Hashcat的使用手册总结

By七友 / 2019-02-13 22:55:00 / 浏览数 8062 安全工具 工具 顶(3) 踩(0)

简介

Hashcat是自称世界上最快的密码恢复工具。它在2015年之前拥有专有代码库，但现在作为免费软件发布。适用于Linux，OS X和Windows的版本可以使用基于CPU或基于GPU的变体。支持hashcat的散列算法有Microsoft LM哈希，MD4，MD5，SHA系列，Unix加密格式，MySQL和Cisco PIX等。

hashcat支持多种计算核心：

```
GPU
CPU
APU
DSP
FPGA
Coprocessor
```

GPU的驱动要求

```
AMD GPUs on Linux require "RadeonOpenCompute (ROCm)" Software Platform (1.6.180 or later)
AMD GPUs on Windows require "AMD Radeon Software Crimson Edition" (15.12 or later)
Intel CPUs require "OpenCL Runtime for Intel Core and Intel Xeon Processors" (16.1.1 or later)
Intel GPUs on Linux require "OpenCL 2.0 GPU Driver Package for Linux" (2.0 or later)
Intel GPUs on Windows require "OpenCL Driver for Intel Iris and Intel HD Graphics"
NVIDIA GPUs require "NVIDIA Driver" (367.x or later)
```

最新版hashcat下载地址：https://hashcat.net/files/hashcat-5.1.0.7z
GitHub地址：https://github.com/hashcat/hashcat

参数

下面使常见的参数，想了解更多的参数可以hashcat --help查看

```
-a  ■■■■■■■■■■■■■■■■■■■■■■“-a 0”■■■■■■“-a 1” ■■■■■■“-a 3”■■■■■■
-m  ■■■■■■■hash■■■■■■■■■■■■■■■■■MD5
-o  ■■■■■■■■■hash■■■■■■■■■■■■■■■■,■■■■■■■■■■■hash■■■■■■■■■
--force ■■■■■■■■■■■■■,■■■hash■■■■■■■■■■
--show  ■■■■■■■■hash■■hash■■■■■■■
--increment  ■■■■■■■■■,■■■■■■■■■■■hashcat■■■■■■■■■■■■■■■■■■
--increment-min  ■■■■■■■,■■■■■■■■■■■■■■,■■increment■■■■■■■
--increment-max  ■■■■■■■,■■
--outfile-format ■■■■■■■■■■■■■id,■■■3
--username   ■■hash■■■■■■■■■■■■,■■■linux■■■■■■■hash■■■■■■
--remove     ■■■■■■■■■■hash
-r       ■■■■■■■■■■
```

攻击模式：

```
# | Mode
===+======
 0 | Straight■■■■■■■
 1 | Combination■■■■■■■
 3 | Brute-force■■■■■■■■■
 6 | Hybrid Wordlist + Mask■■■■+■■■■■■
 7 | Hybrid Mask + Wordlist■■■■+■■■■■■
```

输出格式

```
1 = hash[:salt]
2 = plain
3 = hash[:salt]:plain
4 = hex_plain
5 = hash[:salt]:hex_plain
6 = plain:hex_plain
```

```
7 = hash[:salt]:plain:hex_plain
8 = crackpos
9 = hash[:salt]:crackpos
10 = plain:crackpos
11 = hash[:salt]:plain:crackpos
12 = hex_plain:crackpos
13 = hash[:salt]:hex_plain:crackpos
14 = plain:hex_plain:crackpos
15 = hash[:salt]:plain:hex_plain:crackpos
```

