TP1 : NMAP

Rushed Kanawati

13 mars 2017

Ce TP est à réaliser en binôme. Chaque binôme dispose de deux machines Linux. Un seul rapport est à rendre à la fin de la séance

Résumé

L'objectif de ce TP est de se familiariser avec l'outil `nmap` : un outil de référence pour l'audit de sécurité des réseaux. Les outils à utiliser sont : `nmap, zenmap` et `wireshark`

NMAP

`nmap` est un logiciel de balayage des réseaux. L'outil est disponible, sous licence de logiciel libre sur le site `http://insecure.org`. Cet outil est utilisé par les *pirates* pour préparer des attaques (attaques par balayage), et par les administrateurs systèmes pour tester la vulnérabilité de leurs systèmes (ex. études d'audit système).

Attention : tout acte de balayage de réseau non-autorisé est assimilé à une attaque et donc répréhensible selon la loi française. Nous nous limitons à faire des balayages des machines dans la salle du TP, et la machine distante `scanme.nmap.org` mise en ligne spécifiquement par les développeurs de l'outil à des fins pédagogiques.

Le principe de `map` est de solliciter des machines à balayer des réponses montrant l'état des différents ports. Différentes techniques de balayages sont possibles et s'appuient sur l'utilisation des protocoles de base : `tcp, udp, ip` et `icmp`. Par défaut, `nmap` balaye les ports systèmes (numéro de port < 1024). Selon `nmap` un port d'une machine peut être dans une des états suivants :

- **open** (ouvert) : port associé à un service actif.
- **closed** (fermé) : port associé à un service inactif.
- **filtered** (filtré) : Port inaccessible à cause d'un pare-feu par exemple.
- **unfiltered** (non filtré) : port accessible mais `nmap` n'arrive par à déterminer s'il est ouvert ou fermé.

Le balayage d'une machine destination se fait en quatre étapes (qui peuvent être modifiées en utilisant des options appropriées) :

1. Si l'adresse de la machine cible est donnée sous forme symbolique, une résolution `DNS` est déclenchée (à moins que l'adresse IP de la machine est donnée dans le fichier local *hosts*)
2. `nmap` envoie un paquet ICMP et attends le retour (opération ping). Cette phase peut être évitée en utilisant l'option `-P0`.
3. Si la destination est spécifiée sous forme d'adresse IP, une phase de résolution inverse `DNS` est déclenchée. Cette phase peut être évitée en utilisant l'option `-n`.
4. Le balayage spécifié est exécuté.

La syntaxe générale d'une commande `nmap` est la suivante :

nmap [types de scans] [options] *cibles*

Spécification de cibles

Les cibles peuvent être désignées par adresses symboliques ou adresses IP. L'adressage `CIDR` peut être utilisée afin de désigner des sous-réseaux. Il est aussi possible d'utiliser des intervalles pour designer des pans des réseaux. Par exemple : `192.168.34.1-17` désigne l'intervalle d'adresses [192.168.34.1, 192.168.34, 17]. Des virgules peuvent être employés aussi pour désigner un ensemble de valeurs non-ordonné. Exemple : `192.168.34.1,2` désigne les deux machines 1 et 2 sur le réseau `192.168.34.0/24`.

L'option `-p` permet de spécifier aussi le numéro de ports à utiliser. Des intervalles et des virgules peuvent aussi être utilisés pour désigner des ensembles de numéro de ports. Des exemples sont :

- `nmap -p80,443 localhost`
- `nmap -p1-1023 192.168.34,1-4`

Différents types de balayages sont possibles. Le tableau 1 donne les principaux types que nous allons étudiés dans ce TP. Par défaut le type de balayage utilisé est le balayage `TCP SYN`.

Prise en main de `nmap`

1. Installer si nécessaire, les logiciels `nmap`, `zenmap` (un client graphique pour nmap) et `wireshark`.

2. Comparer et justifier les différences des résultats entre les deux commandes suivantes : `nmap @PC1` et `netstat -a` exécutée sur PC1.

