

# HASHCAT



Password Cracking on  
Steroids



```
[root@tools ~]# whoami
```

**Martin (purehate) Bos**

- Core Developer for Backtrack Linux
- Owner of Computer Rehab
- Co-Founder Question-Defense
- Security Enthusiast





```
[root@tools ~]# whoami
```

Alex (dakykilla) Kah

- Technology Consultant





# Disclaimer

- We do not crack passwords for a living
- We do not claim to be experts
- We did not write Hashcat
- We have been known to be wrong
- We are just a couple geeks who happen to get excited by cracking password hash's





# Passwords are Important

- Primary user authentication
- Weakest link in a network
- Admin passwords





# So whats new?

- GPU based passed word cracking
- More complex rule sets
- Pattern detection software
- Faster CPU processing with SSE2
- Distributed Cracking
- Online hashlookup web sites





# What does this mean?

- Passwords that were once considered secure are no longer so
- Passwords that are hashed with out being salted are almost a joke
- Anything under 12 characters is easily broken





# Old School Attacks

The screenshot displays the main interface of Cain & Able, a network traffic analysis tool. The left sidebar lists various protocols and hash types, including LM & NTLM Hashes, MD5 Hashes (1), and SHA-1 Hashes (0). The main window shows a table with columns for MD5 Hash, Password, and Note. A single entry is visible with a red 'X' icon and the hash value 'b86fc6b051f63d73de2...'. A 'Brute-Force Attack' dialog box is open in the foreground, showing the following configuration:

- Charset:  Predefined (abcdefghijklmnopqrstuvwxyz0123456789)
- Password length: Min 1, Max 16
- Start from: (empty)
- Keyspace: 8.1860514273734411E+024
- Current password: cskrcc
- Key Rate: 6611313 Pass/Sec
- Time Left: 3.92707e+010 years

At the bottom of the dialog box, the text 'Cain & Able' is overlaid in a large, bold, black font. The status bar at the bottom of the application window shows 'http://www.oxid.it'.



# Old School Attacks

```
tools.question-defense.com - SecureCRT
File Edit View Options Transfer Script Tools Help
tools.question-defense.com
[root@tools run]# echo b86fc6b051f63d73de262d4c34e3a0a9 > md5.txt
[root@tools run]# ./john md5.txt
Loaded 2 password hashes with no different salts (LM DES [128/128 BS SSE2-16])
guesses: 0 time: 0:00:00:05 (3) c/s: 9234K trying: CL*AM16 - CL*ACHY
guesses: 0 time: 0:00:00:12 (3) c/s: 11441K trying: MCTJR - MCC88
guesses: 0 time: 0:00:00:14 (3) c/s: 11787K trying: 4MAIS09 - 4MAAASI
guesses: 0 time: 0:00:00:18 (3) c/s: 12206K trying: #X% - SAM1H
guesses: 0 time: 0:00:00:21 (3) c/s: 12208K trying: PFAK73 - PFAKYP
guesses: 0 time: 0:00:00:23 (3) c/s: 12276K trying: SOOKHY0 - SOONKPH
guesses: 0 time: 0:00:00:26 (3) c/s: 12480K trying: JIGGNCI - JIGB031
guesses: 0 time: 0:00:00:28 (3) c/s: 12724K trying: NT14G! - NT1S2T
guesses: 0 time: 0:00:00:31 (3) c/s: 12469K trying: 07JHG3 - 07JHU5
guesses: 0 time: 0:00:00:33 (3) c/s: 12459K trying: 0B63KU - 0B64W2
guesses: 0 time: 0:00:00:38 (3) c/s: 12689K trying: OCKLDVY - OCKLTH*
guesses: 0 time: 0:00:00:40 (3) c/s: 12984K trying: HDIISTA - HDI1S2!
guesses: 0 time: 0:00:00:43 (3) c/s: 13141K trying: 0S5G7J - 0S5MI2
guesses: 0 time: 0:00:00:47 (3) c/s: 13187K trying: OHEADC - OHEAH#
```

## John the Ripper



If Those are Old School, Then  
Whats New School?

