

Lab 3.1

By: Nathan Metens (metens@pdx.edu)

Task 1: Using Base64 and OpenSSL	1
Task 2: Cryptographic hash functions	3
Task 3: Salt & Stretch	5
Task 4: Hash types	6
Task 5: Cracking passwords with hashcat	7
Task 6: Cracking SSH passphrases	9

Task 1: Using Base64 and OpenSSL

```
metens@cs591-metens-kali-1: ~/Lab-3.1
File Actions Edit View Help

└──(metens@cs591-metens-kali-1)-[~]
    $ mkdir Lab-3.1

└──(metens@cs591-metens-kali-1)-[~]
    $ cd Lab-3.1

└──(metens@cs591-metens-kali-1)-[~/Lab-3.1]
    $ echo -ne '\x0\x0\x0' | openssl enc -base64
AAAA

└──(metens@cs591-metens-kali-1)-[~/Lab-3.1]
    $ python3 -c 'import sys; sys.stdout.buffer.write(3*b"\x00")' | openssl enc -base64
AAAA

└──(metens@cs591-metens-kali-1)-[~/Lab-3.1]
    $ python3 -c 'import sys; sys.stdout.buffer.write(3*b"\x00")' > input.txt

└──(metens@cs591-metens-kali-1)-[~/Lab-3.1]
    $ 
```

```
└──(metens@cs591-metens-kali-1)-[~/Lab-3.1]
    $ ls
    OWNED by metens
    input.txt

    └──(metens@cs591-metens-kali-1)-[~/Lab-3.1]
        $ cat input.txt
        AAAA

    └──(metens@cs591-metens-kali-1)-[~/Lab-3.1]
        $ openssl enc -base64 -A -in input.txt -out output.txt

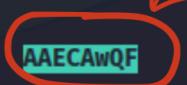
    └──(metens@cs591-metens-kali-1)-[~/Lab-3.1]
        $ xxd output.txt
        00000000: 4141 4141
        AAAA

    └──(metens@cs591-metens-kali-1)-[~/Lab-3.1]
        $ cat output.txt
        AAAA

    └──(metens@cs591-metens-kali-1)-[~/Lab-3.1]
        $ 
```

```
(metens@cs591-metens-kali-1) [~/Lab-3.1]
$ openssl enc -base64 -A -d -in output.txt -out decode.txt; xxd decode.txt
00000000: 0000 00 ...
```

```
metens@cs591-metens-kali-1
File Actions Edit View Help
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ echo -ne '\x00\x00\x00' | openssl enc -base64
AAAA
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ echo -ne '\x00\x01\x02\x03\x04\x05' | openssl enc -base64
AAECAwQF
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ echo -ne '\x00\x01\x02\x03\x04\x05' > input.txt
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ cat input.txt
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ openssl enc -base64 -A -d -in input.txt -out output.txt
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ xxd output.txt
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ vim input.txt
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ openssl enc -base64 -A -in input.txt -out output.txt
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ xxd output.txt
00000000: 4141 4543 4177 5146
```



```
metens@cs591-metens-kali-1
File Actions Edit View Help
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ echo 'ABCDEFGH' > encoded.txt
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ cat encoded.txt
ABCDEFGH
[metens@cs591-metens-kali-1] ~/Lab-3.1
$ openssl enc -base64 -d -in encoded.txt -out input.txt; xxd input.txt
00000000: 0010 8310 5187 ... Q.
[metens@cs591-metens-kali-1] ~/Lab-3.1
$
```

