

2024  
2025

# Travaux Pratiques 3

## Test d'intrusion d'un serveur Web

RSX112 - SÉCURITÉ DES RÉSEAUX  
STÉPHANE LARCHER



# Test d'intrusion d'un serveur Web

# Principe du test d'intrusion

Un test d'intrusion permet de :

- i. Vérifier si un système est vulnérable.
- ii. Comprendre comment ces vulnérabilités peuvent être découvertes et exploitées (et le cas échéant, en combien de temps).
- iii. Faire des recommandations pour mieux protéger le système.

# Préparation de l'environnement de test

“Que l'on me donne six heures pour couper un arbre, j'en passerai quatre à préparer ma hache.” Abraham Lincoln...

## Installation de l'outil KALI

- i. Téléchargez la machine virtuelle [ici](#) (kali/kali),
- ii. Changez le clavier :
  - a. `$ setxkbmap –layout fr`
  - b. `$ azerty`
  - c. `$ sudo dpkg-reconfigure keyboard-configuration`

Sinon, prenez le matériel : « Generic 104 key » et le format « français : french – french AZERTY »

# Recherche des informations sur la cible avec Google

On va utiliser google pour trouver :

- i. Des vulnérabilités,
- ii. Des données sensibles,
- iii. Avec le robot « Googlebot »
  - a. Filtrage précis des résultats
  - b. Ne permet pas d'avoir accès à des informations interdites (ce n'est pas magique...)
- iv. Grace à Google
  - a. C'est ce qu'on appelle un **Google Dork** (ou du "Google hacking").
  - b. Cette activité existe depuis le début des années 2000 et a été popularisée par Johnny Long lors de sa conférence (diapositives en anglais) à la Black Hat 2005 (une des plus grosses conférences de cybersécurité au monde). Il a également écrit des livres sur le sujet.

## La recherche

Recherche tous les documents PDF indexés sur le site "root-me.org", contenant les mots :

- i. "pass"
- ii. "password"
- iii. "mot de passe"

Dans la barre de recherche : "root-me.org ext:"extension des fichiers recherchés" "mot recherché", vous pouvez exclure aussi avec l'option "-filetype : "extension"

# Le Challenge

Petite précision : « Root Me » est prévu expressément pour les tests donc sans risque légale...

## Attention !

Le site [Exploit-DB](#) (en anglais) recense une liste de Google Dorks dans ce qu'ils appellent la [Google Hacking DataBase \(GHDB\)](#).

**C'est à utiliser avec précaution et sur un périmètre pour lequel vous êtes autorisé.** Sinon, vous risquez de vous retrouver dans [la même situation que ce blogueur](#), qui a écopé d'une amende de 3 000 € pour avoir téléchargé des documents indexés par Google !

Donc le but est de trouver la requête pour faire une recherche sur un article écrit par une personne se nommant « Lebrun », présent sur le site de « Root Me », peut-être est-ce un PDF.

## Autres recherches

Pour bien mener votre reconnaissance passive, c'est-à-dire sans toucher directement l'application de healthtech pour le moment, il vous faut chercher le maximum d'informations :

- i. Qui sont les employés de l'entreprise éditrice de example.com, et quelles sont leurs adresses électroniques ?
- ii. Est-ce que ces adresses électroniques auraient été utilisées dans une base de données qui a fuité ?
- iii. Est-ce que l'entreprise dispose d'un dépôt de code sur GitHub ou GitLab, par exemple ?
- iv. Est-ce que des messages relatifs à l'application cible ont été postés sur des forums spécialisés ?

Sur la machine virtuelle Kali, lancer la commande :

```
$ theHarvester -d root-me.org -b all
```

Qu'enseigne cette commande, quel est le résultat ?

