

SUPNUM

RSS

TD 1

Énumération, Fuzzing et OSINT

Introduction à la sécurité informatique

Mohamed Said M'Bareck
2025-2026

Objectifs du TD

Ce TD vise à vous initier à la phase de reconnaissance, étape fondamentale de tout audit de sécurité. Vous apprendrez à identifier des services, à énumérer des informations spécifiques et à découvrir des ressources cachées sur un serveur web.

Partie 1 : Rappels Théoriques

Qu'est-ce que l'énumération ?

L'énumération est le processus qui consiste à **sonder activement une cible** pour en extraire des informations utiles. C'est comme faire le tour d'un bâtiment pour en repérer les portes, les fenêtres, le type de serrures, et le nombre d'étages avant d'essayer d'y entrer. Les informations collectées incluent :

- Noms d'utilisateurs et de machines
- Adresses IP et partages réseaux
- Services réseaux et leurs versions

OSINT (Open-Source Intelligence)

L'OSINT est l'art de **collecter des informations à partir de sources publiques**. Avant même d'envoyer un seul paquet à la cible, on peut déjà trouver des indices précieux via les moteurs de recherche (Google Dorking), les réseaux sociaux, les dépôts de code (GitHub) ou des sites spécialisés (Shodan).

Fuzzing

Le fuzzing consiste à envoyer des **données inattendues, malformées ou aléatoires** à une application pour observer son comportement. Dans le contexte web, on l'utilise surtout pour "deviner" l'existence de fichiers ou de répertoires non listés, en testant une liste de noms courants (une *wordlist*).

Partie 2 : Exercices Pratiques

Exercice 1 : Enquête OSINT - Le Cas Random Corp

Objectif :

Utiliser des techniques d'OSINT pour collecter des informations sur un employé et son entreprise afin de découvrir un point d'entrée potentiel dans leur système d'information.

Scénario :

"Someone Somewhere" est un ingénieur DevOps chez "Random Corp". Le site web de l'entreprise est `randomcorp.lab`. Votre mission est de mener une enquête en sources ouvertes pour évaluer les risques liés à l'empreinte numérique de cet employé.

Instructions :

1. Énumération des sous-domaines

La première étape est de cartographier la surface d'attaque externe de `randomcorp.lab`.

- Utilisez des outils comme `dnsenum`, `subfinder` ou des services en ligne (ex : DNSdumpster) pour découvrir les sous-domaines.

```
dnsenum --threads 5 randomcorp.lab
```

- Notez les sous-domaines qui semblent intéressants, comme ceux liés aux RH, au développement ou à l'administration.

2. Recherche de l'empreinte numérique de l'employé

Trouver les comptes publics d'un employé technique peut révéler des informations précieuses.

- Utilisez des moteurs de recherche pour trouver les comptes GitHub, X (Twitter) et Reddit de "Som1 Somewhere". Essayez des requêtes comme "Som1 Somewhere" devops github ou "Som1 Somewhere" dans LinkedIn.
- L'objectif est de trouver un **nom d'utilisateur** unique qu'il pourrait réutiliser sur d'autres plateformes ou de trouver des références à d'autres comptes.

3. Analyse de l'organisation sur GitHub

- Cherchez le compte GitHub de l'organisation "Some Random Corp". Explorez ses dépôts publics pour comprendre les technologies utilisées.

4. Enquête sur Docker Hub

Les DevOps publient souvent des images de conteneurs, qui peuvent parfois contenir des secrets.

- En utilisant le nom d'utilisateur de "Som1 Somewhere", cherchez son profil sur **Docker Hub**.
- **Question** : Quelles images a-t-il publiées ? Y en a-t-il une qui semble liée à un projet interne ?

5. Analyse d'une image Docker

Agissons comme un attaquant et analysons ce qui a été publié.

- Téléchargez et exécutez localement l'image Docker suspecte.

```
docker pull somisomewhere/accounts
docker run -p 4444:8000 somisomewhere/accounts
```

- Accédez à l'application sur `http://localhost:4444`.
- **Question** : N'avez-vous jamais vu cette interface auparavant ? Explorez l'application. Y a-t-il des informations d'identification par défaut ou des indices laissés dans l'image ?

6. Tentative d'accès

Il est temps de connecter toutes nos découvertes.

- Retournez à votre liste de sous-domaines. Avez-vous identifié un portail pertinent (ex : `accounts.randomcorp.lab`) ?
- Essayez d'utiliser les informations d'identification ou les indices trouvés dans le conteneur Docker pour vous connecter.