Task 2: Cryptographic hash functions

```
metens@cs591-metens-kali-1: ~/Lab-3.1
File Actions Edit View Help
(chatpt.com) 68797705.html:8000 [d47551d96347ab5]
(metens@cs591-metens-kali-1)-[~/Lab-3.1]
$ echo -n 'Password' | openssl dgst -sha256
SHA2-256(stdin)= e7cf3ef4f17c3999a94f2c6f612e8a888e5b1026878e4e19398b23bd38ec221a

(metens@cs591-metens-kali-1)-[~/Lab-3.1]
$ echo -n 'password' | openssl dgst -sha256
SHA2-256(stdin)= 5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8

(metens@cs591-metens-kali-1)-[~/Lab-3.1]
$ echo -n 'Password' | openssl dgst -sha256 | awk '{print $2}'
e7cf3ef4f17c3999a94f2c6f612e8a888e5b1026878e4e19398b23bd38ec221a

(metens@cs591-metens-kali-1)-[~/Lab-3.1]
$ echo -n 'password' | openssl dgst -sha256 | awk '{print $2}'
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8

(metens@cs591-metens-kali-1)-[~/Lab-3.1]
$ 
```

The screenshot shows the CrackStation website interface. At the top, there's a navigation bar with 'CrackStation', 'Password Hashing Security', and 'Defuse Security'. Below that is a sub-navigation bar with 'Free Password Hash Cracker'. The main area has a heading 'Enter up to 20 non-salted hashes, one per line:' followed by a text input field containing the SHA-256 hash 'e7cf3ef4f17c3999a94f2c6f612e8a888e5b1026878e4e19398b23bd38ec221a'. A note below says 'Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, wCryptV3.1BackupDefaults'. To the right, there's a user profile for 'metens' with a timestamp 'July 17, 2025 at 2:34 PM'. Below the input field is a table with one row:

Hash	Type	Result
e7cf3ef4f17c3999a94f2c6f612e8a888e5b1026878e4e19398b23bd38ec221a	sha256	Password

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

The most important properties of cryptographic hash functions are their pre-image resistance, collision resistance, and their ability to act as pseudo-random functions. In the case of <https://crackstation.net/>, the property being attacked is the pre-image resistance. The site stores passwords and their hashes, making it easy to find the output generated from the input. "CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash."

0d107d09f5bbe40cade3de5c71e9e9b7



I'm not a robot


 reCAPTCHA
[Privacy](#) - [Terms](#)

[Crack Hashes](#)

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
0d107d09f5bbe40cade3de5c71e9e9b7	md5	letmein

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

cbfdac6008f9cab4083784cbd1874f76618d2a97



I'm not a robot


 reCAPTCHA
[Privacy](#) - [Terms](#)

[Crack Hashes](#)

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
cbfdac6008f9cab4083784cbd1874f76618d2a97	sha1	password123

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

65e84be33532fb784c48129675f9eff3a682b27168c0ea744b2cf58ee023
37c5



I'm not a robot


 reCAPTCHA
[Privacy](#) - [Terms](#)

[Crack Hashes](#)

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
65e84be33532fb784c48129675f9eff3a682b27168c0ea744b2cf58ee02337c5	sha256	qwerty

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Task 3: Salt & Stretch

```
metens@cs591-metens-kali-1:~/Lab-3.1
File Actions Edit View Help

[metens@cs591-metens-kali-1] ~[~/Lab-3.1]
$ touch password.txt

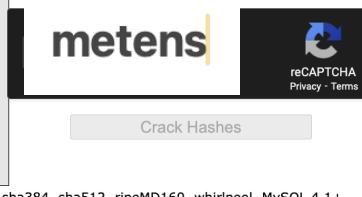
[metens@cs591-metens-kali-1] ~[~/Lab-3.1]
$ echo -n "password" | openssl dgst -sha256 | awk '{print $2}' > password.txt;

[metens@cs591-metens-kali-1] ~[~/Lab-3.1]
$ echo -n "12345:password" | openssl dgst -sha256 | awk '{print $2}' >> password.txt;

[metens@cs591-metens-kali-1] ~[~/Lab-3.1]
$ cat password.txt
5e884898da28047151d0e56f8dc6292773603d0d6aabbd62a11ef721d1542d8
0ded8b22175c4c71c99a957557c0136abbaa95b7ba069bf73118ae509a286cf

[metens@cs591-metens-kali-1] ~[~/Lab-3.1]
$
```

0ded8b22175c4c71c99a957557c0136abbaa95b7ba069bf73118ae509a286cf



Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+(sha1(bin)), QubesV3.1BackupDefaults

Hash	Type	Result
0ded8b22175c4c71c99a957557c0136abbaa95b7ba069bf73118ae509a286cf	Unknown	Not found.