Hash id对照表

因为实在是太多了，所有我就贴一部分常见的hash类型，要想了解所有的参数可到hashcat的Wiki上去看，或者直接hashcat --help查看hash对照表

```
- [ Hash modes ] -

     # | Name                                               | Category
 ======+====================================================+======================================
   900 | MD4                                                | Raw Hash
     0 | MD5                                                | Raw Hash
  5100 | Half MD5                                           | Raw Hash
   100 | SHA1                                               | Raw Hash
  1300 | SHA2-224                                           | Raw Hash
  1400 | SHA2-256                                           | Raw Hash
 10800 | SHA2-384                                           | Raw Hash
  1700 | SHA2-512                                           | Raw Hash
 17300 | SHA3-224                                           | Raw Hash
 17400 | SHA3-256                                           | Raw Hash
 17500 | SHA3-384                                           | Raw Hash
 17600 | SHA3-512                                           | Raw Hash
    10 | md5($pass.$salt)                                   | Raw Hash, Salted and/or Iterated
    20 | md5($salt.$pass)                                   | Raw Hash, Salted and/or Iterated
    30 | md5(utf16le($pass).$salt)                          | Raw Hash, Salted and/or Iterated
    40 | md5($salt.utf16le($pass))                          | Raw Hash, Salted and/or Iterated
  3800 | md5($salt.$pass.$salt)                             | Raw Hash, Salted and/or Iterated
  3710 | md5($salt.md5($pass))                              | Raw Hash, Salted and/or Iterated
  4010 | md5($salt.md5($salt.$pass))                        | Raw Hash, Salted and/or Iterated
  4110 | md5($salt.md5($pass.$salt))                        | Raw Hash, Salted and/or Iterated
  2600 | md5(md5($pass))                                    | Raw Hash, Salted and/or Iterated
  3910 | md5(md5($pass).md5($salt))                         | Raw Hash, Salted and/or Iterated
  4300 | md5(strtoupper(md5($pass)))                        | Raw Hash, Salted and/or Iterated
  4400 | md5(sha1($pass))                                   | Raw Hash, Salted and/or Iterated
   110 | sha1($pass.$salt)                                  | Raw Hash, Salted and/or Iterated
   120 | sha1($salt.$pass)                                  | Raw Hash, Salted and/or Iterated
   130 | sha1(utf16le($pass).$salt)                         | Raw Hash, Salted and/or Iterated
   140 | sha1($salt.utf16le($pass))                         | Raw Hash, Salted and/or Iterated
  4500 | sha1(sha1($pass))                                  | Raw Hash, Salted and/or Iterated
  4520 | sha1($salt.sha1($pass))                            | Raw Hash, Salted and/or Iterated
  4700 | sha1(md5($pass))                                   | Raw Hash, Salted and/or Iterated
  4900 | sha1($salt.$pass.$salt)                            | Raw Hash, Salted and/or Iterated
 14400 | sha1(CX)                                           | Raw Hash, Salted and/or Iterated
  1410 | sha256($pass.$salt)                                | Raw Hash, Salted and/or Iterated
  1420 | sha256($salt.$pass)                                | Raw Hash, Salted and/or Iterated
  1430 | sha256(utf16le($pass).$salt)                       | Raw Hash, Salted and/or Iterated
  1440 | sha256($salt.utf16le($pass))                       | Raw Hash, Salted and/or Iterated
  1710 | sha512($pass.$salt)                                | Raw Hash, Salted and/or Iterated
  1720 | sha512($salt.$pass)                                | Raw Hash, Salted and/or Iterated
  1730 | sha512(utf16le($pass).$salt)                       | Raw Hash, Salted and/or Iterated
  1740 | sha512($salt.utf16le($pass))                       | Raw Hash, Salted and/or Iterated
 14000 | DES (PT = $salt, key = $pass)                      | Raw Cipher, Known-Plaintext attack
 14100 | 3DES (PT = $salt, key = $pass)                     | Raw Cipher, Known-Plaintext attack
 14900 | Skip32 (PT = $salt, key = $pass)                   | Raw Cipher, Known-Plaintext attack
 15400 | ChaCha20                                           | Raw Cipher, Known-Plaintext attack
  2500 | WPA-EAPOL-PBKDF2                                   | Network Protocols
  2501 | WPA-EAPOL-PMK                                      | Network Protocols
 16800 | WPA-PMKID-PBKDF2                                   | Network Protocols
 16801 | WPA-PMKID-PMK                                      | Network Protocols
  7300 | IPMI2 RAKP HMAC-SHA1                               | Network Protocols
  7500 | Kerberos 5 AS-REQ Pre-Auth etype 23                | Network Protocols
```

```
 8300 | DNSSEC (NSEC3)                                 | Network Protocols
10200 | CRAM-MD5                                       | Network Protocols
11100 | PostgreSQL CRAM (MD5)                          | Network Protocols
11200 | MySQL CRAM (SHA1)                              | Network Protocols
16100 | TACACS+                                        | Network Protocols
16500 | JWT (JSON Web Token)                           | Network Protocols
  121 | SMF (Simple Machines Forum) > v1.1             | Forums, CMS, E-Commerce, Frameworks
  400 | phpBB3 (MD5)                                   | Forums, CMS, E-Commerce, Frameworks
 2811 | MyBB 1.2+                                      | Forums, CMS, E-Commerce, Frameworks
 2811 | IPB2+ (Invision Power Board)                   | Forums, CMS, E-Commerce, Frameworks
 8400 | WBB3 (Woltlab Burning Board)                   | Forums, CMS, E-Commerce, Frameworks
   11 | Joomla < 2.5.18                                | Forums, CMS, E-Commerce, Frameworks
  400 | Joomla >= 2.5.18 (MD5)                         | Forums, CMS, E-Commerce, Frameworks
  400 | WordPress (MD5)                                | Forums, CMS, E-Commerce, Frameworks
 2612 | PHPS                                           | Forums, CMS, E-Commerce, Frameworks
 7900 | Drupal7                                        | Forums, CMS, E-Commerce, Frameworks
   21 | osCommerce                                     | Forums, CMS, E-Commerce, Frameworks
   21 | xt:Commerce                                    | Forums, CMS, E-Commerce, Frameworks
11000 | PrestaShop                                     | Forums, CMS, E-Commerce, Frameworks
  124 | Django (SHA-1)                                 | Forums, CMS, E-Commerce, Frameworks
10000 | Django (PBKDF2-SHA256)                         | Forums, CMS, E-Commerce, Frameworks
   12 | PostgreSQL                                     | Database Server
  131 | MSSQL (2000)                                   | Database Server
  132 | MSSQL (2005)                                   | Database Server
 1731 | MSSQL (2012, 2014)                             | Database Server
  200 | MySQL323                                       | Database Server
  300 | MySQL4.1/MySQL5                                | Database Server
 3100 | Oracle H: Type (Oracle 7+)                     | Database Server
  112 | Oracle S: Type (Oracle 11+)                    | Database Server
12300 | Oracle T: Type (Oracle 12+)                    | Database Server
 8000 | Sybase ASE                                     | Database Server
15000 | FileZilla Server >= 0.9.55                     | FTP Server
11500 | CRC32                                          | Checksums
 3000 | LM                                             | Operating Systems
 1000 | NTLM                                           | Operating Systems
  500 | md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)      | Operating Systems
 3200 | bcrypt $2*$, Blowfish (Unix)                   | Operating Systems
 7400 | sha256crypt $5$, SHA256 (Unix)                 | Operating Systems
 1800 | sha512crypt $6$, SHA512 (Unix)                 | Operating Systems
  122 | macOS v10.4, MacOS v10.5, MacOS v10.6          | Operating Systems
 1722 | macOS v10.7                                    | Operating Systems
 7100 | macOS v10.8+ (PBKDF2-SHA512)                   | Operating Systems
11600 | 7-Zip                                          | Archives
12500 | RAR3-hp                                        | Archives
13000 | RAR5                                           | Archives
13600 | WinZip                                         | Archives
 9700 | MS Office <= 2003 $0/$1, MD5 + RC4             | Documents
 9710 | MS Office <= 2003 $0/$1, MD5 + RC4, collider #1| Documents
 9720 | MS Office <= 2003 $0/$1, MD5 + RC4, collider #2| Documents
 9800 | MS Office <= 2003 $3/$4, SHA1 + RC4            | Documents
 9810 | MS Office <= 2003 $3, SHA1 + RC4, collider #1  | Documents
 9820 | MS Office <= 2003 $3, SHA1 + RC4, collider #2  | Documents
 9400 | MS Office 2007                                 | Documents
 9500 | MS Office 2010                                 | Documents
 9600 | MS Office 2013                                 | Documents
10400 | PDF 1.1 - 1.3 (Acrobat 2 - 4)                  | Documents
10410 | PDF 1.1 - 1.3 (Acrobat 2 - 4), collider #1     | Documents
10420 | PDF 1.1 - 1.3 (Acrobat 2 - 4), collider #2     | Documents
10500 | PDF 1.4 - 1.6 (Acrobat 5 - 8)                  | Documents
10600 | PDF 1.7 Level 3 (Acrobat 9)                    | Documents
10700 | PDF 1.7 Level 8 (Acrobat 10 - 11)              | Documents
99999 | Plaintext                                      | Plaintext
```

