INE - Bruteforce and Password cracking Offline

ambiente de laboratório

Neste ambiente de laboratório, os usuários têm acesso a uma instância de GUI do Kali. Esta máquina possui ferramentas John the Ripper (JTR) e Hashcat.

Objetivo: Realizar as seguintes atividades:

1. John The Ripper/Hashcat quebra o hash do usuário "admin" e recupera a senha.
2. Use John The Ripper/Hashcat para encontrar senhas para arquivos .docx protegidos do Microsoft Office.

Use um dicionário de senhas: `/root/Desktop/wordlists/1000000-password-seclists.txt`

Quebrando o hash do usuário "admin" e recuperando a senha

JRT

```
root@INE:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:103:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:105::/nonexistent:/usr/sbin/nologin
rtkit:x:105:106:RealtimeKit,,,:/proc:/usr/sbin/nologin
xrdp:x:106:112::/run/xrdp:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
avahi:x:108:115:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
pulse:x:109:116:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
sddm:x:110:118:Simple Desktop Display Manager:/var/lib/sddm:/bin/false
geoclue:x:111:119::/var/lib/geoclue:/usr/sbin/nologin
tomcat:x:1000:1000::/opt/tomcat:/bin/false
mysql:x:112:120:MySQL Server,,,:/nonexistent:/bin/false
stunnel4:x:113:122::/var/run/stunnel4:/usr/sbin/nologin
_rpc:x:114:65534::/run/rpcbind:/usr/sbin/nologin
dnsmasq:x:115:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
sslh:x:116:124::/nonexistent:/usr/sbin/nologin
ntp:x:117:125::/nonexistent:/usr/sbin/nologin
arpwatch:x:118:127:ARP Watcher,,,:/var/lib/arpwatch:/bin/sh
Debian-exim:x:119:128::/var/spool/exim4:/usr/sbin/nologin
debian-tor:x:120:129::/var/lib/tor:/bin/false
redsocks:x:121:130::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:122:65534::/var/spool/rwho:/usr/sbin/nologin
freerad:x:123:131::/etc/freeradius:/usr/sbin/nologin
iodine:x:124:65534::/run/iodine:/usr/sbin/nologin
tcpdump:x:125:132::/nonexistent:/usr/sbin/nologin
miredo:x:126:65534::/var/run/miredo:/usr/sbin/nologin
statd:x:127:65534::/var/lib/nfs:/usr/sbin/nologin
```

```
redis:x:128:134::/var/lib/redis:/usr/sbin/nologin
postgres:x:129:135:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
inetsim:x:130:138::/var/lib/inetsim:/usr/sbin/nologin
sshd:x:131:65534::/run/sshd:/usr/sbin/nologin
Debian-snmp:x:132:139::/var/lib/snmp:/bin/false
_gvm:x:133:141::/var/lib/openvas:/usr/sbin/nologin
saned:x:134:142::/var/lib/saned:/usr/sbin/nologin
king-phisher:x:135:143::/var/lib/king-phisher:/usr/sbin/nologin
_caldera:x:136:144::/var/lib/caldera:/usr/sbin/nologin
dradis:x:137:145::/var/lib/dradis:/usr/sbin/nologin
beef-xss:x:138:146::/var/lib/beef-xss:/usr/sbin/nologin
uuidd:x:139:147::/run/uuidd:/usr/sbin/nologin
memcache:x:140:148:Memcached,,,:/nonexistent:/bin/false
arangodb:x:999:999:ArangoDB Application User:/usr/share/arangodb3:/bin/false
admin:x:1001:1001:,,,:/home/admin:/bin/bash
```

Os usuários de interesse são root e admin:

```
root:x:0:0:root:/root:/bin/bash
admin:x:1001:1001:,,,:/home/admin:/bin/bash
```

