BENCH-4-SCAN

RÉALISÉ PAR ACHRAF BENCHEHLA

SCANING

La phase de scanning est essentielle dans les tests d'intrusion, car elle consiste à explorer activement un système ou un réseau à la recherche de vulnérabilités. Cela permet de comprendre les risques, de prendre des mesures préventives et de renforcer la sécurité. Le scanning prévient les potentielles failles de sécurité avant qu'elles ne soient exploitées par des cybercriminels.

BENCH-4-SCAN

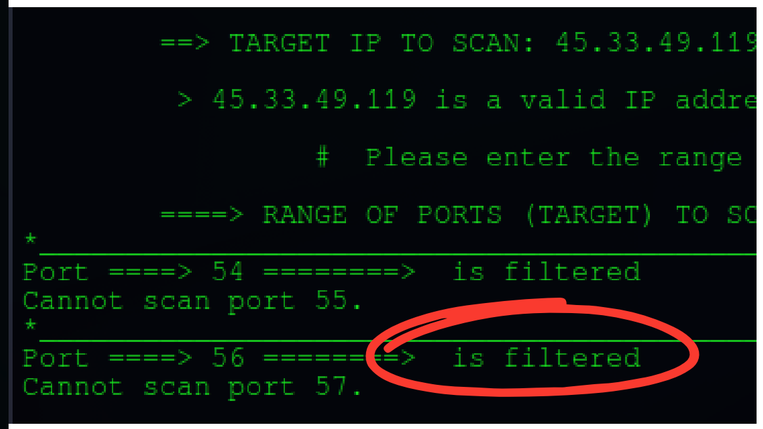
"Bench-4-Scan" est un outil écrit en Python qui se compose de différentes parties et d'options permettant de scanner une machine cible, ainsi que de mener à bien le processus d'enumeration d'une cible.

INTERFACE


BENCH-4-SCAN
==> https://github.com/achraf-benchehla/bench44scan.py
==> aHR0cHM6Ly93d3cubGlua2VkaW4uY29tL2luL2JlbmNoZWhsYS1hY2hyYWYv
==> Author: Achraf Benchehla Contact: benchehlaachraf@gmail.com
base64
**** Choose your options to get started ****
* >1< SCAN PORT OF TARGET
* >#< SCAN WITH NMAP
+ >21< unknown SCAN
+ >22< OS TARGET DETECT
+ >23< FAST SCAN
+ >24< SCAN VERSION OF PROTOCOL
* >3< SCAN WITH MASSCAN
* >4< BRUTE_FORCE & Enumeraion
(**) FOR EXAMPLE : 1 , 21 , 22 , 23 , 24 , 3 , 4
* >1< SCAN PORT OF TARGET
* >#< SCAN WITH NMAP
+ >21< unknown SCAN
+ >22< OS TARGET DETECT
+ >23< FAST SCAN
+ >24< SCAN VERSION OF PROTOCOL
* >3< SCAN WITH MASSCAN
* >4< BRUTE_FORCE & Enumeraion
=====> : NUMBER OF OPTIONS _

SCAN PORT OF TARGET

- Dans l'option "Scan Port of Target", nous fournissons le statut des ports, qui peut être soit "ouvert" (open), "fermé" (closed), ou "filtré" (filtered).



- Port Ouvert (Open) : Le port est actif et répond aux requêtes réseau, permettant ainsi l'accès aux services qui y sont associés.
- Port Fermé (Closed) : Bien que le port soit physiquement ouvert, aucun service ne fonctionne à cet emplacement, ce qui entraîne un refus de connexion.
- Port Filtré (Filtered) : Les paquets réseau destinés à ce port sont bloqués ou filtrés, généralement par un pare-feu ou un dispositif de sécurité, rendant incertain l'état du port.

COMPARAISON AVEC NMAP

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -p73-88 127.0.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2023-09-03 11:35 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000018s latency).

PORT   STATE  SERVICE
73/tcp closed netrjs-3
74/tcp closed netrjs-4
75/tcp closed priv-dial
76/tcp closed deos
77/tcp closed priv-rje
78/tcp closed vettcp
79/tcp closed finger
80/tcp open   http
81/tcp closed hosts2-ns
82/tcp closed xfer
83/tcp closed mit-ml-dev
84/tcp closed ctf
85/tcp closed mit-ml-dev
86/tcp closed mfcobol
87/tcp closed priv-term-l
88/tcp closed kerberos-sec
```

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -p 56 45.33.49.119
Starting Nmap 7.92 ( https://nmap.org ) at 2023-09-03 12:18 EDT
Nmap scan report for ack.nmap.org (45.33.49.119)
Host is up (0.22s latency).

PORT   STATE    SERVICE
56/tcp filtered xns-auth
```

SCAN WITH NMAP

UNKNOWN SCAN

```
 ENTER  @IP OF THE TARGET

 FOR EXAMPLE: 127.0.0.1, 10.99.88.77, ...

  VALEUR OF IP  =======>   : 127.0.0.1


Starting Nmap 7.92 ( https://nmap.org ) at 2023-09-03 11:47 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up, received localhost-response (0.000011s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack ttl 64

Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
```

Cette option va lancer la commande "nmap" (1000 port)avec des options nous permettant d'effectuer un scan de manière à ce que la machine cible trouve un ensemble d'adresses qui envoient la même requête.