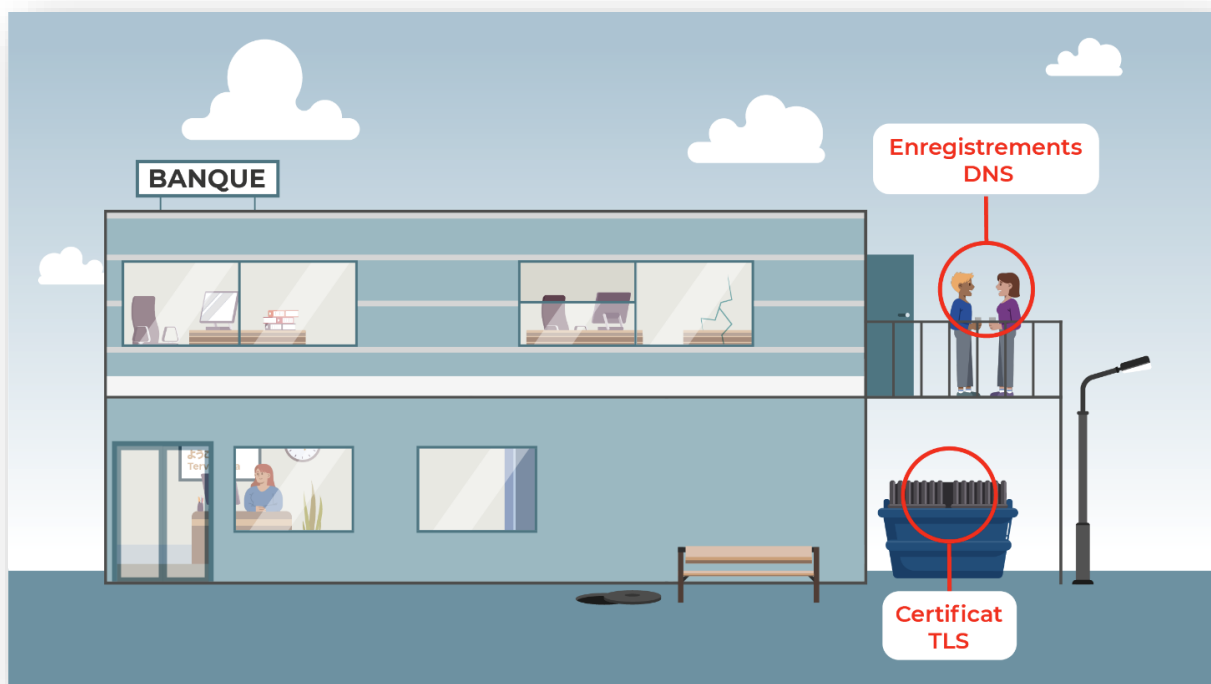
## La reconnaissance active

Il s'agit ici de trouver des cibles connexes comme :

- i. Énumération de noms de domaine et de sous-domaine, à partir d'une liste que vous aurez définie ;
- ii. Le *scraping* de la cible principale, pour trouver des références à des sous-domaines dans le contenu de la cible ; (scraping : parcourir les pages d'un site de manière automatique)
- iii. Lecture des certificats TLS (Transport Layer Security), qui sont parfois utilisés pour plusieurs sites ;
- iv. Registres whois ;
- v. Historique DNS...

On utilise ici l'outil « amass » pour l'énumération des domaines :

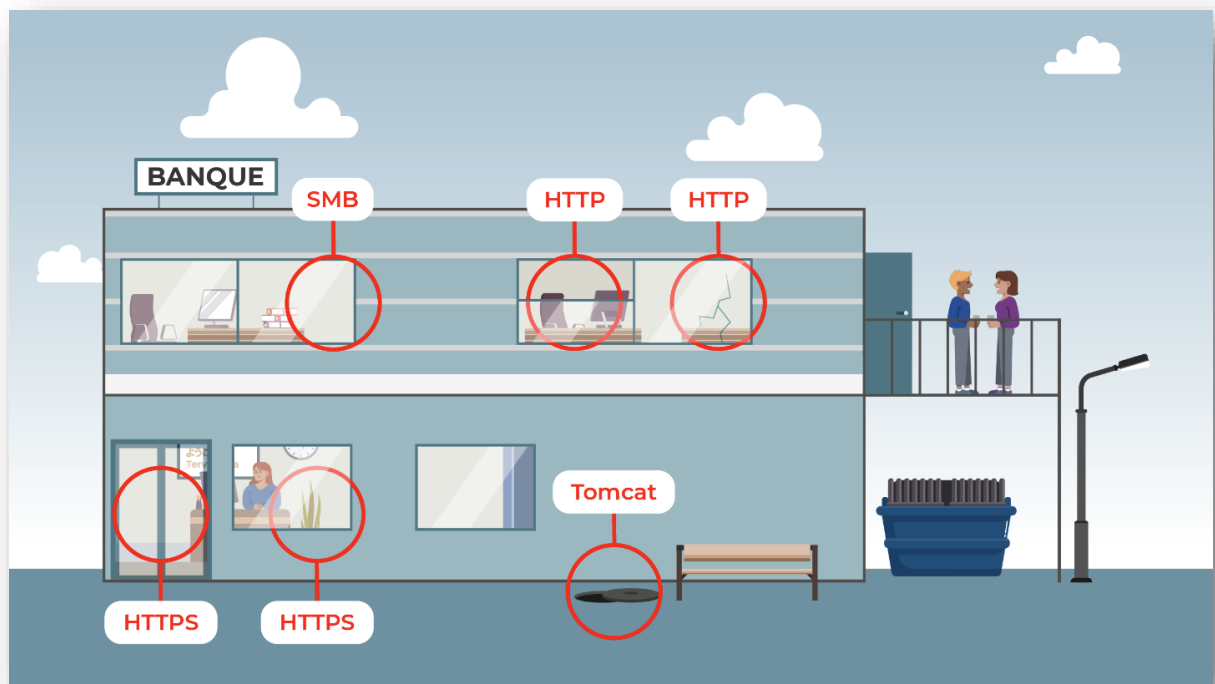
```
$ amass enum -d root-me.org -active
```