Color Codes: **Green:** Exact match, **Yellow:** Partial match, **Red:** Not found.

```
[metens@cs591-metens-kali-1] ~[~/Lab-3.1]
$ time (echo -n '12345:password' | openssl dgst -sha256); \
> time (echo -n 'password' | openssl enc -aes128 -k 1 -pbkdf2 -iter 31000 | xxd); \
> time (echo -n 'password' | openssl enc -aes128 -k 1 -pbkdf2 -iter 310000 | xxd);
SHA2-256(stdin)= 0ded8b22175c4c71c99a957557c0136abbaa95b7ba069bf73118ae509a286cf

real    0.01s
user    0.01s
sys     0.00s
cpu     97%
00000000: 5361 6c74 6564 5f5f daee 6454 8656 7367  Salted__dT.Vsg
00000010: 287a bfb1 856d c87d a4a3 d7ee 9fe1 84ef  {z ...m}.....
real    0.02s
user    0.01s
sys     0.00s
cpu     83%
00000000: 5361 6c74 6564 5f5f 49cb 1b0e 78df 1feb  Salted__I...x...
00000010: ddac ce22 ef89 85a0 e954 7c7f 022e c91d  ...".....T|.....
real    0.06s
user    0.06s
sys     0.00s
cpu     100%

[metens@cs591-metens-kali-1] ~[~/Lab-3.1]
$
```

It appears as though the PBKDF with 31000 iterations is 2 times slower, and the PBKDF with 310000 iterations is 6 times slower compared to the SHA-256 hash.

Task 4: Hash types

```
Try half with a good dictionary and a bit of knowledge of the command switches.

Hashcat is the self-proclaimed world's fastest CPU-based password recovery
hashcat supported hashing algorithms are Microsoft LM Hashes, MD4, MD5, S
formats, MySQL, Cisco PIX.

OPTIONS
-h, --help
    Show summary of options.

-v, --version
    Show version of program.

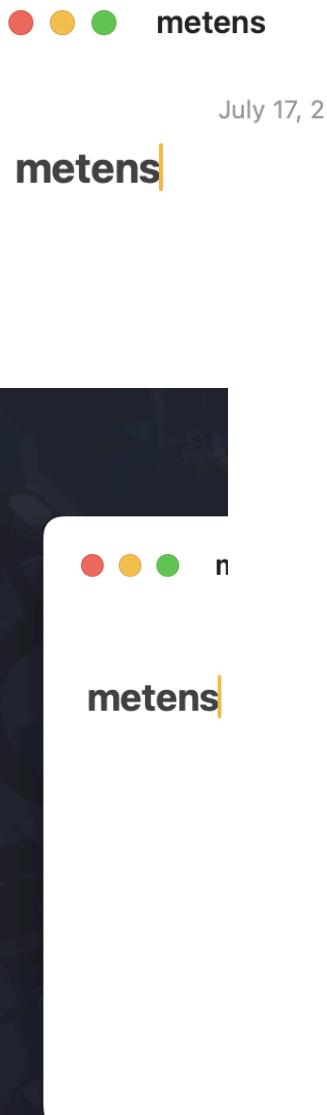
-m, --hash-type=NUM
    Hash-type, see references below

-a, --attack-mode=NUM
    Attack-mode, see references below

--quiet
    Suppress output

--force
    Ignore warnings

0 = MD5
10 = md5($pass.$salt)
20 = md5($salt.$pass)
30 = md5(unicode($pass).$salt)
40 = md5($salt.unicode($pass))
50 = HMAC-MD5 (key = $pass)
60 = HMAC-MD5 (key = $salt)
100 = SHA1
110 = sha1($pass.$salt)
120 = sha1($salt.$pass)
130 = sha1(unicode($pass).$salt)
140 = sha1($salt.unicode($pass))
150 = HMAC-SHA1 (key = $pass)
160 = HMAC-SHA1 (key = $salt)
200 = MySQL323
300 = MySQL4.1/MySQL5
400 = phpass, MD5(Wordpress), MD5/phpBB3, MD5(Joomla)
500 = md5crypt, MD5(Unix), FreeBSD MD5, Cisco-IOS MD5
600 = MD4
1000 = NTLM
1100 = Domain Cached Credentials (DCC), MS Cache
1400 = SHA256
1410 = sha256($pass.$salt)
1420 = sha256($salt.$pass)
1430 = sha256(unicode($pass).$salt)
1431 = base64(sha256(unicode($pass)))
1440 = sha256($salt.unicode($pass))
1450 = HMAC-SHA256 (key = $pass)
1460 = HMAC-SHA256 (key = $salt)
1600 = md5apr1, MD5(APR), Apache MD5
1700 = SHA512
1710 = sha512($pass.$salt)
1720 = sha512($salt.$pass)
1730 = sha512(unicode($pass).$salt)
1740 = sha512($salt.unicode($pass))
1750 = HMAC-SHA512 (key = $pass)
```