Although John the Ripper  
and Cain are still very  
good password crackers  
they lack many of the  
combination, hybrid  
attack and speed  
characteristics of  
Hashcat & Oclhashcat





## Where Can I get the tools used in this demo?

- <http://hashcat.net/files/hashcat-0.34.rar>
- <http://hashcat.net/files/hashcat-gui-0.2.433.rar>
- <http://hashcat.net/files/oclHashcat-0.23.rar>
- <http://hashcat.net/files/hashcat-utils-0.1.rar>





# The Coolest Cat in Town...

Since We only have a hour  
it would be impossible to  
show all the features of  
Hashcat & Oclhashcat but we  
are going to try to show a  
few highlights





# Hashcat Features

- Free
- Multi-Threaded
- Multi-Hash
- Linux & Windows native binaries
- Fastest cpu-based multihash cracker
- SSE2 accelerated
- Rules mostly compatible with JTR and PasswordsPro
- Possible to resume or limit session
- Automatically recognizes recovered hashes from outfile at startup
- Can automatically generate random rules for Hybrid-Attack
- Able to work in an distributed environment
- Specify multiple wordlists and also multiple directories of wordlists
- Number of threads can be configured
- Threads run on lowest priority





# Hashcat Supported Algorithms

- MD5
- `md5 ($pass.$salt)`
- `md5 ($salt.$pass)`
- `md5 (md5 ($pass) )`
- `md5 (md5 (md5 ($pass) ) )`
- `md5 (md5 ($pass) .$salt)`
- `md5 (md5 ($salt) .$pass)`
- `md5 ($salt.md5 ($pass) )`
- `md5 ($salt.$pass.$salt)`
- `md5 (md5 ($salt) .md5 ($pass) )`
- `md5 (md5 ($pass) .md5 ($salt) )`
- `md5 ($salt.md5 ($salt.$pass) )`
- `md5 ($salt.md5 ($pass.$salt) )`
- `md5 ($username.0.$pass)`
- `md5 (strtoupper (md5 ($pass) ) )`
- SHA1
- `sha1 ($pass.$salt)`
- `sha1 ($salt.$pass)`
- `sha1 (sha1 ($pass) )`
- `sha1 (sha1 (sha1 ($pass) ) )`
- `sha1 (strtolower ($username) .$pass)`
- MySQL
- MySQL4.1/MySQL5
- MD5 (Wordpress)
- MD5 (phpBB3)
- MD5 (Unix)
- SHA-1 (Base64)
- SSHA-1 (Base64)
- SHA-1 (Django)
- MD4
- NTLM
- Domain Cached Credentials
- MD5 (Chap)
- MSSQL





# Gotta Have Some Stats

Windows 7, 64 bit - Core2Quad Q6600 @ 3.2Ghz - hashcat v0.34 - Pass wordsPro v3.0.0.0

Name	MD5 2 hashes	MD5 500k hashes	Wordpress 1 hash
hashcat (4 threads)	33.13M c/s	30.10M c/s	6450 c/s
hashcat (1 thread)	8.81M c/s	8.13M c/s	1870 c/s
PasswordsPro	7.27M c/s	0.73M c/s	945 c/s

Windows XP SP3, 32 bit - Intel Core 2 Duo E8400 @ 3.00 Ghz - hashcat v0.34 - Pass wordsPro v3.0.0.0

Name	MD5 2 hashes	MD5 500k hashes	Wordpress 1 hash
hashcat (2 threads)	18.34M c/s	17.72M c/s	3460 c/s
hashcat (1 thread)	9.74M c/s	9.46M c/s	1750 c/s
PasswordsPro	6.81M c/s	0.76M c/s	894 c/s

Ubuntu 9.10, 64 bit - AMD Athlon(tm) 64 X2 Dual Core Processor 6000+ - hashcat v0.34  
- JTR 1.7.5 (64 bit target) + jumbo3 patch

Name	MD5 2 hashes 31mb dict 35 rules	MD5 500k hashes 31mb dict 35 rules	Wordpress 1 hash 300kb dict
hashcat (2 threads)	7.43M c/s 0m14.574s	7.21M c/s 0m15.498s	2260 c/s 0m16.959s
hashcat (1 thread)	3.77M c/s 0m28.506s	3.66M c/s 0m29.794s	1130 c/s 0m33.878s
JTR	3.18M c/s 0m33.494s	2.31M c/s 0m46.811s	715 c/s 0m53.495s