掩码设置

这里列一下常见的掩码字符集

```
l | abcdefghijklmnopqrstuvwxyz          ■■■■■
u | ABCDEFGHIJKLMNOPQRSTUVWXYZ          ■■■■■
```

```
d | 0123456789                  ■■■
h | 0123456789abcdef                ■■■■■■■■■■■
H | 0123456789ABCDEF                ■■■■■■■■■■
s |  !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~       ■■■■
a | ?l?u?d?s                     ■■■■■■■■■■■
b | 0x00 - 0xff                  ■■■■■■■■■■■■■■■■
```

下面举几个简单的例子来了解一下掩码的设置

```
■■■■■■■■?d?d?d?d?d?d?d?d
■■■■■■■■?a?a?a?a?a?a?a?a
■■■■■■■■■■■■■■■■■■■?u?u?u?u?d?d?d?d
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■?h?h?h?h?H?H?H?H
■■■■■■■■■■■■admin■■■■■■■■?a?a?aadmin?a?a?a
6-8■■■■■■■--increment --increment-min 6 --increment-max 8 ?l?l?l?l?l?l?l?l
6-8■■■■+■■■■■■■■--increment --increment-min 6 --increment-max 8 ?h?h?h?h?h?h?h?h
```

如果我们想设置字符集为：abcd123456!@-+，那该怎么做呢。这就需要用到自定义字符集这个参数了，hashcat支持用户最多定义4组字符集

```
--custom-charset1 [chars]■■■■ -1
--custom-charset2 [chars]■■■■ -2
--custom-charset3 [chars]■■■■ -3
--custom-charset4 [chars]■■■■ -4
■■■■■■?1■?2■?3■?4■■■■■
```

再来举几个例子：

```
--custom-charset1 abcd123456!@-+■■■■■■■■■■"?1"■■■■■■■■■■
--custom-charset2 ?l?d■■■■■?2■■■■■?h
-1 ?d?l?u■?1■■■■■■+■■■■■+■■■■■
-3 abcdef -4 123456 ■■?3?3?3?3?4?4?4?4■■■■■■■■■■■“abcdef”■■■■■■■■“123456”
```