Hash Type numbers: MD5(0), HMAC-SHA1(150/160), NTLM(1000), SHA-256(1400), SHA-512(1700).

Task 5: Cracking passwords with hashcat

```
File Actions Edit View Help

[(metens@cs591-metens-kali-1)-[~/Lab-3.1]
$ curl -LO https://web.cecs.pdx.edu/~rchaney/Classes/cs491/Labs/Lab3-1.tar.gz; tar xvfa Lab3-1.tar.gz
% Total    % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload Total Spent   Left Speed
100 1310 100 1310    0     0  5974      0 --::-- --::-- --::--  5981
crack-hashes.sh
create-hashes.sh

[(metens@cs591-metens-kali-1)-[~/Lab-3.1]
$ ls -al
OWNED by metens
total 24
drwxrwxr-x  2 metens metens 4096 Jul 17 15:29 .
drwxr-xr-x 19 metens metens 4096 Jul 17 15:29 ..
-rwxr-xr-x  1 metens metens 1280 Jul 15 19:15 crack-hashes.sh
-rwxr-xr-x  1 metens metens 1601 Jul 15 20:24 create-hashes.sh
-rw-rw-r--  1 metens metens 1310 Jul 17 15:29 Lab3-1.tar.gz
-rw-rw-r--  1 metens metens 130  Jul 17 14:48 password.txt

[(metens@cs591-metens-kali-1)-[~/Lab-3.1]
$ ./create-hashes.sh
Creating 10 passwords with nt
Creating 10 passwords with decrypt
Creating 10 passwords with md5
Creating 10 passwords with sha-256
Creating 10 passwords with sha-512
Creating 10 passwords with bcrypt

[(metens@cs591-metens-kali-1)-[~/Lab-3.1]
$ ls -al
OWNED by metens
total 244
drwxrwxr-x  2 metens metens 4096 Jul 17 15:30 .
drwxr-xr-x 19 metens metens 4096 Jul 17 15:29 ..
-rwxr-xr-x  1 metens metens 1280 Jul 15 19:15 crack-hashes.sh
-rwxr-xr-x  1 metens metens 1601 Jul 15 20:24 create-hashes.sh
-rw-rw-r--  1 metens metens 1310 Jul 17 15:29 Lab3-1.tar.gz
-rw-rw-r--  1 metens metens 610  Jul 17 15:30 password.bcrypt
-rw-rw-r--  1 metens metens 140  Jul 17 15:30 password.decrypt
-rw-rw-r--  1 metens metens 350  Jul 17 15:30 password.md5
-rw-rw-r--  1 metens metens 330  Jul 17 15:30 password.nt
-rw-rw-r--  1 metens metens 640  Jul 17 15:30 password.sha-256
-rw-rw-r--  1 metens metens 1070 Jul 17 15:30 password.sha-512
-rw-rw-r--  1 metens metens 130  Jul 17 14:48 password.txt
-rw-rw-r--  1 metens metens 83   Jul 17 15:30 plain.txt
```

```
└─(metens㉿cs591-metens-kali-1)─[~/Lab-3.1]
$ ./crack-hashes.sh

Mode: nt
Recovered.....: 10/10 (100.00%) Digests (total), 10/10 (100.00%) Digests (new)
Progress.....: 15360/20010 (76.76%)

real    0m20.715s
user    0m13.093s
sys     0m5.163s

File System
Mode: descript
Recovered.....: 10/10 (100.00%) Digests (total), 10/10 (100.00%) Digests (new), 10/10 (100.00%) Salts
Progress.....: 150784/200100 (75.35%)

real    0m12.654s
user    0m10.482s
sys     0m0.870s

Tech
Mode: md5
Recovered.....: 10/10 (100.00%) Digests (total), 10/10 (100.00%) Digests (new), 10/10 (100.00%) Salts
Progress.....: 150016/200100 (74.97%)

real    0m11.498s
user    0m16.816s
sys     0m0.598s

Mode: sha-256
Recovered.....: 10/10 (100.00%) Digests (total), 10/10 (100.00%) Digests (new), 10/10 (100.00%) Salts
Progress.....: 149760/200100 (74.84%)

real    1m41.525s
user    2m51.186s
sys     0m1.304s

Mode: sha-512
Recovered.....: 10/10 (100.00%) Digests (total), 10/10 (100.00%) Digests (new), 10/10 (100.00%) Salts
Progress.....: 150784/200100 (75.35%)

real    1m39.164s
user    2m40.456s
sys     0m1.358s

Mode: bcrypt
Recovered.....: 10/10 (100.00%) Digests (total), 10/10 (100.00%) Digests (new), 10/10 (100.00%) Salts
Progress.....: 149986/200100 (74.96%)

real    2m14.062s
user    3m20.016s
sys     0m1.849s

└─(metens㉿cs591-metens-kali-1)─[~/Lab-3.1]
$ █
```