# Hashcat Attacks

- Straight Words
- Combination Words
- Togglecase
- Permutation
- Brute force





# Hashcat Rules Demo Attack

```
hashcat-cli.exe -a 1  
-r C:\tools\hashcat-0.34\rules\d3ad0ne.rule -  
    m 0  
-o C:\tools\hashcat-0.34\cracked.txt  
-n 2 C:\tools\hashcat-0.34\rockyou.txt  
    C:\tools\hashcat-0.34\big.lst
```





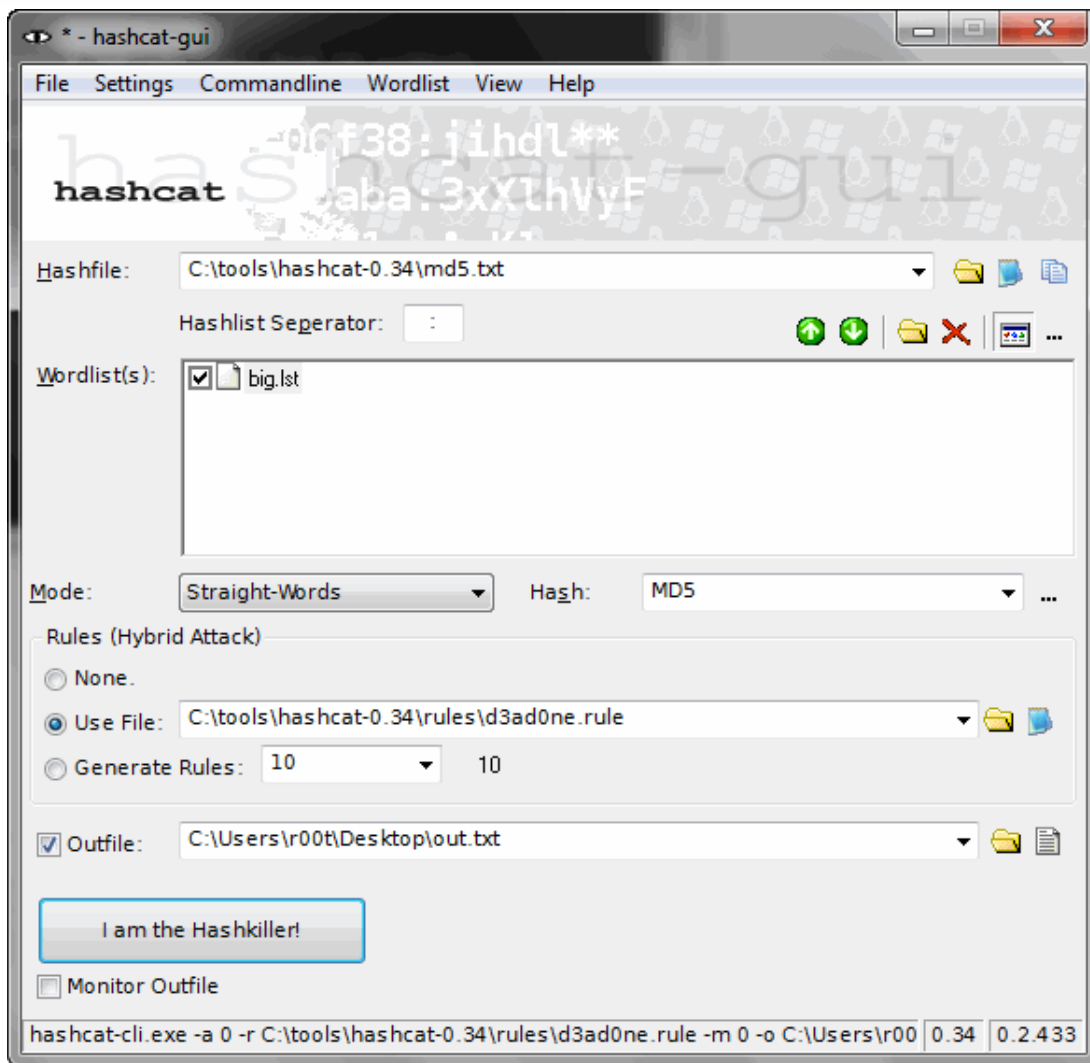
# Hashcat Permutation Demo Attack

```
hashcat-cli.exe -a 4 --perm-max 6 -m 0  
-o C:\tools\hashcat-0.34\cracked.txt  
-n 2 C:\tools\hashcat-0.34\rockyou.txt  
C:\tools\hashcat-0.34\big.lst
```





