenir tous les enregistrements IPv4, on peut filtrer le résultat de la commande précédente de la manière suivante :

```
grc dig axfr <domaine> @<serveur_dns>| awk '/IN[[:space:]]+A/
{print $1, "->", $NF}'
```

2.1.3 Vulnérabilités : Quick Compromise

Une vulnérabilité représente une faiblesse ou une faille dans un système informatique, une application ou un réseau, généralement due à une erreur de conception, de configuration ou d'implémentation. De nombreuses organisations recensent et publient ces vulnérabilités : CERT, GitHub, CNVD, et bien d'autres encore. Pour faciliter la centralisation de ces informations provenant de multiples sources, l'outil **Vulnerability Lookup** permet de corréler rapidement les vulnérabilités, même lorsque celles-ci ne disposent pas du même identifiant. Sa mission est de simplifier la gestion du CVD (*Coordinated Vulnerability Disclosure*) et de rendre plus accessible l'analyse des vulnérabilités issues de dépôts hétérogènes. Une liste récente de vulnérabilités issues de différentes sources est disponible à l'adresse suivante :
https://vulnerability.circl.lu/recent#cvelistv5

Parmi toutes ces sources, les **CVE** (*Common Vulnerabilities and Exposures*) sont les identifiants les plus connus du grand public et constituent une référence mondiale pour répertorier les failles de sécurité. Il est possible sur certains sites d'effectuer une recherche par produit de manière plus conviviale, par exemple pour **Windows 10** :
https://www.cvedetails.com/vulnerability-list/vendor_id-26/product_id-32238/Microsoft-Windows-10.html

À ce stade de l'audit, il est essentiel de répertorier les vulnérabilités sans chercher à les exploiter immédiatement. Plusieurs outils permettent de rechercher des CVE et d'évaluer le niveau de risque associé aux systèmes et services identifiés. Parmi les solutions open source, on trouve **Nuclei**, adapté à la détection automatisée de vulnérabilités sur les applications web, ainsi que **GVM/OpenVAS**, qui propose des scans réseau et applicatifs complets. Des solutions commerciales comme **Nessus**, **QualysGuard**, **Acunetix** ou **Nexpose** offrent des bases de données régulièrement mises à jour et des capacités d'analyse approfondies.

▶Pour installer GVM, exécutez les commandes suivantes :

```
curl -f -O -L https://greenbone.github.io/docs/latest/_static/
docker-compose.yml
docker compose up -d
```

Ce projet illustre l'adoption croissante des technologies de conteneurisation pour simplifier le déploiement d'outils et d'applications en général, en évitant les problèmes de dépendances et en préservant ainsi l'intégrité du système hôte.

Il existe de nombreux outils permettant de tester et d'exploiter les vulnérabilités des différents systèmes et plateformes. Parmi eux, l'un des plus connus est le framework **Metasploit (MSF)**. Celui-ci intègre un vaste catalogue d'**exploits** (programmes destinés à tirer parti d'une vulnérabilité) et de **payloads**, dont le fameux **Meterpreter**, qui permet d'interagir de manière avancée avec un système compromis. Mais **Metasploit** ne se limite pas à l'exploitation ; il propose également de nombreux **modules auxiliaires** dédiés à la reconnaissance et à l'inventaire des services.

Par exemple, le module **auxiliary/scanner/smb/smb_version** permet de scanner les services SMB d'un hôte afin d'identifier la version et la signature du protocole utilisé.

Pour utiliser Metasploit et exécuter le module **auxiliary/scanner/smb/smb_version**, vous pouvez suivre ces étapes :

– Lancer Metasploit : ouvrez votre terminal et tapez **msfconsole** pour démarrer l'interface de ligne de commande de Metasploit.
– Rechercher le module : utilisez la commande `search smb_version` pour localiser le module spécifique que vous souhaitez utiliser.
– Consulter les options : une fois le module sélectionné avec **use auxiliary/scanner/smb/smb_version**, tapez **show options** pour afficher les paramètres configurables, comme **RHOSTS**, qui spécifie l'adresse IP de la cible.
– **Exécuter le module** : après avoir configuré les options nécessaires, utilisez la commande `run` pour lancer le scan.