The quickest algorithm was MD5 and took about 11 seconds, while the slowest algorithm was BCRYPT which took nearly 2 minutes and 14 seconds.

Task 6: Cracking SSH passphrases

```
(metens㉿cs591-metens-kali-1) [~/Lab-3.1]
$ PASSWORD=$(head -n 500 wordlist.txt | tail -n 1); echo ${PASSWORD}
Derrikus5
```

```
$ ssh-keygen -t rsa -f ./id_rsa -b 1024 -N ${PASSWORD}
Generating public/private rsa key pair.
Your identification has been saved in ./id_rsa
Your public key has been saved in ./id_rsa.pub
The key fingerprint is:
SHA256:kNWEGMCgUyhtNjj1C1IwymGbbs+K+crrWjKfhXBRMk metens@cs591-metens-kali-1
The key's randomart image is:
+---[RSA 1024]---+
|o*+=++oo.+
|BB*+.E.o. .
|B+=.o =
| +...o .
| .. . S
| . .
| o .
|=.=.
|B0=+
+---[SHA256]---+
```

```
(metens㉿cs591-metens-kali-1) [~/Lab-3.1]
$ ls
OWNED by metens
cracked.bcrypt    cracked.nt      crack-hashes.sh   id_rsa.pub      password.decrypt
cracked.decrypt  cracked.sha-256  create-hashes.sh Lab3-1.tar.gz  password.md5
cracked.md5       cracked.sha-512  id_rsa          password.bcrypt  password.nt
```

```
(metens㉿cs591-metens-kali-1) [~/Lab-3.1]
$ ls | grep "id_rsa"
id_rsa
id_rsa.pub
```

```
[metens@cs591-metens-kali-1]~[Lab-3.1]
$ python3 /usr/share/john/ssh2john.py ./id_rsa > id_rsa.hash

[metens@cs591-metens-kali-1]~[Lab-3.1]
$ cat id_rsa.hash
./id_rsa:$sshng6$16$81422b56c3678685f45ae2ffdb539896$758$6f70656e7373682d6b65792d7631000000000a6165733235362d637472000000066
26372797074000000180000001081422b56c3678685f45ae2ffdb53989600000018000000010000000070000000077373682d727361000000003010001000000
8100c6f117cb1d78d9e7169f1e2536b90f6ff249aed97051614c983eef5f65f36fe04e5e4a1819189cb5213e4377f5c4bd9441577490d1e438503ea52bbb9
51399b14b8331018f067903b864a5b54c6a7788a47b1600b309dc1e9902bf7f895690f9cd12822ec6016bd896bc03aa9af5f0d7b6f2106d036d413809217d