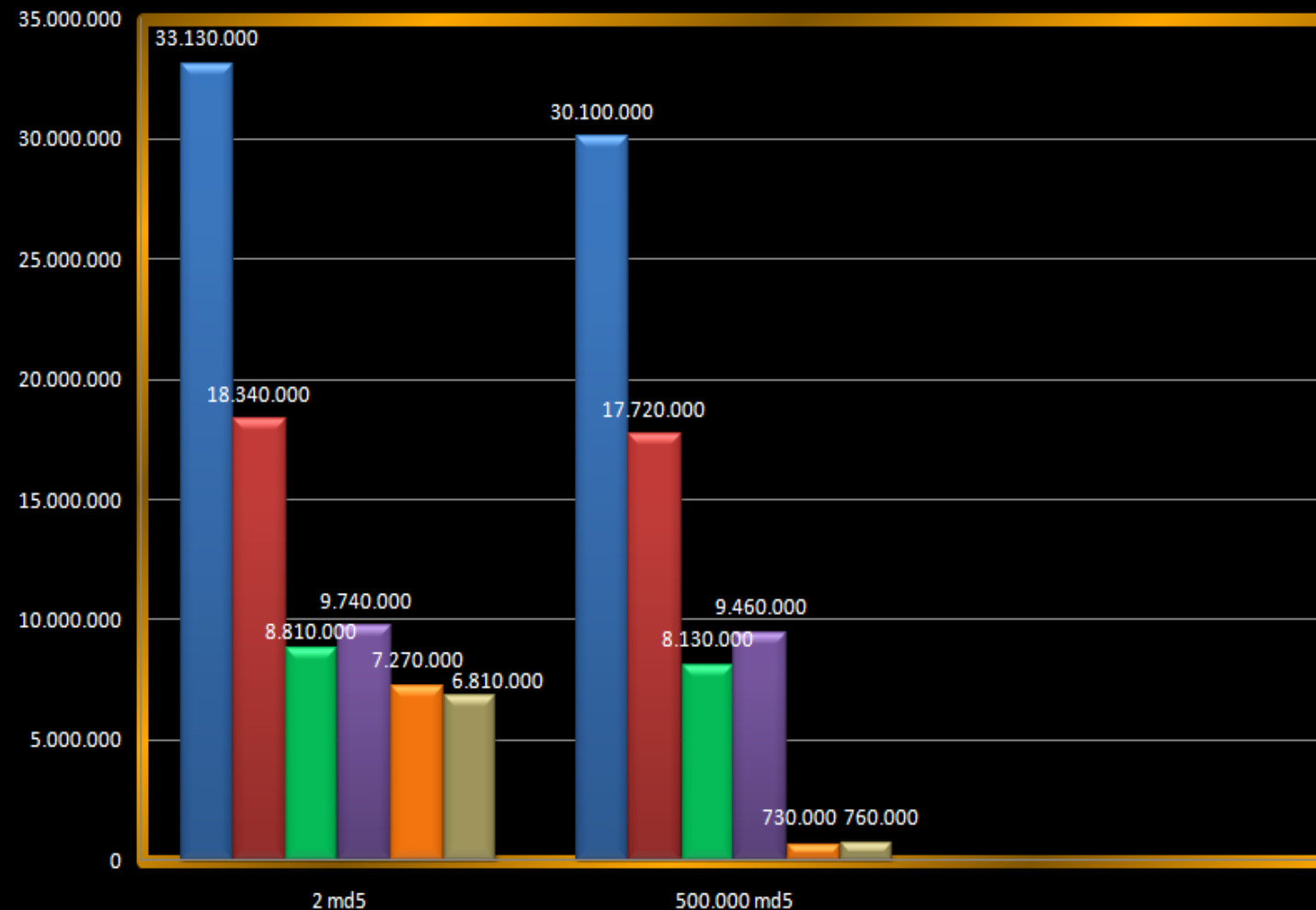
# Hashcat GUI





# Some more Hashcat Speed Tests...

## Speed Comparison



- hashcat (4 threads) Sys1
- hashcat (2 threads) Sys2
- hashcat (1 thread) Sys1
- hashcat (1 thread) Sys2
- PasswordsPro Sys1
- PasswordsPro Sys2

**System 1:**  
Windows 7, 64 bit  
Core2Quad Q6600 @ 3.2Ghz

**System 2:**  
Windows XP SP3, 32 bit Intel  
Core 2 Duo E8400 @ 3.00  
Ghz



# Oclhashcat Features

- Free
- Multi-GPU
- Multi-Hash
- Linux & Windows native binaries
- Uses OpenCL
- Fastest multihash MD5 cracker on NVidia cards
- Fastest multihash MD5 cracker on ATI 5xxx cards
- Supports wordlists (not limited to Brute-Force / Mask-Attack)
- Can mix wordlists with Mask-Attack to emulate Hybrid-Attacks
- Runs very cautious, you can still watch movies while cracking
- Kernel workload can be configured while cracking
- Supports pause / resume
- Supports huge numbers of hashes (4 million and more)
- Able to work in a distributed environment
- Includes hashcats entire rule engine to modify wordlists on start





# Oclashcat Supported Algorithms

- MD5
- `md5 ($pass.$salt)`
- `md5 ($salt.$pass)`
- `md5 (md5 ($pass) )`
- `md5 (md5 ($pass) .$salt)`
- SHA1
- `sha1 ($pass.$salt)`
- `sha1 ($salt.$pass)`
- MySQL
- MySQL4.1/MySQL5
- MD4
- NTLM