Então, confira `/etc/shadow`:

```
root@INE:~# cat /etc/shadow
root:$6$iF3IRYx/LG3fxScL$spZt1Ltgnu4zyhnD2nYDZ9FSSKKcV0fRwnA3JNIGveciJQTWkPgv3qhE.URwgSakGvrJFNdHrfwYc9JLEZiGF/:18604:0:99999:7:::
daemon:*:19002:0:99999:7:::
bin:*:19002:0:99999:7:::
sys:*:19002:0:99999:7:::
sync:*:19002:0:99999:7:::
games:*:19002:0:99999:7:::
man:*:19002:0:99999:7:::
lp:*:19002:0:99999:7:::
mail:*:19002:0:99999:7:::
news:*:19002:0:99999:7:::
uucp:*:19002:0:99999:7:::
proxy:*:19002:0:99999:7:::
www-data:*:19002:0:99999:7:::
backup:*:19002:0:99999:7:::
list:*:19002:0:99999:7:::
irc:*:19002:0:99999:7:::
gnats:*:19002:0:99999:7:::
nobody:*:19002:0:99999:7:::
_apt:*:19002:0:99999:7:::
systemd-network:*:19004:0:99999:7:::
systemd-resolve:*:19004:0:99999:7:::
systemd-timesync:*:19004:0:99999:7:::
messagebus:*:19004:0:99999:7:::
rtkit:*:19004:0:99999:7:::
xrdp:!:19004:0:99999:7:::
usbmux:*:19004:0:99999:7:::
avahi:*:19004:0:99999:7:::
pulse:*:19004:0:99999:7:::
sddm:*:19004:0:99999:7:::
geoclue:*:19004:0:99999:7:::
tomcat:!:19004:0:99999:7:::
mysql:!:19004:0:99999:7:::
stunnel4:!:19004:0:99999:7:::
_rpc:*:19004:0:99999:7:::
dnsmasq:*:19004:0:99999:7:::
sslh:!:19004:0:99999:7:::
ntp:*:19004:0:99999:7:::
arpwatch:!:19004:0:99999:7:::
Debian-exim:!:19004:0:99999:7:::
debian-tor:*:19004:0:99999:7:::
redsocks:!:19004:0:99999:7:::
rwhod:*:19004:0:99999:7:::
freerad:*:19004:0:99999:7:::
iodine:*:19004:0:99999:7:::
tcpdump:*:19004:0:99999:7:::
miredo:*:19004:0:99999:7:::
statd:*:19004:0:99999:7:::
redis:*:19004:0:99999:7:::
postgres:*:19004:0:99999:7:::
```

```
inetsim:*:19004:0:99999:7:::
sshd:*:19004:0:99999:7:::
Debian-snmp:!:19004:0:99999:7:::
_gvm:*:19004:0:99999:7:::
saned:*:19004:0:99999:7:::
king-phisher:*:19004:0:99999:7:::
_caldera:*:19004:0:99999:7:::
dradis:*:19004:0:99999:7:::
beef-xss:*:19004:0:99999:7:::
uuidd:*:19004:0:99999:7:::
memcache:!:19004:0:99999:7:::
arangodb:!:19004:::::
admin:$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/:18945:0:99999:7:::/
```

Descobriu que os usuários root e admin têm hashes:

```
root:$6$iF3IRYx/LG3fxScL$spZt1Ltgnu4zyhnD2nYDZ9FSSKKcV0fRwnA3JNIGveciJQTWkPgv3qhE.URwgSakGvrJFNdHrfwYc9JLEZiGF/:18604:0:99999:7:::
admin:$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/:18945:0:99999:7:::/
```

Use unshadow para mesclar `/etc/passwd` e `/etc/shadow`:

```
unshadow /etc/passwd /etc/shadow > hashes
```

Veja o arquivo mesclado:

```
root@INE:~# cat hashes
root:$6$iF3IRYx/LG3fxScL$spZt1Ltgnu4zyhnD2nYDZ9FSSKKcV0fRwnA3JNIGveciJQTWkPgv3qhE.URwgSakGvrJFNdHrfwYc9JLEZiGF/:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:*:2:2:bin:/bin:/usr/sbin/nologin
sys:*:3:3:sys:/dev:/usr/sbin/nologin
sync:*:4:65534:sync:/bin:/bin/sync
games:*:5:60:games:/usr/games:/usr/sbin/nologin
man:*:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:*:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:*:8:8:mail:/var/mail:/usr/sbin/nologin
news:*:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:*:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:*:13:13:proxy:/bin:/usr/sbin/nologin
www-data:*:33:33:www-data:/var/www:/usr/sbin/nologin
backup:*:34:34:backup:/var/backups:/usr/sbin/nologin
list:*:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:*:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:*:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:*:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:*:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:*:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:*:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:*:103:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:*:104:105::/nonexistent:/usr/sbin/nologin
rtkit:*:105:106:RealtimeKit,,,:/proc:/usr/sbin/nologin
xrdp:!:106:112::/run/xrdp:/usr/sbin/nologin
usbmux:*:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
avahi:*:108:115:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
pulse:*:109:116:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
sddm:*:110:118:Simple Desktop Display Manager:/var/lib/sddm:/bin/false
geoclue:*:111:119::/var/lib/geoclue:/usr/sbin/nologin
tomcat:!:1000:1000::/opt/tomcat:/bin/false
mysql:!:112:120:MySQL Server,,,:/nonexistent:/bin/false
stunnel4:!:113:122::/var/run/stunnel4:/usr/sbin/nologin
_rpc:*:114:65534::/run/rpcbind:/usr/sbin/nologin
dnsmasq:*:115:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
sslh:!:116:124::/nonexistent:/usr/sbin/nologin
ntp:*:117:125::/nonexistent:/usr/sbin/nologin
arpwatch:!:118:127:ARP Watcher,,,:/var/lib/arpwatch:/bin/sh
Debian-exim:!:119:128::/var/spool/exim4:/usr/sbin/nologin
debian-tor:*:120:129::/var/lib/tor:/bin/false
redsocks:!:121:130::/var/run/redsocks:/usr/sbin/nologin
rwhod:*:122:65534::/var/spool/rwho:/usr/sbin/nologin
freerad:*:123:131::/etc/freeradius:/usr/sbin/nologin
iodine:*:124:65534::/run/iodine:/usr/sbin/nologin
tcpdump:*:125:132::/nonexistent:/usr/sbin/nologin
miredo:*:126:65534::/var/run/miredo:/usr/sbin/nologin
```

```
statd:*:127:65534::/var/lib/nfs:/usr/sbin/nologin
redis:*:128:134::/var/lib/redis:/usr/sbin/nologin
postgres:*:129:135:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
inetsim:*:130:138::/var/lib/inetsim:/usr/sbin/nologin
sshd:*:131:65534::/run/sshd:/usr/sbin/nologin
Debian-snmp:!:132:139::/var/lib/snmp:/bin/false
_gvm:*:133:141::/var/lib/openvas:/usr/sbin/nologin
saned:*:134:142::/var/lib/saned:/usr/sbin/nologin
king-phisher:*:135:143::/var/lib/king-phisher:/usr/sbin/nologin
_caldera:*:136:144::/var/lib/caldera:/usr/sbin/nologin
dradis:*:137:145::/var/lib/dradis:/usr/sbin/nologin
beef-xss:*:138:146::/var/lib/beef-xss:/usr/sbin/nologin
uuidd:*:139:147::/run/uuidd:/usr/sbin/nologin
memcache:!:140:148:Memcached,,,:/nonexistent:/bin/false
arangodb:!:999:999:ArangoDB Application User:/usr/share/arangodb3:/bin/false
admin:$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/:1001:1001:,,,:/home/admin:
```