3. Exécuter chacune des commandes suivantes et capturer et justifier le trafic généré pour chaque exécution :
   (a) `nmap -p80 scanme.nmap.org`

Table 1 – Principaux types de balayage

Type de balayage	Syntaxe	mode root exigé
TCP SYN	`-sS`	OUI
TCP connect	`-sT`	NON
FIN	`-sF`	OUI
XMAS Tree	`-sX`	OUI
NULL	`-sN`	OUI
PING	`-sP`	NON
Détection de version	`-sV`	NON
UDP	`-sU`	OUI
IP	`-sO`	OUI
Acquittement	`-sA`	OUI
Idle	`-sI`	OUI

(b) `nmap -p113 scanme.nmap.org`

(c) `nmap -p113 -P0 scanme.nmap.org`

4 Recommandiez vous l'emploi de l'option `-P0` ?

5 Comment faire pour éviter les résolutions DNS ?

6 Sur PC1, donner une commande qui permet de tester l'état de ports web sur la machine PC2 en générant le minimum de paquets

7 Donner une commande qui permet de retourner les adresses des machines actives sur votre réseau local

Opérations de balayage

1 Combien de paquets sont générés suit à l'exécution sur PC1 de la commande `nmap -P0 -r -sS @PC2`. Est ce que la balayage des ports se fait dans l'ordre de leurs numéros ?

2 Donner la configuration des paquets générés par les balayages suivants : `-sF, -sX`. Expliquer le fonctionnement de ces attaques. Quel est l'avantage de ce type de balayage ? (indice : tester sur un port ouvert et un autre fermé)

3 Donner une commande qui permet de donner la liste des machines actives sur le réseau locale

4 Peut-on combiner différents type de balayages ?

5 Quel est l'effet de la commande `nmap -sV -p80 scanme.nmap.org`. Expliquer le fonctionnement après observation du trafic généré.

6 Consulter la page de manuel de la commande `nmap` et donner la finalité du balayage de type `-sO`. Tester ce mode de balayage sur le routeur auquel la machine ext connectée. Donner les résultats obtenues.

7 Quel type de paquets est généré par un balayage de type `-sA`. Peut on détecter les ports ouverts avec ce type de balayage ?

8 Comparer les résultats obtenus de l'exécution de ces deux commandes et tenter de justifier une éventuelle différence dans les résultats :

- `sudo nmap -sW -p80 -P0 scanme.nmap.org`
- `sudo nmap -sS -p80 -P0 scanme.nmap.org`

9 Tester et expliquer le fonctionnement de la commande : `nmap -sI @PC2 scanme.nmap.org`

10 Que fait la commande `nmap -O PC2`. Tester aussi avec `scanme.nmap.org` ?

11 L'option `-oX filename` permet de sauvegarder un rapport de balayage de l'opération en format `XML` dans le fichier spécifié. Tester cette option et donner la structure du fichier XML généré

zenamp

`zenmap` est un client graphique qui facilite l'utilisation de l'outil `nmap`. La figure suivante illustre l'interface principale de l'outil.

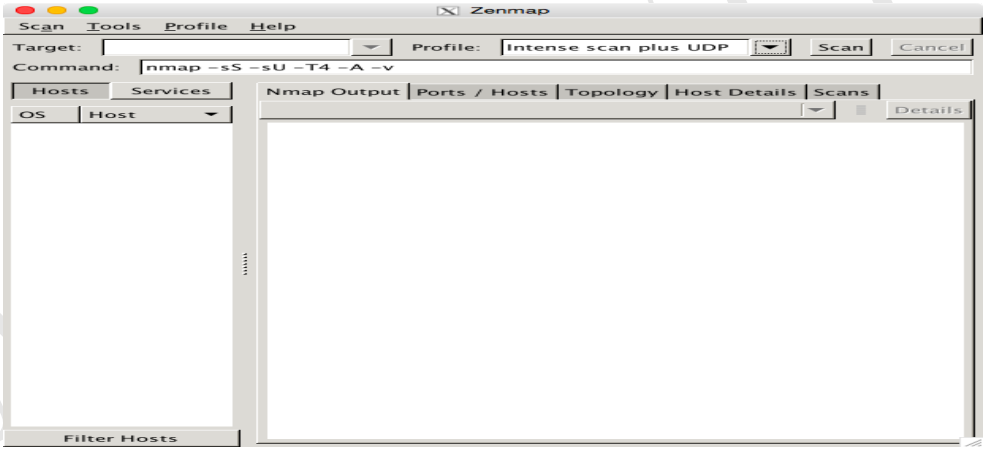

FIGURE 1 – Interface graphique de `zenmap`

`zenmap` offre notamment le concept de *profil* qui permet l'automatisation de génération d'options de la commande `nmap`. Executer le client `zenmap` et dresser les caractéristiques des profils de base fournis (utiliser la page de manuel pour expliquer les options générées).