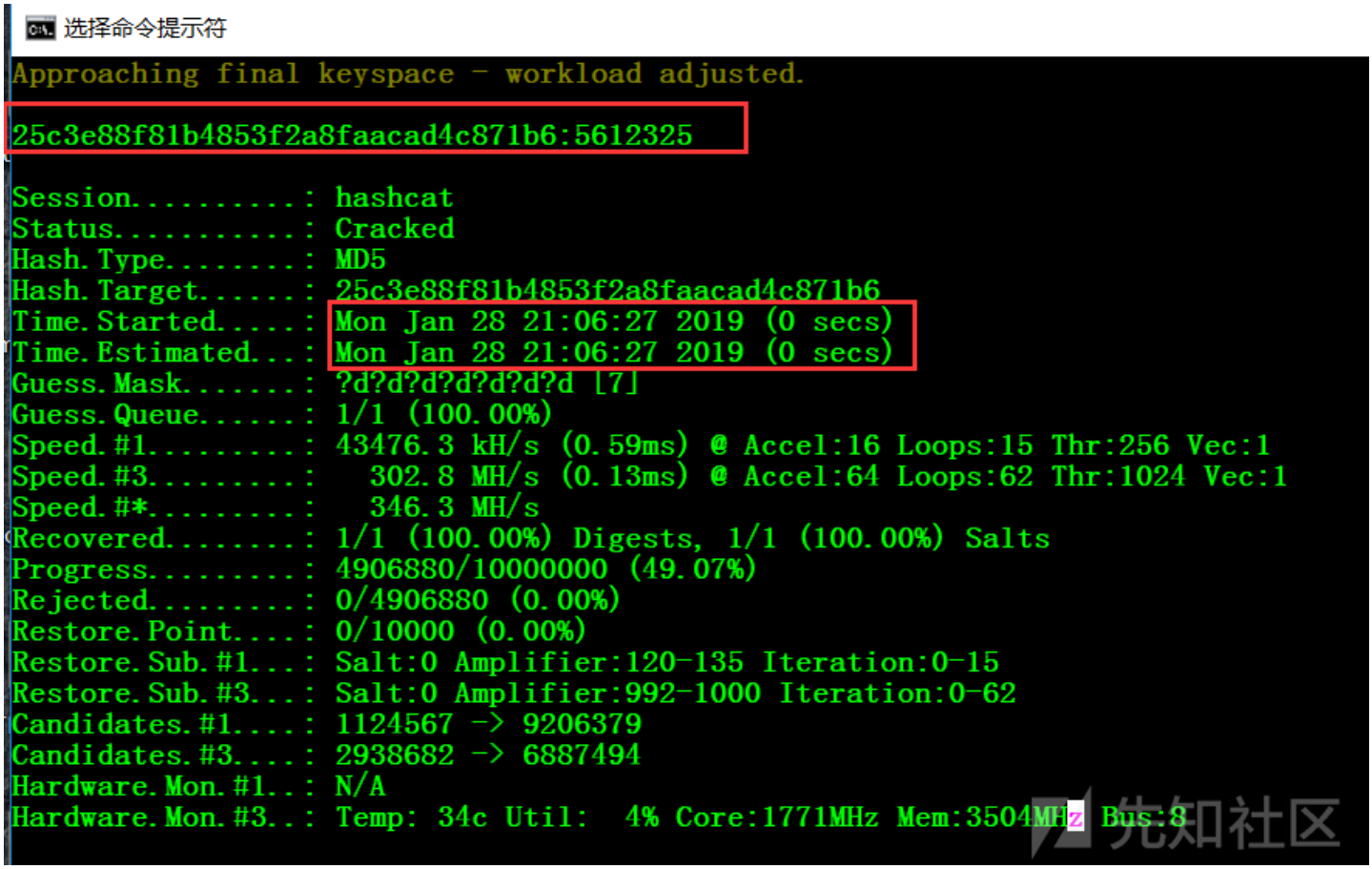
例子

PS：我这里给一下我机子的配置，然后再对比一下破解的速度

```
CPU■Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz
■■■GTX 1050 Ti
```

7位数字破解

```
hashcat64.exe -a 3 -m 0 --force 25c3e88f81b4853f2a8faacad4c871b6 ?d?d?d?d?d?d?d
```



7位小写字母破解：

```
hashcat64.exe -a 3 -m 0 --force 7a47c6db227df60a6d67245d7d8063f3 ?l?l?l?l?l?l?l
```

1-8位数字破解：

```
hashcat64.exe -a 3 -m 0 --force 4488cec2aea535179e085367d8a17d75 --increment --increment-min 1 --increment-max 8 ?d?d?d?d?d?d?
```

1-8位小写字母+数字破解

```
hashcat64.exe -a 3 -m 0 --force ab65d749cba1656ca11dfa1cc2383102 --increment --increment-min 1 --increment-max 8 ?h?h?h?h?h?h?
```

特定字符集：123456abcdf!@+-

```
hashcat64.exe -a 3 -1 123456abcdf!@+- 8b78ba5089b11326290bc15cf0b9a07d ?1?1?1?1?1
[redacted]
```

1-8为位符集:123456abcdf!@+-

```
hashcat64.exe -a 3 -1 123456abcdf!@+- 9054fa315ce16f7f0955b4af06d1aa1b --increment --increment-min 1 --increment-max 8 ?1?1?1?
```

1-8位数字+大小写字母+可见特殊符号

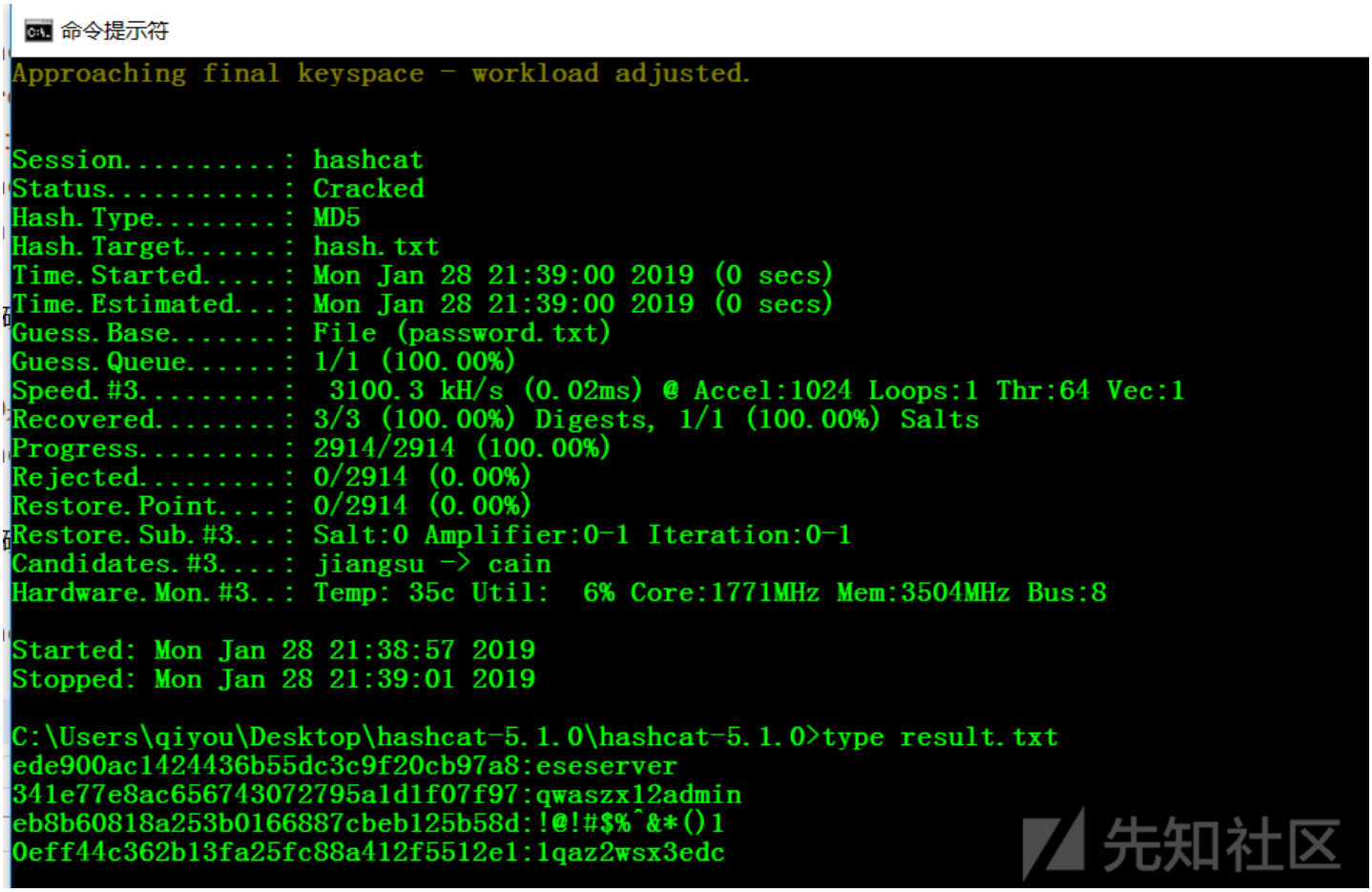
```
hashcat64.exe -a 3 -1 ?d?u?l?s d37fc9ee39dd45a7717e3e3e9415f65d --increment --increment-min 1 --increment-max 8 ?1?1?1?1?1?1?1
[redacted]
hashcat64.exe -a 3 d37fc9ee39dd45a7717e3e3e9415f65d --increment --increment-min 1 --increment-max 8 ?a?a?a?a?a?a?a?a
```