As informações válidas são:

```
root:$6$iF3IRYx/LG3fxScL$spZt1Ltgnu4zyhnD2nYDZ9FSSKKcV0fRwnA3JNIGveciJQTWkPgv3qhE.URwgSakGvrJFNdHrfwYc9JLEZiGF/:0:0:root:/root:/bin/bash
admin:$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/:1001:1001:,,,:/home/admin:
```

Usando um dicionário de senhas `/root/Desktop/wordlists/1000000-password-seclists.txt`, carregue apenas os usuários root e admin:

```
john --wordlist=/root/Desktop/wordlists/1000000-password-seclists.txt --users=root,admin hashes
```

```
root@INE:~# john --wordlist=/root/Desktop/wordlists/1000000-password-seclists.txt --users=root,admin hashes
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 48 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
foxtrot          (admin)
password         (root)
2g 0:00:00:08 DONE (2022-07-15 07:22) 0.2392g/s 734.9p/s 1469c/s 1469C/s 123456..786786
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Mostrar todas as senhas quebradas: `root:password`, `admin:foxtrot`

```
john --show hashes
```

```
root@INE:~# john --show hashes
root:password:0:0:root:/root:/bin/bash
admin:foxtrot:1001:1001:,,,:/home/admin:/bin/bash

2 password hashes cracked, 0 left
```

- `-show /root/.john/john.pot`

  O conteúdo exibido na verdade vem do

  arquivo:

```
root@INE:~# cat /root/.john/john.pot
$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/:foxtrot
$6$iF3IRYx/LG3fxScL$spZt1Ltgnu4zyhnD2nYDZ9FSSKKcV0fRwnA3JNIGveciJQTWkPgv3qhE.URwgSakGvrJFNdHrfwYc9JLEZiGF/:password
```

hashcat

Ao contrário de John, a maneira mais fácil de usar o Hashcat é apenas fornecer o próprio hash da senha. Copie quaisquer hashes que queremos quebrar em um novo arquivo de texto, vamos chamá-lo de hashes.txt:

```
$6$iF3IRYx/LG3fxScL$spZt1Ltgnu4zyhnD2nYDZ9FSSKKcV0fRwnA3JNIGveciJQTWkPgv3qhE.URwgSakGvrJFNdHrfwYc9JLEZiGF/
$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/
```

Hashcat contém vários modos que podem ser executados, dependendo do tipo de hash usado. Vimos anteriormente que John identificou nosso hash de sombra como `sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x]`, para que possamos digitar `hashcat --help` para revelar todas as opções dessa ferramenta e os diferentes modos disponíveis. Na lista, descobrimos que `sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x]` é modo 1800:

```
root@INE:~# hashcat --help | grep sha512
   1770 | sha512(utf16le($pass))                             | Raw Hash
   1710 | sha512($pass.$salt)                                | Raw Hash salted and/or iterated
   1720 | sha512($salt.$pass)                                | Raw Hash salted and/or iterated
   1740 | sha512($salt.utf16le($pass))                       | Raw Hash salted and/or iterated
   1730 | sha512(utf16le($pass).$salt)                       | Raw Hash salted and/or iterated
   6500 | AIX {ssha512}                                      | Operating System
   1800 | sha512crypt $6$, SHA512 (Unix)                     | Operating System
  21600 | Web2py pbkdf2-sha512                               | Framework
  20200 | Python passlib pbkdf2-sha512                       | Framework
  21000 | BitShares v0.x - sha512(sha512_bin(pass))          | Cryptocurrency Wallet
```

```
hashcat -m 1800 -a 0 -o cracked.txt hashes.txt /root/Desktop/wordlists/1000000-password-seclists.txt
```

- `m`

  : Especifica o modo a ser usado.

- `a`

  : Determina o tipo de ataque, 0 significa o modo direto padrão.

- `o` `/root/Desktop/wordlists/1000000-password-seclists.txt`

  : Especifica que o arquivo de saída é cracked.txt e o arquivo de entrada é hashes.txt contendo hashes, usando um dicionário

  .