e339ca275b000021098186751e634d6d981d368dfa5482af34922f90ccf20ffc19608cdfcea50f63cc8aeac854b761802ade51be33759e3b450bd926c1c
dc202828d3086b3f76837d82f7a79f06b69fb4c20144f6adbc79d1d00d9ce0caeab677f8fb2ba93767e7ac8775044157372fb523f6e454b4f0c0169e73b
13c1dcfff0c356d953304e5f0b0c1ae87d15e03295c7655b10f19c163cc43063a3112ef03c747b83216dc22bd5ffa2057dfbd35f37eaaee6833c7b0437e22a6
947597c63029b6a9d653199f14e25d8d33be911352f36f36759e1cadfa995544055128420b35151cb757f8a96f6c648e9fe7c36f666e6de67622f70bc0b5
b4d2c87f094be4920e239ab5b1f892b5e900c5f961a162608f81f9fba8eb4da59f122d5753ba6f6f7225cc06ff9d486e057787f51799ad6ef2ee42c1f18366
1a61f5c665aeaf1a762c16959f573cccd16790a4530571784d0110253ca1312d0e4e5b869cd85c4befb7f8555915158537b1343eccf7fb88959ec1766ff96
68a478d178585d7d97778f19b7f31554cd9e28f2bclaebeafab1cbd55a299a611621d3d3cdadde0c8057b3ee0fea5b3c33c8d08b8f6458574579fab0558f0
11d20d7676d889dfdd8b2c55cb6e6fafed7a5b87450a0c83a6900184fa33d4a1fc3d7f870a1163ff96401373685c7e5454c1cc3b457d991d83900c1329f72
199487bbbe57606a9ba288ebb865206bdeb0dd21a3927633ee21603b0dfaefca4037f1f06$24$230

[metens@cs591-metens-kali-1]~[Lab-3.1]
$ time john --wordlist=wordlist.txt id_rsa.hash
Created directory: /home/metens/.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 24 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:09 0.39% (ETA: 17:01:45) 0g/s 6.680p/s 6.680c/s 6.680C/s burri05..08998474260
0g 0:00:00:11 0.45% (ETA: 17:03:18) 0g/s 6.672p/s 6.672c/s 6.672C/s mar1103..romule93
0g 0:00:00:19 0.70% (ETA: 17:08:13) 0g/s 6.659p/s 6.659c/s 6.659C/s geonik..5buckshot
0g 0:00:00:53 1.80% (ETA: 17:11:55) 0g/s 6.620p/s 6.620c/s 6.620C/s chloe_1996..bazzle85
Derrikus5 (.id_rsa)
1g 0:00:01:17 DONE (2025-07-17 16:24) 0.01291g/s 6.614p/s 6.614c/s 6.614C/s hate boys for ever..rachelle!
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

real    83.29s
user    151.36s
sys     0.36s
cpu     182%
```