7. Synthèse de l'attaque

Question finale : Résumez en quelques phrases comment une simple publication sur Docker Hub par un employé a conduit au piratage du système RH de l'entreprise. Quel a été le maillon faible ?

Exercice 2 : Découverte et Scan Avancé avec Nmap

Objectif :

Apprendre à mener une reconnaissance réseau systématique en plusieurs étapes : découverte d'hôtes, scan de ports, identification de services, détection d'OS et utilisation de scripts pour trouver des vulnérabilités.

Instructions :

Pour cet exercice, nous allons simuler une approche méthodique pour scanner un sous-réseau. L'adresse IP de votre machine d'attaque (Kali) sera notée `<VOTRE_IP>`.

1. Découverte des Hôtes Actifs

La première étape consiste à identifier quelles machines sont "vivantes" sur le réseau, sans les alerter avec un scan de ports agressif.

- Exécutez un scan "Ping" (`-sn`) qui se contente de découvrir les hôtes. Nous sauvegardons la sortie dans un format "grepable" (`-oG`) pour pouvoir l'utiliser facilement.

```
sudo nmap -sn 192.168.9.0/24 -oG targets.txt
```

- Analysez le fichier `targets.txt` et extrayez uniquement les adresses IP des hôtes actifs pour les mettre dans un nouveau fichier `ips.txt`. Vous pouvez le faire manuellement ou avec une commande :

```
# Commande pour extraire automatiquement les IPs
grep "Up" targets.txt | cut -d " " -f 2 > ips.txt
```

Question : Pourquoi est-il plus efficace de commencer par un scan `-sn` avant de lancer un scan de ports complet sur un grand réseau ?

2. Scan des Ports sur les Cibles Identifiées

Maintenant que nous avons notre liste de cibles, nous allons scanner leurs ports ouverts.

- Exécutez un scan SYN (`-sS`, rapide et discret) en utilisant votre fichier de cibles (`-iL`) et en excluant l'adresse IP du gateway.

```
sudo nmap -sS -iL ips.txt --exclude 192.168.9.1 -oG ports-scan
```

Question : Quels sont les avantages d'un scan SYN (`-sS`) par rapport à un scan par connexion TCP (`-sT`) dans un contexte d'audit ?

3. Identification des Services et Versions

Connaître les ports ouverts est bien, mais connaître les logiciels et leurs versions exactes qui tournent dessus est crucial pour trouver des failles.

- Lancez un scan de détection de version (`-sV`) sur vos cibles.

```
sudo nmap -sV -iL ips.txt
```

Question : En analysant les versions des services (ex : Apache, vsftpd, OpenSSH), avez-vous identifié des logiciels qui vous semblent anormalement anciens et qui mériteraient une recherche de vulnérabilités publiques (CVE) ?

4. Détection du Système d'Exploitation (OS)

Identifier l'OS peut nous donner des indices sur les vulnérabilités potentielles au niveau du système.

- Lancez un scan de détection d'OS (`-O`) sur une cible spécifique (ex : 192.168.9.5).

```
sudo nmap -O 192.168.9.5
```

Question : Nmap est-il certain à 100% du système d'exploitation ? Sur quels indices techniques se base-t-il pour faire cette déduction ?

5. Utilisation des Scripts Nmap (NSE)

Le Nmap Scripting Engine (NSE) est une des fonctionnalités les plus puissantes de Nmap. Il permet d'automatiser la recherche de failles connues.

- Lancez un scan qui exécute la catégorie de scripts par défaut (`-script=default`) pour une reconnaissance plus approfondie.

```
nmap -sV --script=default 192.168.9.5
```

Question : Le rapport des scripts a-t-il révélé des informations critiques qu'un simple scan de ports/versions n'aurait pas montrées (ex : une connexion anonyme autorisée, des fichiers de configuration exposés) ?

6. Comparaison des Scans (DNS)

Voyons l'impact de la résolution de nom de domaine (DNS) sur un scan.

- Exécutez les deux commandes suivantes et comparez la vitesse et les informations retournées.

```
# Scan without DNS resolution (-n)
nmap -n supnum.mr

# Scan with DNS resolution (the default behavior)
nmap supnum.mr
```

Question : Dans quel scénario d'audit est-il indispensable d'utiliser l'option `-n` ?

7. Sauvegarde des Résultats pour Analyse

Un bon auditeur garde toujours des traces claires de ses scans. L'option `-oA` est parfaite pour cela.

- Lancez un scan final sur une cible et sauvegardez les résultats dans tous les formats principaux.

```
nmap -oA scan_results 192.168.9.5
```

- Inspectez les trois fichiers générés : `scan_results.nmap`, `scan_results.xml` et `scan_results.gnmap`.