```
root@INE:~# hashcat -m 1800 -a 0 -o cracked.txt hashes.txt /root/Desktop/wordlists/1000000-password-seclists.txt
hashcat (v6.2.5) starting

OpenCL API (OpenCL 2.0 pocl 1.8  Linux, None+Asserts, RELOC, LLVM 11.1.0, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: pthread-AMD EPYC 7642 48-Core Processor, 47227/94518 MB (16384 MB allocatable), 48MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 2 digests; 2 unique digests, 2 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Uses-64-Bit

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB


Dictionary cache built:
* Filename..: /root/Desktop/wordlists/1000000-password-seclists.txt
* Passwords.: 1000003
```

```
* Bytes.....: 8529147
* Keyspace..: 1000003
* Runtime...: 0 secs


Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1800 (sha512crypt $6$, SHA512 (Unix))
Hash.Target......: hashes.txt
Time.Started.....: Fri Jul 15 08:09:14 2022 (6 secs)
Time.Estimated...: Fri Jul 15 08:09:20 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/root/Desktop/wordlists/1000000-password-seclists.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:      548 H/s (42.64ms) @ Accel:1024 Loops:128 Thr:1 Vec:4
Recovered........: 2/2 (100.00%) Digests, 2/2 (100.00%) Salts
Progress.........: 4096/2000006 (0.20%)
Rejected.........: 0/4096 (0.00%)
Restore.Point....: 1024/1000003 (0.10%)
Restore.Sub.#1...: Salt:1 Amplifier:0-1 Iteration:4992-5000
Candidate.Engine.: Device Generator
Candidates.#1....: sports -> ball

Started: Fri Jul 15 08:08:06 2022
Stopped: Fri Jul 15 08:09:21 2022
```

Exiba o conteúdo de cracked.txt e visualize a senha em texto simples:

```
root@INE:~# cat cracked.txt
$6$iF3IRYx/LG3fxScL$spZt1Ltgnu4zyhnD2nYDZ9FSSKKcV0fRwnA3JNIGveciJQTWkPgv3qhE.URwgSakGvrJFNdHrfwYc9JLEZiGF/:password
$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/:foxtrot
```

Encontre senhas para arquivos .docx protegidos do Microsoft Office

JRT

Referência: Quebra de arquivos do Microsoft Office protegidos por senha, incluindo documentos do Word e planilhas do Excel

Primeiro instale o office2john

Extraia hashes de arquivos do Office protegidos por senha

```
python office2johh.py Desktop/MS_Word_Document.docx > hash.txt
cat hash.txt
```

O hash salvo corresponde ao Microsoft Office 2013

```
MS_Word_Document.docx:$office$*2013*100000*256*16*ff2563844faca58a12fc42c5036f9cf8*ffaf52db903dbcb6ac2db4bab6d343ab*c237403ec97e5f68b7be332
```

```
john --wordlist=/root/Desktop/wordlists/1000000-password-seclists.txt hash.txt
```

```
root@INE:~# john --wordlist=/root/Desktop/wordlists/1000000-password-seclists.txt hash.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (Office, 2007/2010/2013 [SHA1 256/256 AVX2 8x / SHA512 256/256 AVX2 4x AES])
Cost 1 (MS Office version) is 2013 for all loaded hashes
Cost 2 (iteration count) is 100000 for all loaded hashes
Will run 48 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
muenchen         (MS_Word_Document.docx)
1g 0:00:05:08 DONE (2022-07-15 11:20) 0.003240g/s 77.14p/s 77.14c/s 77.14C/s shake..963741
```

```
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

```
cat /root/.john/john.pot
john --show hash.txt
```

A senha para o documento MS_Word_Document.docx é `muenchen`.

```
root@INE:~# cat /root/.john/john.pot
$office$*2013*100000*256*16*ff2563844faca58a12fc42c5036f9cf8*ffaf52db903dbcb6ac2db4bab6d343ab*c237403ec97e5f68b7be3324a8633c9ff95e0bb44b1ef
root@INE:~# john --show hash.txt
MS_Word_Document.docx:muenchen

1 password hash cracked, 0 left
```

hashcat

Referência: crack a proteção por senha do Microsoft Office via hashcat localmente ou na nuvem

Verifique as propriedades do arquivo de MS_Word_Document.docx, o tipo de arquivo é documento do Word 2007.

