I - Nmap Scripting Engine (NSE)

Qu'est-ce que le NSE ?

Le **Nmap Scripting Engine (NSE)** est un moteur intégré à Nmap qui permet d'exécuter des scripts automatisés durant les scans.
Ces scripts, écrits en Lua, facilitent des tâches avancées comme :

- La collecte d'informations supplémentaires
- La détection de vulnérabilités
- La vérification de configurations spécifiques
- L'exécution de tests de sécurité

Le NSE enrichit ainsi considérablement les capacités d'analyse de Nmap, au-delà du simple scan de ports.

Utilisation basique

- L'option `-sC` active l'exécution des scripts NSE « par défaut » (catégorie **default**), qui sont sûrs et utiles pour la reconnaissance initiale.
- Exemple de commande avec NSE par défaut :

```
sudo nmap -sC -sV -Pn -oN nmap_scripts.txt 10.0.10.2
```

Ici, on combine la détection de versions (`-sV`) avec les scripts par défaut (`-sC`).

Catégories principales de scripts NSE

Nmap classe ses scripts NSE en différentes catégories :

Catégorie	Description	Utilisation typique
default	Scripts sûrs pour reconnaissance de base	Découverte et info rapide
discovery	Exploration approfondie du réseau	Identification de services
intrusive	Tests pouvant impacter la cible	Tests d'exploitation (prudence)
vuln	Recherche de vulnérabilités connues	Audit de sécurité
auth	Scripts liés à l'authentification	Tests d'identification
exploit	Exploitation active de failles	Attaques ciblées

Exemples de scripts NSE par défaut (`-sC`)

- `http-title` : récupère le titre d'une page web (aide à identifier un service web).

- `ssh-hostkey` : récupère la clé hôte SSH (empreinte et version).
- `smtp-commands` : liste les commandes SMTP acceptées par un serveur mail.
- `dns-recursion` : teste si un serveur DNS permet la récursion (risque de sécurité).
- `ftp-anon` : teste l'accès anonyme sur un serveur FTP.

Personnalisation avancée

- On peut exécuter des scripts spécifiques ou par catégorie avec `--script` :

```
sudo nmap --script vuln 10.0.10.2
```

Lance uniquement les scripts de la catégorie vulnérabilité.

- Pour connaître tous les scripts disponibles :

```
ls /usr/share/nmap/scripts/
```

- Pour obtenir la documentation détaillée d'un script :

```
nmap --script-help <nom_du_script>
```

Pourquoi utiliser NSE ?

- **Automatisation** : facilite et accélère l'audit en intégrant plusieurs tests en une seule passe.
- **Richesse fonctionnelle** : couvre une large gamme de protocoles, vulnérabilités, et tâches.
- **Personnalisation** : possibilité d'écrire ses propres scripts Lua.
- **Complémentarité** : enrichit l'analyse des versions et des services avec des informations exploitables.

> Le NSE est un outil indispensable pour les audits de sécurité et la reconnaissance avancée.

II - Partie Avancée

1. Comment écrire un script NSE simple (Lua)

Un script NSE est un fichier .nse écrit en Lua avec cette structure basique :

```
description = [[
Test script example qui retourne l'IP de la cible.
]]

categories = {"discovery"}
```

```
action = function(host)
  return "Host is " .. host.ip
end
```

- description : texte explicatif du script
- categories : catégories auxquelles le script appartient
- action(host) : fonction principale exécutée pour chaque hôte, elle retourne un résultat (texte, table, etc.)

Pour aller plus loin, tu peux utiliser des fonctions Nmap comme nmap.new_socket(), stdnse.sleep(), etc., pour interagir avec le réseau.

---

2. Gestion des risques et précautions lors de l'utilisation des scripts NSE

- Scripts intrusifs (catégorie intrusive) peuvent provoquer des interruptions de service, des alertes ou des détections.
- Toujours demander l'autorisation avant de lancer des scans approfondis, surtout avec NSE.
- Utiliser les options --max-rate ou --scan-delay pour limiter la vitesse et réduire l'impact sur la cible.
- Préférer d'abord les scripts « default » ou « discovery » avant d'utiliser des scripts vulnérabilité ou exploit.
- Analyser les résultats avant d'enchaîner avec des scans plus lourds.

---

3. Intégration du NSE dans un workflow automatisé (bash/python)

Exemple simple de script bash qui enchaîne un scan SYN et un scan vulnérabilité avec NSE :

```
#!/bin/bash

TARGET="10.0.10.2"

echo "Scan SYN TCP"
sudo nmap -sS -Pn -oN syn_scan.txt $TARGET

echo "Scan vulnérabilités NSE"
sudo nmap --script vuln -Pn -oN vuln_scan.txt $TARGET
```

En Python, tu peux lancer Nmap via un sous-processus (subprocess) ou utiliser une bibliothèque comme python-nmap :

```
```

```
import nmap

nm = nmap.PortScanner()
nm.scan('10.0.10.2', arguments='-sS -Pn')
print(nm.csv())

nm.scan('10.0.10.2', arguments='--script vuln -Pn')
print(nm.csv())
```

Cela permet d’intégrer Nmap dans des outils plus larges (analyse, reporting, alerting).

---

4. Ressources et documentation pour aller plus loin

- Documentation officielle Nmap NSE : https://nmap.org/book/nse.html
- Répertoire des scripts : /usr/share/nmap/scripts/
- Liste complète des scripts avec descriptions : https://nmap.org/nsedoc/
- Tutoriels et exemples sur GitHub et sites de pentesting (ex : https://github.com/nmap/nmap/tree/master/scripts)
- Forums et communautés (StackExchange, Reddit r/netsec, etc.) pour échanges avancés
- Documentation Lua : https://www.lua.org/manual/5.3/