Pouvez-vous donc cartographier le site « root-me.org » ?



## Identification des points d'entrée



## Le scan des ports

Utilisation de l'outil « nmap » :

```
$nmap root-me.org -p XXX-YYY
```

En décrire le résultat

Scanner avec cet outil les ports de 0 à 65535 (toute la plage donc)

## Pratique

Pour participer à ce challenge, il faut posséder un compte sur Root Me, c'est gratuit.

Connectez-vous à [Root Me](#).

- i. Rendez-vous sur la page de connexion root-me [ici](#) pour démarrer ou rejoindre l'environnement.
- ii. Une fois sur la page, attendez qu'un cartouche vert apparaisse :

L'environnement virtuel à attaquer est disponible à cette adresse : **ctfXX.root-me.org**

- iii. L'adresse à scanner y sera indiquée.

- iv. Vos objectifs sont les suivants :
  - a. Scanner la machine pour identifier son exposition et trouver le nombre de ports qui sont ouverts ;
  - b. Trouver quel est le service en écoute sur le port 2121.
  - c. En trouver ses vulnérabilités
- v. Pour cela :
  - a. Renseigner dans la fenêtre de recherche « Google » : WordPress 3.5.1 vulnérability.(un exemple)

## Vérification du chiffrement des échanges

Les points d'attention :

- i. La taille de clé du certificat,
- ii. L'algorithme de hachage
- iii. Avec les outils suivants :
  - a. Certains en ligne comme [SSLlabs](#) de la société Qualys ;
  - b. Et d'autres, locaux et en CLI (ligne de commande), comme [SSLscan](#) et [testssl.sh](#).

N.B : Utilisation de testssl.sh :

- i. `$ git clone https://github.com/drwetter/testssl.sh`
- ii. `$ cd testssl.sh`
- iii. `$ chmod +x testssl.sh`
- iv. `$ ./testssl.sh ctfXXX.root-me.org`

## Pratique

- i. Comme pour l'exercice précédent, connectez-vous à [Root Me](#).
- ii. Une fois sur la page, attendez qu'un cartouche vert apparaisse

L'environnement virtuel à attaquer est disponible à cette adresse : [ctfXX.root-me.org](#)

- iii. Vos **objectifs** sont de vérifier les points suivants de la configuration SSL du service :
  - a. La qualité du certificat utilisé :
    - i. Est-ce que le CN ou le SAN du certificat correspond au nom de domaine du site ?
    - ii. Quelle est la taille de la clé ?
    - iii. Quel est l'algorithme de signature ?
  - b. Les protocoles proposés par le serveur.
  - c. Les suites de chiffrement supportées.
  - d. La présence de vulnérabilités connues dans la librairie utilisée. Utilisez [testssl.sh](#).
  - e. Pouvez-vous en faire un bilan ?

## Contrôle des requêtes avec un proxy d'interception web

Quand vous consultez une application Web, Pour visualiser les données qui sont échangées entre vous et celle-ci, nous allons utiliser « Burp Suite Community »

On va configurer notre PC pour que les flux web soient envoyés à ce proxy, et ce proxy va ensuite les relayer aux serveurs web. La différence avec le proxy d'entreprise, c'est que nous avons la main sur le proxy pour intercepter, modifier et rejouer les requêtes.

### Installation

```
$ sudo apt update (comme toujours...)
```

```
$ sudo apt install burpsuite -y
```

- i. Lancer Burp.
- ii. Configurer le proxy, pour définir comme proxy votre navigateur
- iii. Configurer le magasin de certificats.