Use office2hashcat.py para recuperar o hash criptográfico de um documento do Word protegido por senha, salvando-o em hash.txt:

```
./office2hashcat.py Desktop/MS_Word_Document.docx > hash.txt
cat hash.txt
```

ID de hash para MS Office 2013

```
$office$*2013*100000*256*16*ff2563844faca58a12fc42c5036f9cf8*ffaf52db903dbcb6ac2db4bab6d343ab*c237403ec97e5f68b7be3324a8633c9ff95e0bb44b1ef
```

Foi atribuído o número de modo de hash 9600 por hashcat

```
root@INE:~# hashcat --help | grep "MS Office"
   9400 | MS Office 2007                                  | Document
   9500 | MS Office 2010                                  | Document
   9600 | MS Office 2013                                  | Document
  25300 | MS Office 2016 - SheetProtection                | Document
   9700 | MS Office <= 2003 $0/$1, MD5 + RC4              | Document
   9710 | MS Office <= 2003 $0/$1, MD5 + RC4, collider #1 | Document
   9720 | MS Office <= 2003 $0/$1, MD5 + RC4, collider #2 | Document
   9810 | MS Office <= 2003 $3, SHA1 + RC4, collider #1   | Document
   9820 | MS Office <= 2003 $3, SHA1 + RC4, collider #2   | Document
   9800 | MS Office <= 2003 $3/$4, SHA1 + RC4             | Document
```

Passe os seguintes parâmetros para o hashcat:

- `a 0`

  : defina o modo de ataque para ataque direto/dicionário

- `m 9600`

  : defina o modo de hash para MS Office 2013

- `-status`

  : Tela de status de atualização automática

- `o found.txt`

  : Saída da senha recuperada para found.txt

- `hash.txt`

  : o hash salvo na etapa anterior

- `/root/Desktop/wordlists/1000000-password-seclists.txt`

  : dicionário de senhas

```
hashcat -a 0 -m 9600 --status -o found.txt hash.txt /root/Desktop/wordlists/1000000-password-seclists.txt
```

```
root@INE:~# hashcat -a 0 -m 9600 --status -o found.txt hash.txt /root/Desktop/wordlists/1000000-password-seclists.txt
hashcat (v6.2.5) starting

OpenCL API (OpenCL 2.0 pocl 1.8  Linux, None+Asserts, RELOC, LLVM 11.1.0, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
============================================================================================================================================
* Device #1: pthread-AMD EPYC 7642 48-Core Processor, 47227/94518 MB (16384 MB allocatable), 48MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Slow-Hash-SIMD-LOOP
* Uses-64-Bit

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename..: /root/Desktop/wordlists/1000000-password-seclists.txt
* Passwords.: 1000003
* Bytes.....: 8529147
* Keyspace..: 1000003
* Runtime...: 0 secs

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit =>

Session..........: hashcat
Status...........: Running
Hash.Mode........: 9600 (MS Office 2013)
Hash.Target......: $office$*2013*100000*256*16*ff2563844faca58a12fc42c...4336a2
Time.Started.....: Fri Jul 15 10:21:09 2022 (7 secs)
Time.Estimated...: Fri Jul 15 13:46:06 2022 (3 hours, 24 mins)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/root/Desktop/wordlists/1000000-password-seclists.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:       81 H/s (6.91ms) @ Accel:1024 Loops:64 Thr:1 Vec:4
Recovered........: 0/1 (0.00%) Digests
Progress.........: 0/1000003 (0.00%)
Rejected.........: 0/0 (0.00%)
Restore.Point....: 0/1000003 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:56448-56512
Candidate.Engine.: Device Generator
Candidates.#1....: 123456 -> time

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit =>

Session..........: hashcat
Status...........: Running
Hash.Mode........: 9600 (MS Office 2013)
Hash.Target......: $office$*2013*100000*256*16*ff2563844faca58a12fc42c...4336a2
Time.Started.....: Fri Jul 15 10:21:09 2022 (4 mins, 19 secs)
Time.Estimated...: Fri Jul 15 13:28:41 2022 (3 hours, 3 mins)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/root/Desktop/wordlists/1000000-password-seclists.txt)
Guess.Queue......: 1/1 (100.00%)
```

```
Speed.#1.........:       89 H/s (6.10ms) @ Accel:1024 Loops:64 Thr:1 Vec:4
Recovered........: 0/1 (0.00%) Digests
Progress.........: 22528/1000003 (2.25%)
Rejected.........: 0/22528 (0.00%)
Restore.Point....: 22528/1000003 (2.25%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:51200-51264
Candidate.Engine.: Device Generator
Candidates.#1....: sandeep -> luckyboy

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 9600 (MS Office 2013)
Hash.Target......: $office$*2013*100000*256*16*ff2563844faca58a12fc42c...4336a2
Time.Started.....: Fri Jul 15 10:21:09 2022 (4 mins, 24 secs)
Time.Estimated...: Fri Jul 15 10:25:33 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/root/Desktop/wordlists/1000000-password-seclists.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:       89 H/s (6.16ms) @ Accel:1024 Loops:64 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests
Progress.........: 23552/1000003 (2.36%)
Rejected.........: 0/23552 (0.00%)
Restore.Point....: 22528/1000003 (2.25%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: sandeep -> luckyboy
Started: Fri Jul 15 10:19:58 2022
Stopped: Fri Jul 15 10:25:36 2022
```