Question : Si vous deviez présenter un rapport à un client, quel fichier utiliseriez-vous comme base ? Si vous deviez intégrer les résultats dans un autre outil de sécurité, lequel choisiriez-vous ?

Exercice 3 : Fuzzing Web Avancé avec FFUF

Objectif :

Découvrir des répertoires et des fichiers cachés sur un serveur web en utilisant des techniques de fuzzing, de la plus simple à la plus avancée.

Instructions :

1. Préparation de la Wordlist

Une bonne wordlist est la clé d'un fuzzing réussi. Le paquet `seclists` contient une collection très complète de listes de mots pour toutes sortes de tests de sécurité.

- Installez `seclists` (recommandé) :

```
sudo apt update && sudo apt install seclists
# Or
sudo snap install seclists
```

- Si vous ne pouvez pas l'installer, une liste plus simple est souvent déjà disponible : `/usr/share/dirb/wordlists/common.txt`. Pour la suite, nous utiliserons une liste de `seclists`.

2. Fuzzing de Répertoires (scan de base)

Commençons par chercher des répertoires courants. FUZZ est le mot-clé que `ffuf` remplace par chaque ligne de la wordlist.

- Lancez le scan :

```
# Basic directory scan
ffuf -u http://app3.rss.lab/FUZZ \
    -w /usr/share/seclists/Discovery/Web-Content/common.txt
```

- Pour explorer récursivement :

```
# Recursive scan (explore discovered directories)
ffuf -recursion -u http://app3.rss.lab/FUZZ \
    -w /usr/share/seclists/Discovery/Web-Content/common.txt
```

Question : Avez-vous trouvé un répertoire sensible (ex : `/admin`, `/config`, `/backups`) ? Quelle information un fichier `robots.txt` peut-il vous donner ?

3. Fuzzing de Fichiers avec Extensions

Un attaquant recherche aussi des fichiers de configuration, des scripts ou des sauvegardes oubliées.

- Cherchez différents types de fichiers :

```
# Search for PHP and text files
ffuf -u http://app3.rss.lab/FUZZ \
    -w /usr/share/seclists/Discovery/Web-Content/common.txt \
    -e .php,.txt

# Search for .bak files under /backups
ffuf -u http://app3.rss.lab/backups/FUZZ \
    -w /usr/share/seclists/Discovery/Web-Content/common.txt \
    -e .bak,.php.bak
```

Question : Pourquoi la découverte d'un fichier comme `config.php.bak` serait-elle une vulnérabilité critique ?

4. Optimisation et Options Avancées

Pour des audits réels, il faut affiner ses commandes pour être plus efficace.

- Augmenter la vitesse :

```
# Use 50 threads (do not overload the target)
ffuf -t 50 -u http://app3.rss.lab/FUZZ \
    -w /usr/share/seclists/Discovery/Web-Content/common.txt
```

— Filtrer les résultats :

```
# Show only 200/301/302 responses
ffuf -u http://app3.rss.lab/FUZZ \
     -w /usr/share/seclists/Discovery/Web-Content/common.txt \
     -mc 200,301,302

# Variant: show only 200 (OK) and 500 (server error)
ffuf -u http://app3.rss.lab/FUZZ \
     -w /usr/share/seclists/Discovery/Web-Content/common.txt \
     -mc 200,500
```

— Exporter les résultats :

```
# Save results as JSON
ffuf -u http://app3.rss.lab/FUZZ \
     -w /usr/share/seclists/Discovery/Web-Content/common.txt \
     -o results.json -of json
```

— Combiner plusieurs options :

```
# Recursion + multiple extensions + filtering + JSON output
ffuf -recursion -t 30 \
     -u http://app3.rss.lab/FUZZ \
     -w /usr/share/seclists/Discovery/Web-Content/common.txt \
     -e .php,.txt,.bak -mc 200,301 \
     -o full_scan.json -of json
```

Question finale : Après avoir découvert une page de connexion `/admin/login.php`, quelle serait la prochaine étape logique de votre audit ?

Synthèse et Conclusion

Aujourd'hui, vous avez suivi la méthodologie d'un auditeur en phase de reconnaissance :

OSINT (passif) → Scan réseau (nmap) → Énumération de services (NSE) → Fuzzing (ffuf)

Ce processus est **itératif**. La découverte d'un nouveau répertoire via le fuzzing peut mener à un nouveau scan sur ce répertoire. L'énumération d'un service peut révéler des noms d'utilisateurs à tester ailleurs.

La clé du succès est la **rigueur** et la **prise de notes** !