SCAN WITH NMAP

- **Dans ce mode, l'outil va exécuter Nmap avec des options pour détecter le système d'exploitation en cours de fonctionnement sur la machine cible.**

```
SCAN OS TARGET


 ENTER  @IP OF THE TARGET

 FOR EXAMPLE: 127.0.0.1, 10.99.88.77, ...

  VALEUR OF IP  =======>   : 127.0.0.1


Starting Nmap 7.92 ( https://nmap.org ) at 2023-09-03 11:56 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up, received localhost-response (0.000046s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack ttl 64
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.81 seconds
```

```
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack ttl 64
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.32
OS details: Linux 2.6.32
Network Distance: 0 hops
```

SCAN WITH NMAP

FAST SCAN

```
 ENTER  @IP OF THE TARGET

 FOR EXAMPLE: 127.0.0.1, 10.99.88.77, ...

  VALEUR OF IP  ======>   : 127.0.0.1


Starting Nmap 7.92 ( https://nmap.org ) at 2023-09-03 12:01 EDT
Initiating SYN Stealth Scan at 12:01
Scanning localhost (127.0.0.1) [1000 ports]
Discovered open port 80/tcp on 127.0.0.1
Completed SYN Stealth Scan at 12:01, 0.08s elapsed (1000 total ports)
Nmap scan report for localhost (127.0.0.1)
Host is up, received localhost-response (0.0000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack ttl 64

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
           Raw packets sent: 1000 (44.000KB) | Rcvd: 2001 (84.044KB)
```

Ce mode activera Nmap avec une vitesse élevée pour envoyer les paquets, ce qui permettra d'obtenir rapidement les résultats du scan sur la machine cible. Il est important de noter que ce mode n'est pas favorable en présence d'un pare-feu, car il est plus facile de détecter le scan.

```
Completed SYN Stealth Scan at 12:01, 0.08s elapsed (1000 total ports)
Nmap scan report for localhost (127.0.0.1)
Host is up, received localhost-response (0.0000040s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE REASON
80/tcp open  http    syn-ack ttl 64
```

SCAN WITH NMAP

SCAN VERSION OF PROTOCOL

```
 ENTER  @IP OF THE TARGET TO FIND THE VULNERABILITIES

 FOR EXAMPLE: 127.0.0.1, 10.99.88.77, ...

  VALEUR OF IP  ======>   : 127.0.0.1


Starting Nmap 7.92 ( https://nmap.org ) at 2023-09-03 12:05 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000030s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.57 ((Debian))
|_http-server-header: Apache/2.4.57 (Debian)
|_http-title: Page de connexion

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.55 seconds
Segmentation fault
```

```
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.57 ((Debian))
|_http-server-header: Apache/2.4.57 (Debian)
|_http-title: Page de connexion
```

Dans le mode "Scan Version of Protocol", l'outil exécutera Nmap avec des options qui nous permettront d'obtenir des informations sur la version du protocole utilisé sur chaque port de la machine cible.

BRUTE_FORCE & Enumeraion

BAT FR & ENUMERATION

```
        *  for example wordlist : /usr/share/wordlists/wordlist.txt

        ==>  Entrez le chemin de la wordlist: wordlist3.txt

        *  for example URL  : https://test.com , http://test2.com

        ==>  Entrez l'URL cible: https://google.com

       resultat :    ## 200 = succès de la requête  ,  404 =  ressource non trouvée  ,   301 = redirection  ##

https://google.com/account_login : 404
https://google.com/Account_Management : 404
https://google.com/accountability : 404
https://google.com/accountancy : 404
https://google.com/accountancyage : 404
https://google.com/accountant : 404
https://google.com/%%%%%%%wfaknvnnk rr ovnnkv eag vnen : 404
https://google.com/accounting : 404
https://google.com/Accounting : 404
https://google.com/accounting-finance : 404
https://google.com/accounting_software : 404
https://google.com/Accounting1 : 404
https://google.com/AccountingMyAccount : 404
https://google.com/Accounts : 404
https://google.com/accounts : 200
https://google.com/Accounts_Payable : 404
https://google.com/accountteam : 404
https://google.com/accountz : 404
https://google.com/accreditation : 404
https://google.com/accreditations : 404
https://google.com/acct : 404
https://google.com/acct_login : 404
https://google.com/Acct_Main : 404
https://google.com/acct_mgmt : 404
https://google.com/accu_logo : 404
https://google.com/Accueil : 404
https://google.com/accueil : 404
https://google.com/accuracy : 404
https://google.com/accutane : 404
```

```
https://google.com/accounting-finance : 404
https://google.com/accounting_software : 404
https://google.com/Accounting1 : 404
https://google.com/AccountingMyAccount : 404
https://google.com/Accounts : 404
https://google.com/accounts : 200
https://google.com/Accounts_Payable : 404
```


```
        *  for example wordlist : /usr/share/wordlists/wordlist.txt

        ==>  Entrez le chemin de la wordlist: wordlist3.txt

        *  for example URL  : https://test.com , http://test2.com

        ==>  Entrez l'URL cible: https://google.com

       resultat :    ## 200 = succès de la requête  ,  404 =  ressource non trouvée  ,

https://google.com/account_login : 404
https://google.com/Account_Management : 404
https://google.com/accountability : 404
https://google.com/accountancy : 404
https://google.com/accountancyage : 404
https://google.com/accountant : 404
https://google.com/%%%%%%%wfaknvnnk rr ovnnkv eag vnen : 404
https://google.com/accounting : 404
https://google.com/Accounting : 404
https://google.com/accounting-finance : 404
https://google.com/accounting_software : 404
https://google.com/Accounting1 : 404
https://google.com/AccountingMyAccount : 404
https://google.com/Accounts : 404
https://google.com/accounts : 200
https://google.com/Accounts_Payable : 404
```


La dernière option est "Brute Force & Enumeration". Cette option nous permet de tester tous les mots d'une liste (wordlist3.txt par exemple). L'outil enverra des requêtes à chaque URL en combinant chaque mot de la liste et nous indiquera si la requête a été acceptée avec le code 200 ou non avec le code 404.

FIN...