A senha para o documento MS_Word_Document.docx é `muenchen`.

```
root@INE:~# cat found.txt
$office$*2013*100000*256*16*ff2563844faca58a12fc42c5036f9cf8*ffaf52db903dbcb6ac2db4bab6d343ab*c237403ec97e5f68b7be3324a8633c9ff95e0bb44b1ef
```

solução

Quebrando o hash do usuário do Linux

Etapa 1: abra o link do laboratório para acessar a instância da GUI do Kali.

Etapa 2: Nos sistemas operacionais Linux, o hash da senha do usuário é armazenado em um `/etc/shadow` arquivo . Use o comando para `cat` ler o arquivo e verifique a entrada do usuário "admin".

Ordem

```
cat /etc/shadow
```

```
root@INE:~# cat /etc/shadow
root:$6$iF3IRYx/LG3fxScL$spZt1Ltgnu4zyhnD2nYDZ9FSSKKcV0fRwnA3JNIGveciJQTWkPgv3qhE.URwgSakGvrJFNdHrfwYc9JLEZiGF/:18604:0:99999:7:::
daemon:*:19002:0:99999:7:::
bin:*:19002:0:99999:7:::
sys:*:19002:0:99999:7:::
sync:*:19002:0:99999:7:::
games:*:19002:0:99999:7:::
man:*:19002:0:99999:7:::
lp:*:19002:0:99999:7:::
mail:*:19002:0:99999:7:::
news:*:19002:0:99999:7:::
uucp:*:19002:0:99999:7:::
proxy:*:19002:0:99999:7:::
www-data:*:19002:0:99999:7:::
backup:*:19002:0:99999:7:::
list:*:19002:0:99999:7:::
irc:*:19002:0:99999:7:::
gnats:*:19002:0:99999:7:::
nobody:*:19002:0:99999:7:::
_apt:*:19002:0:99999:7:::
```

```
systemd-network:*:19004:0:99999:7:::
systemd-resolve:*:19004:0:99999:7:::
systemd-timesync:*:19004:0:99999:7:::
messagebus:*:19004:0:99999:7:::
rtkit:*:19004:0:99999:7:::
xrdp:!:19004:0:99999:7:::
usbmux:*:19004:0:99999:7:::
avahi:*:19004:0:99999:7:::
pulse:*:19004:0:99999:7:::
sddm:*:19004:0:99999:7:::
geoclue:*:19004:0:99999:7:::
tomcat:!:19004:0:99999:7:::
mysql:!:19004:0:99999:7:::
stunnel4:!:19004:0:99999:7:::
_rpc:*:19004:0:99999:7:::
dnsmasq:*:19004:0:99999:7:::
sslh:!:19004:0:99999:7:::
ntp:*:19004:0:99999:7:::
arpwatch:!:19004:0:99999:7:::
Debian-exim:!:19004:0:99999:7:::
debian-tor:*:19004:0:99999:7:::
redsocks:!:19004:0:99999:7:::
rwhod:*:19004:0:99999:7:::
freerad:*:19004:0:99999:7:::
iodine:*:19004:0:99999:7:::
tcpdump:*:19004:0:99999:7:::
miredo:*:19004:0:99999:7:::
statd:*:19004:0:99999:7:::
redis:*:19004:0:99999:7:::
postgres:*:19004:0:99999:7:::
inetsim:*:19004:0:99999:7:::
sshd:*:19004:0:99999:7:::
Debian-snmp:!:19004:0:99999:7:::
_gvm:*:19004:0:99999:7:::
saned:*:19004:0:99999:7:::
king-phisher:*:19004:0:99999:7:::
_caldera:*:19004:0:99999:7:::
dradis:*:19004:0:99999:7:::
beef-xss:*:19004:0:99999:7:::
uuidd:*:19004:0:99999:7:::
memcache:!:19004:0:99999:7:::
arangodb:!:19004:::::
admin:$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/:18945:0:99999:7:::/
```

O hash do usuário administrador é descrito abaixo:

hash do usuário administrador

```
admin:$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/:18945:0:99999:7:::
```

Etapa 3: verifique o algoritmo de hash usado para fazer o hash da senha presente no arquivo shadow verificando `/etc/login.defs`