字典破解

```
-a 0[redacted]-o[redacted]
hashcat64.exe -a 0 ede900ac1424436b55dc3c9f20cb97a8 password.txt -o result.txt
```

批量破解

```
hashcat64.exe -a 0 hash.txt password.txt -o result.txt
```



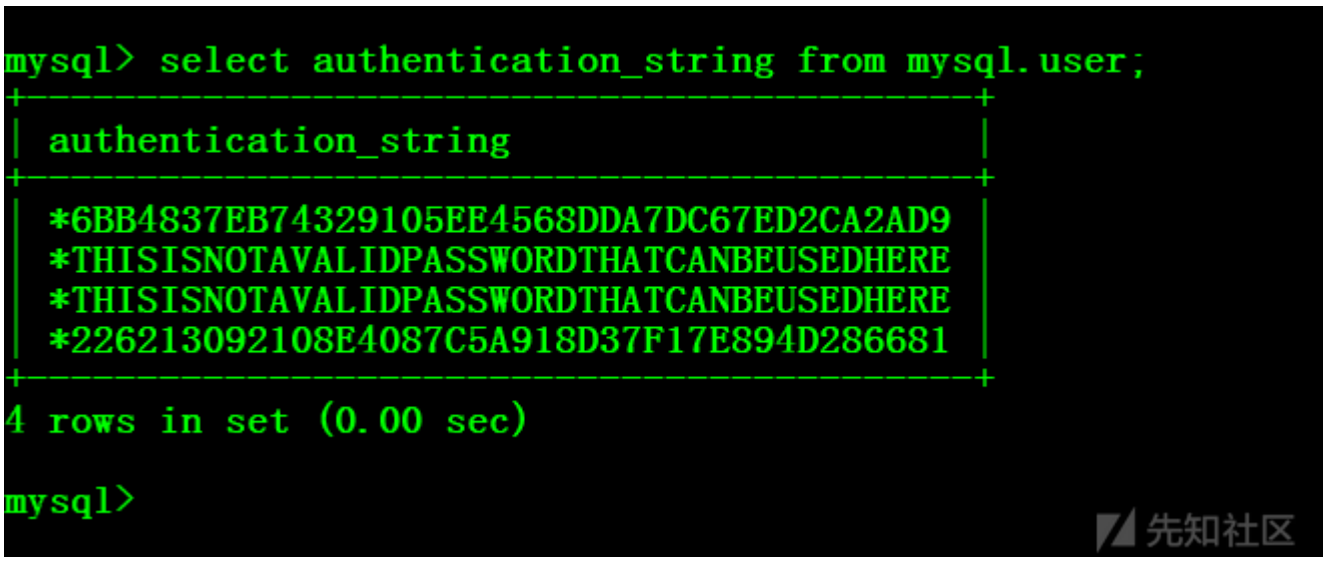
字典组合破解：

```
hashcat64.exe -a 1 25f9e794323b453885f5181f1b624d0b pwd1.txt pwd2.txt
```

字典+掩码破解

```
hashcat64.exe -a 6 9dc9d5ed5031367d42543763423c24ee password.txt ?l?l?l?l?l
```

Mysql4.1/5的PASSWORD函数




```
hashcat64.exe -a 3 -m 300 --force 6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9 ?d?d?d?d?d?d
```

sha512crypt 6, SHA512 (Unix)破解

可以cat /etc/shadow获取

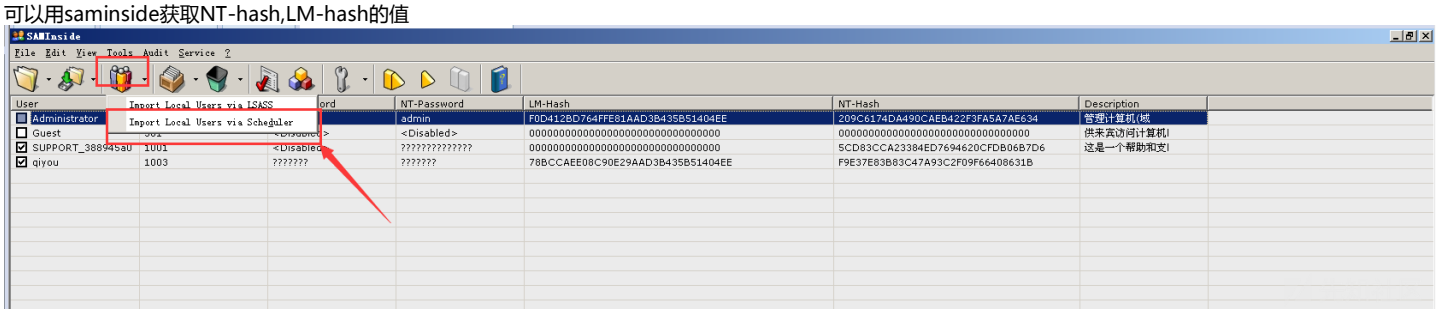
```
hashcat64.exe -a 3 -m 1800 --force $6$mxuA5cdy$XZRk0CvnPFqOgVopqiPEFAFK72SogKVwwwp7gWaUOb7b6tVwfCpcSUsCEk64ktLLYmzyew/xd0O0hPG
```