Ordem

```
grep -A 18 ENCRYPT_METHOD /etc/login.defs
```

```
root@INE:~# grep -A 18 ENCRYPT_METHOD /etc/login.defs
# This variable is deprecated. You should use ENCRYPT_METHOD.
#
#MD5_CRYPT_ENAB no

#
# If set to MD5 , MD5-based algorithm will be used for encrypting password
# If set to SHA256, SHA256-based algorithm will be used for encrypting password
# If set to SHA512, SHA512-based algorithm will be used for encrypting password
# If set to DES, DES-based algorithm will be used for encrypting password (default)
# Overrides the MD5_CRYPT_ENAB option
#
# Note: It is recommended to use a value consistent with
# the PAM modules configuration.
#
```

```
ENCRYPT_METHOD SHA512

#
# Only used if ENCRYPT_METHOD is set to SHA256 or SHA512.
#
# Define the number of SHA rounds.
# With a lot of rounds, it is more difficult to brute forcing the password.
# But note also that it more CPU resources will be needed to authenticate
# users.
#
# If not specified, the libc will choose the default number of rounds (5000).
# The values must be inside the 1000-999999999 range.
# If only one of the MIN or MAX values is set, then this value will be used.
# If MIN > MAX, the highest value will be used.
#
# SHA_CRYPT_MIN_ROUNDS 5000
# SHA_CRYPT_MAX_ROUNDS 5000

################# OBSOLETED BY PAM ##############
#                                               #
# These options are now handled by PAM. Please  #
# edit the appropriate file in /etc/pam.d/ to   #
```

Use um algoritmo de hash `SHA512`.

Passo 4: Copie a entrada para o usuário "admin" e salve-a em um novo arquivo.

Ordem

```
tail -n 1 /etc/shadow > admin.hash
cat admin.hash
```

- `n 1`

  : Leia a última linha do arquivo /etc/shadow (entrada admin é a última linha do arquivo shadow)

```
root@INE:~# tail -n 1 /etc/shadow > admin.hash
root@INE:~# cat admin.hash
admin:$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/:18945:0:99999:7:::/
```

Passo 5: Temos que modificar a entrada para ser compatível com a entrada da ferramenta. Remova "admin:" e ":18945:0:99999:7:::"

```
admin:$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/:18945:0:99999:7:::/
```

```
$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/
```

Após a modificação, a entrada ficará assim:

hash final

```
$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/
```

Usando Hashcat

Passo 6: Use a ferramenta Hashcat para quebrar o hash

Ordem

```
hashcat -m 1800 -a 0 admin.hash /root/Desktop/wordlists/1000000-password-seclists.txt
```

Hashcat:

- Código aberto, ferramenta poderosa de quebra de senha
- Suporta vários sistemas operacionais (Linux, Windows e macOS)
- Suporta mais de 350 algoritmos

```
root@INE:~# hashcat -m 1800 -a 0 admin.hash /root/Desktop/wordlists/1000000-password-seclists.txt
hashcat (v6.2.5) starting

OpenCL API (OpenCL 2.0 pocl 1.8  Linux, None+Asserts, RELOC, LLVM 11.1.0, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
====================================================================================================================================
* Device #1: pthread-AMD EPYC 7642 48-Core Processor, 47227/94518 MB (16384 MB allocatable), 48MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Uses-64-Bit

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 0 MB

Dictionary cache built:
* Filename..: /root/Desktop/wordlists/1000000-password-seclists.txt
* Passwords.: 1000003
* Bytes.....: 8529147
* Keyspace..: 1000003
* Runtime...: 0 secs

$6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9isy7knhpskIEQG.s9eB8auxVqrroksib7uQyiCtrJIgr48XmR8o7Pa7O/:foxtrot

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 1800 (sha512crypt $6$, SHA512 (Unix))
Hash.Target......: $6$2PjhBcvO4tMWKi5W$k/UUyb5mb3qTJ6Fr15cReTb0n/DQ9is...7Pa7O/
Time.Started.....: Fri Jul 15 17:47:07 2022 (4 secs)
Time.Estimated...: Fri Jul 15 17:47:11 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/root/Desktop/wordlists/1000000-password-seclists.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:      538 H/s (9.40ms) @ Accel:1024 Loops:32 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests
Progress.........: 2048/1000003 (0.20%)
Rejected.........: 0/2048 (0.00%)
Restore.Point....: 1024/1000003 (0.10%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4992-5000
Candidate.Engine.: Device Generator
Candidates.#1....: sports -> ball

Started: Fri Jul 15 17:45:57 2022
Stopped: Fri Jul 15 17:47:13 2022
```

A senha para o usuário "admin" é foxtrot.

Etapa 7: quebrando o hash usando JTR

Podemos executar o JTR diretamente no arquivo shadow.

Ordem

```
john /etc/shadow --wordlist=/root/Desktop/wordlists/1000000-password-seclists.txt
```

- Ferramenta de quebra de senha gratuita e de código aberto
- Suporta vários sistemas operacionais (Linux, Windows e macOS)

```
root@INE:~# john /etc/shadow --wordlist=/root/Desktop/wordlists/1000000-password-seclists.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 48 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
foxtrot          (admin)
password         (root)
2g 0:00:00:07 DONE (2022-07-15 17:52) 0.2531g/s 777.7p/s 1555c/s 1555C/s 123456..786786
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

root@INE:~# john --show /etc/shadow
root:password:18604:0:99999:7:::
admin:foxtrot:18945:0:99999:7:::/

2 password hashes cracked, 0 left
```

As senhas para os usuários "root" e "admin" são senha e foxtrot respectivamente.