不用整理用户名，使用--username

```
hashcat64.exe -a 3 -m 1800 --force qiyou:$6$QDq75ki3$jsKm7qTDHz/xBob0kF1Lp170Cgg0i5Tslf3JW/sm9k9Q916mBTyilU3PoOsbRdxV8TAmzvdgN
```

Windows NT-hash，LM-hash破解

可以用saminside获取NT-hash,LM-hash的值




```
NT-hash:
hashcat64.exe -a 3 -m 1000 209C6174DA490CAEB422F3FA5A7AE634 ?l?l?l?l?l
LM-hash:
hashcat64.exe -a 3 -m 3000 F0D412BD764FFE81AAD3B435B51404EE ?l?l?l?l?l
```

mssql

```
hashcat64.exe -a 3 -m 132 --force 0x01008c8006c224f71f6bf0036f78d863c3c4ff53f8c3c48edafb ?l?l?l?l?l?d?d?d
```

wordpress密码hash破解

具体加密脚本在./wp-includes/class-phpass.php的HashPassword函数

```
hashcat64.exe -a 3 -m 400 --force $P$BYEYcHEj3vDhV1lwGBv6rpxurKOEWY/ ?d?d?d?d?d?d
```

discuz用户密码hash破解

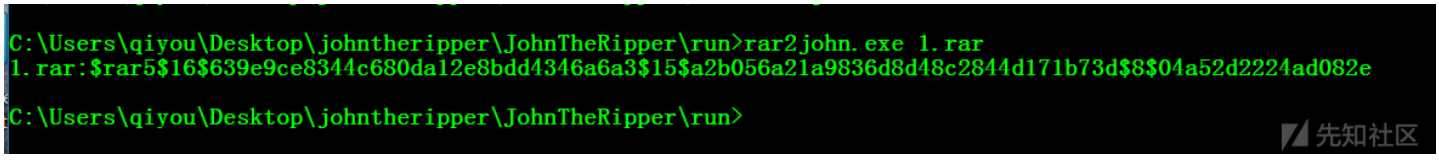
其密码加密方式md5(md5($pass).$salt)

```
hashcat64.exe -a 3 -m 2611 --force 14e1b600b1fd579f47433b88e8d85291: ?d?d?d?d?d?d
```

破解RAR压缩密码

首先rar2john获取rar文件hash值[下载地址]

```
■■rar■■■hash■■rar2john.exe 1.rar
■■■
1.rar:$rar5$16$639e9ce8344c680da12e8bdd4346a6a3$15$a2b056a21a9836d8d48c2844d171b73d$8$04a52d2224ad082e
```




```
hashcat64.exe -a 3 -m 13000 --force $rar5$16$639e9ce8344c680da12e8bdd4346a6a3$15$a2b056a21a9836d8d48c2844d171b73d$8$04a52d2224
```

命令提示符

```
$rar5$16$639e9ce8344c680da12e8bdd4346a6a3$15$a2b056a21a9836d8d48c2844d171b73d$8$04a52d2224ad082e:123456

Session..........: hashcat
Status...........: Cracked
Hash.Type........: RAR5
Hash.Target......: $rar5$16$639e9ce8344c680da12e8bdd4346a6a3$15$a2b056...ad082e
Time.Started.....: Tue Jan 29 16:27:22 2019 (1 min, 18 secs)
Time.Estimated...: Tue Jan 29 16:28:40 2019 (0 secs)
Guess.Mask.......: ?d?d?d?d?d?d [6]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:      606 H/s (8.24ms) @ Accel:8 Loops:4 Thr:256 Vec:1
Speed.#3.........:     7511 H/s (0.76ms) @ Accel:32 Loops:16 Thr:640 Vec:1
Speed.#*.........:     8117 H/s
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 576064/1000000 (57.61%)
Rejected.........: 0/576064 (0.00%)
Restore.Point....: 0/100000 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:32796-32799
Restore.Sub.#3...: Salt:0 Amplifier:9-10 Iteration:32784-32799
Candidates.#1....: 123456 -> 191151
Candidates.#3....: 667537 -> 676464
Hardware.Mon.#1..: N/A
Hardware.Mon.#3..: Temp: 50c Util:  9% Core:1493MHz Mem:3504MHz Bus:8

Started: Tue Jan 29 16:26:51 2019
Stopped: Tue Jan 29 16:28:42 2019
```


注意：

hashcat ■■ RAR3-hp ■ RAR5■■■■■■■■

-m■■	■■	■■ hash
12500	RAR3-hp	$RAR3$*0*45109af8ab5f297a*adbf6c5385d7a40373e8f77d7b89d317
13000	RAR5	$rar5$16$74575567518807622265582327032280$15$f8b4064de34ac02ecabfe

zip密码破解

■zip2john■■■■■hash■■zip2john.exe 1.zip
■■■1.zip:$zip2$*0*3*0*554bb43ff71cb0cac76326f292119dfd*ff23*5*24b28885ee*d4fe362bb1e91319ab53*$/zip2$:::::1.zip-1.txt

```
C:\Users\qiyou\Desktop\johntheripper\JohnTheRipper\run>zip2john.exe 1.zip
1.zip:$zip2$*0*3*0*554bb43ff71cb0cac76326f292119dfd*ff23*5*24b28885ee*d4fe362bb1e91319ab53*$/zip2$:::::1.zip-1.txt
```



```
hashcat64.exe -a 3 -m 13600 $zip2$*0*3*0*554bb43ff71cb0cac76326f292119dfd*ff23*5*24b28885ee*d4fe362bb1e91319ab53*$/zip2$ --for
```