Quebre a senha de um arquivo do Microsoft Word

Passo 8: Vamos quebrar o arquivo "MS_Word_Document.docx" usando JTR e Hashcat.

Localização do arquivo MS_Word_Document.docx `/root/Desktop/MS_Word_Document.docx`

Usando JTR

Primeiro, precisamos `office2john` extrair as informações quebráveis do arquivo usando um script python.

Ordem

```
cd /root/Desktop/
/usr/share/john/office2john.py MS_Word_Document.docx > hash
cat hash
```

`office2john.py` é um script python para extrair informações quebráveis de arquivos .docx do Microsoft Office.

```
root@INE:~# cd /root/Desktop/
root@INE:~/Desktop# /usr/share/john/office2john.py MS_Word_Document.docx > hash
root@INE:~/Desktop# cat hash
MS_Word_Document.docx:$office$*2013*100000*256*16*ff2563844faca58a12fc42c5036f9cf8*ffaf52db903dbcb6ac2db4bab6d343ab*c237403ec97e5f68b7be332
```

Podemos observar que o arquivo é salvo da versão MS Office 2013.

Use a ferramenta John The Ripper para quebrar o hash.

Ordem

```
john --wordlist=/root/Desktop/wordlists/1000000-password-seclists.txt hash
```

```
root@INE:~/Desktop# john --wordlist=/root/Desktop/wordlists/1000000-password-seclists.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (Office, 2007/2010/2013 [SHA1 256/256 AVX2 8x / SHA512 256/256 AVX2 4x AES])
Cost 1 (MS Office version) is 2013 for all loaded hashes
Cost 2 (iteration count) is 100000 for all loaded hashes
Will run 48 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
muenchen         (MS_Word_Document.docx)
1g 0:00:04:53 DONE (2022-07-15 18:09) 0.003405g/s 81.08p/s 81.08c/s 81.08C/s shake..963741
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

A senha usada para criptografar o arquivo docx é **muenchen**

Usando Hashcat

Passo 9: Novamente, usaremos a ferramenta Hashcat para recuperar a senha.

Precisamos tornar as informações quebráveis extraídas compatíveis com a ferramenta Hashcat. Remova o `MS_Word_Document.docx:` valor do arquivo.

```
MS_Word_Document.docx:$office$*2013*100000*256*16*ff2563844faca58a12fc42c5036f9cf8*ffaf52db903dbcb6ac2db4bab6d343ab*c237403ec97e5f68b7be332
```

Exclua a área destacada.

```
$office$*2013*100000*256*16*ff2563844faca58a12fc42c5036f9cf8*ffaf52db903dbcb6ac2db4bab6d343ab*c237403ec97e5f68b7be3324a8633c9ff95e0bb44b1ef
```

conteúdo final

```
$office$*2013*100000*256*16*ff2563844faca58a12fc42c5036f9cf8*ffaf52db903dbcb6ac2db4bab6d343ab*c237403ec97e5f68b7be3324a8633c9ff95e0bb44b1ef
```

Quebrando o hash com Hashcat

Ordem

```
hashcat -a 0 -m 9600 --status hash /root/Desktop/wordlists/1000000-password-seclists.txt --force
```

Descrição do comando:

- `a 0`

  : defina o modo de ataque para o dicionário

- `m 9600`

  : Defina o método para MS Office 2013

- `-status`

  : Habilitar atualizações automáticas da tela de status

`hash`: arquivo contendo informações quebráveis

`/root/Desktop/wordlists/1000000-password-seclists.txt`: vocabulário de senha

```
$office$*2013*100000*256*16*ff2563844faca58a12fc42c5036f9cf8*ffaf52db903dbcb6ac2db4bab6d343ab*c237403ec97e5f68b7be3324a8633c9ff95e0bb44b1ef
Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 9600 (MS Office 2013)
Hash.Target......: $office$*2013*100000*256*16*ff2563844faca58a12fc42c...4336a2
Time.Started.....: Fri Jul 15 18:22:48 2022, (4 mins, 23 secs)
Time.Estimated...: Fri Jul 15 18:27:11 2022, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/root/Desktop/wordlists/1000000-password-seclists.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:       89 H/s (13.95ms) @ Accel:1024 Loops:128 Thr:1 Vec:4
Recovered........: 1/1 (100.00%) Digests
Progress.........: 23552/1000003 (2.36%)
Rejected.........: 0/23552 (0.00%)
Restore.Point....: 22528/1000003 (2.25%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: sandeep -> luckyboy
Started: Fri Jul 15 18:22:07 2022
Stopped: Fri Jul 15 18:27:14 2022
```

A senha usada para criptografar o arquivo docx é `muenchen`

É assim que usamos Hashcat e JTR para quebrar hashes e recuperar senhas para arquivos de documentos criptografados da Microsoft