命令提示符

```
Approaching final keyspace - workload adjusted.

$zip2$*0*3*0*554bb43ff71cb0cac76326f292119dfd*ff23*5*24b28885ee*d4fe362bb1e91319ab53*$/zip2$:123456

Session..........: hashcat
Status...........: Cracked
Hash.Type........: WinZip
Hash.Target......: $zip2$*0*3*0*554bb43ff71cb0cac76326f292119dfd*ff23*.../zip2$
Time.Started.....: Tue Jan 29 16:45:07 2019 (1 sec)
Time.Estimated...: Tue Jan 29 16:45:08 2019 (0 secs)
Guess.Mask.......: ?d?d?d?d?d?d [6]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:        0 H/s (8.49ms) @ Accel:8 Loops:3 Thr:256 Vec:1
Speed.#3.........:   118.1 kH/s (2.64ms) @ Accel:32 Loops:15 Thr:1024 Vec:1
Speed.#*.........:   118.1 kH/s
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 51064/1000000 (5.11%)
Rejected.........: 0/51064 (0.00%)
Restore.Point....: 0/100000 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-3
Restore.Sub.#3...: Salt:0 Amplifier:0-1 Iteration:990-999
Candidates.#1....: 104314 -> 136173
Candidates.#3....: 123456 -> 196684
Hardware.Mon.#1..: N/A
Hardware.Mon.#3..: Temp: 42c Util:  4% Core:1759MHz Mem:3504MHz Bus:8

Started: Tue Jan 29 16:44:50 2019
Stopped: Tue Jan 29 16:45:09 2019

C:\Users\qiyou\Desktop\hashcat-5.1.0\hashcat-5.1.0>_
```


破解office密码

■■office■hash■■python office2john.py 11.docx
■■■11.docx:$office$*2013*100000*256*16*e4a3eb62e8d3576f861f9eded75e0525*9eeb35f0849a7800d48113440b4bbb9c*577f8d8b2e1c5f60fed7

```
C:\Users\qiyou\Desktop\johntheripper\JohnTheRipper\run>python office2john.py 11.docx
11.docx:$office$*2013*100000*256*16*e4a3eb62e8d3576f861f9eded75e0525*9eeb35f0849a7800d48113440b4bbb9c*577f8d8b2e1c5f60fed76e62327b38d28f25230f6c7dfd66588d9ca8097aabb9
```

```
hashcat64.exe -a 3 -m 9600 $office$*2013*100000*256*16*e4a3eb62e8d3576f861f9eded75e0525*9eeb35f0849a7800d48113440b4bbb9c*577f8
```

```
命令提示符

$office$*2013*100000*256*16*e4a3eb62e8d3576f861f9eded75e0525*9eeb35f0849a7800d48113440b4bbb9c*577f8d8b2e1c5f60fed76e62327b38d28f2523
0f6c7dfd66588d9ca8097aabb9:123456

Session..........: hashcat
Status...........: Cracked
Hash.Type........: MS Office 2013
Hash.Target......: $office$*2013*100000*256*16*e4a3eb62e8d3576f861f9ed...7aabb9
Time.Started.....: Tue Jan 29 16:57:17 2019 (25 secs)
Time.Estimated...: Tue Jan 29 16:57:42 2019 (0 secs)
Guess.Mask.......: ?d?d?d?d?d?d [6]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:       85 H/s (9.09ms) @ Accel:8 Loops:2 Thr:256 Vec:1
Speed.#3.........:     2101 H/s (1.61ms) @ Accel:32 Loops:8 Thr:1024 Vec:1
Speed.#*.........:     2186 H/s
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 51064/1000000 (5.11%)
Rejected.........: 0/51064 (0.00%)
Restore.Point....: 0/100000 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:4376-4378
Restore.Sub.#3...: Salt:0 Amplifier:0-1 Iteration:99992-100000
Candidates.#1....: 104314 -> 136173
Candidates.#3....: 123456 -> 196684
Hardware.Mon.#1..: N/A
Hardware.Mon.#3..: Temp: 55c Util: 88% Core:1733MHz Mem:3504MHz Bus:8

Started: Tue Jan 29 16:56:29 2019
Stopped: Tue Jan 29 16:57:44 2019

C:\Users\qiyou\Desktop\hashcat-5.1.0\hashcat-5.1.0>
```

破解WIFI密码

首先先把我们的握手包转化为hccapx格式，现在最新版的hashcat只支持hccapx格式了，以前的hccap格式已经不支持了

■■■■■■https://hashcat.net/cap2hccapx/

```
hashcat64.exe -a 3 -m 2500 1.hccapx 1391040?d?d?d?d
```

```
选择命令提示符

The wordlist or mask that you are using is too small.
This means that hashcat cannot use the full parallel power of your device(s).
Unless you supply more work, your cracking speed will drop.
For tips on supplying more work, see: https://hashcat.net/faq/morework

Approaching final keyspace - workload adjusted.

85910b386dd855476a3f5bef82632e9f:bcf6859e4ea3:74de2b685f7c:D-Link_DIR-600A:13910407686

Session..........: hashcat
Status...........: Cracked
Hash.Type........: WPA-EAPOL-PBKDF2
Hash.Target......: D-Link_DIR-600A (AP:bc:f6:85:9e:4e:a3 STA:74:de:2b:68:5f:7c)
Time.Started.....: Tue Jan 29 17:23:58 2019 (0 secs)
Time.Estimated...: Tue Jan 29 17:23:58 2019 (0 secs)
Guess.Mask.......: 1391040?d?d?d?d [11]
Guess.Queue......: 1/1 (100.00%)
Speed.#3.........:    64879 H/s (0.43ms) @ Accel:32 Loops:16 Thr:1024 Vec:1
Recovered........: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.........: 10000/10000 (100.00%)
Rejected.........: 0/10000 (0.00%)
Restore.Point....: 0/10000 (0.00%)
Restore.Sub.#3...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#3....: 13910400000 -> 13910409496
Hardware.Mon.#3..: Temp: 45c Util: 77% Core:1771MHz Mem:3504MHz Bus:8

Started: Tue Jan 29 17:23:50 2019
Stopped: Tue Jan 29 17:23:59 2019

C:\Users\qiyou\Desktop\hashcat-5.1.0\hashcat-5.1.0>
```

Others

1. 对于破解过的hash值，用hashcat64.exe hash --show查看结果
2. 所有的hash破解结果都在hashcat.potfile文件中
3. 如果破解的时间太长，可以按s键可以查看破解的状态，p键暂停，r键继续破解，q键退出破解。
4. 在使用GPU模式进行破解时，可以使用-O参数自动进行优化
5. 在实际破解中的建议，如果我们盲目的去破解，会占用我们大量的时间和资源

1. ■■■■■■■■■■■■■■
2. ■■■■■■■■zhang1999■■
3. ■■■■■■■■■■■■■■■■masks■■.hcmask■■■■■■■■■■■■■■■■

4.[redacted]

6.HashCat参数优化
考虑到hashcat的破解速度以及资源的分配，我们可以对一些参数进行配置
1.Workload tuning 负载调优。
该参数支持的值有1,8,40,80,160

```
--gpu-accel 160 [redacted]GPU[redacted]
```

2.Gpu loops 负载微调
该参数支持的值的范围是8-1024（有些算法只支持到1000）。

```
--gpu-loops 1024 [redacted]GPU[redacted]
```

3.Segment size 字典缓存大小
该参数是设置内存缓存的大小，作用是将字典放入内存缓存以加快字典破解速度，默认为32MB，可以根据自身内存情况进行设置，当然是越大越块了。

```
--segment-size 512 [redacted]
```

#Reference
Hashcat Wiki
https://klionsec.github.io/2017/04/26/use-hashcat-crack-hash/

上一篇：区块链安全—当游戏遇见区块链机制 下一篇：Ursnif木马分析： 隐写术万岁！

1. 11 条回复


爱学习的松松 2019-02-15 16:17:41

做的很好，非常受用，但是我有一些问题可以问问你吗？首先，特殊字符集我没有测出来，其次就是破解wifi的手握包也不会，还有就是提取部分hash值的方式！打扰了

0 回复Ta


By七友 2019-02-16 09:21:26

@爱学习的松松
1.特殊字符集看我上面的例子应该能理解吧。2.破解wifi握手包的话，需要转化格式的，转化格式的链接我上面已经给出来了。3.提取hash我上面也给了链接了，你直接按

0 回复Ta

爱学习的松松 2019-02-16 10:59:53

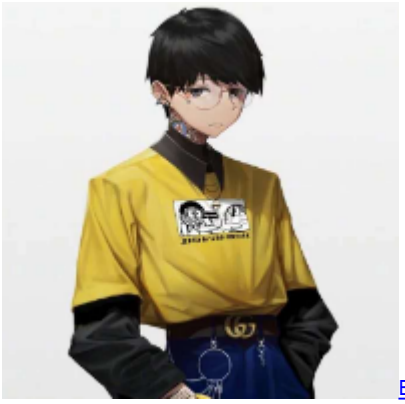
这是我测试的上面特定字符集的结果，能帮我看看有什么毛病吗？ 

0 回复Ta

爱学习的松松 2019-02-16 11:00:51

```
[root@mu01 hashcat]# ./hashcat -a 3 -1 123456abcdf!@+- 8b78ba5089b11326290bc15cf0b9a07d ?1?1?1?1?1
-bash: !@+-: event not found
```


0 回复Ta



By七友 2019-02-16 14:21:59

@爱学习的松松 你用单引号把特殊字符集引起来吧，因为在bash中!是有特殊意义的

0 回复Ta

hodo****o2019 2019-02-17 18:42:32

c:\1\hashcat64.exe -a 3 -m 0 --force 25c3e88f81b4853f2a8faacad4c871b6 ?d?d?d?d?d?d?d

```
C:\Users\Administrator>c:\1\hashcat64.exe -a 3 -m 0 --force 25c3e88f81b4853f2a8faacad4c871b6 ?d?d?d?d?d?d?d
hashcat (v5.1.0) starting...

./hashcat.hctune: No such file or directory

Started: Sun Feb 17 18:36:28 2019
Stopped: Sun Feb 17 18:36:28 2019
```


我新下载的5.1.0版本的，命令格式应该没有错，我用以前的旧版本，都破解出来了，用这个新版本，出现./hashcat.hctune：no no such file or directory

0 回复Ta

hodo****o2019 2019-02-17 18:43:22

请教一下，谢谢！难道是下载的文件出错了？我又重新到官网下载了一次。

0 回复Ta

hodo****o2019 2019-02-17 18:44:13

而这个hashcat.hctune文件，明明在文件夹里面的。

0 回复Ta

hodo****o2019 2019-02-17 20:55:35

我再去它官网上下载v4.2.1版本，一样出现了这个问题。
我再用我硬盘里面的2年前的3.2版本，却正常。
怪事了。

0 回复Ta

hodo****o2019 2019-02-17 21:03:11

一气之下，全部下载所有版本来试，一直试到4.0.1版本，正常了。
我怀疑是不是后面的新版本，对我的显卡兼容性有问题。

0 回复Ta


啦啦0咯咯 2019-07-23 15:40:50

@hodo****o2019 跟你一样的问题，我是把目录切到hashcat文件夹再运行命令就解决了

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板