- Domain Cached Credentials





# More Stats...

Wn7, 64bit, Catalyst 10.7b, ATI Steam SDK v2.2, ATI HD5770 x 2, oclHashcat v0.23, IGHASHGPU v0.80.16.1

Name	MD5 2 hashes	MD5 500k hashes	LOSS
oclHashcat	2392M c/s	1089M c/s	54.4%
IGHASHGPU	2164M c/s	832M c/s	61.5%

Wn7, 64bit, ForceWare 258.96, NVidia GTX285 (oc), oclHashcat v0.23, IGHASHGPU v0.80.16.1, EGB v1.6.1  
Lightning Hash Cracker v0.55

Name	MD5 2 hashes	MD5 500k hashes	LOSS
oclHashcat	665M c/s	654M c/s	1.7%
IGHASHGPU	647M c/s	615M c/s	5.0%
EGB	632M c/s	495M c/s	21.6%
LHC	430M c/s	130M c/s	69.7%

Ubuntu 9.10, 64bit, ForceWare 256.40, NVidia 8800gt, oclHashcat v0.23, 64 bit, CUDA-Multiforcer v0.72, 64 bit

Name	MD5 2 hashes	MD5 500k hashes	LOSS
oclHashcat	300M c/s	293M c/s	5.7%
CUDA-Multiforcer	271M c/s	119M c/s	57.1%





# Oclashcat Attacks

- **Bruteforce**
- **Hybrid Masks**
- **Fingerprint**





# Character Sets Explained

?l = Lowercase

?u = Uppercase

?d = Digits

?s = Special Characters

You can specify more than one with -l

?l?u?d and the specify ?l in the left  
and right mask.

```
--custom-charset1= ?dabcdef
```

```
sets charset ?l to 0123456789abcdef
```





# Bruteforce example attack

```
oclHashcat64.bin md5.txt -m 0 -n 160 --gpu-  
devices=1,2,3,4,5,6,7,8  
--gpu-loops=1024 -1 ?1?d?s?u ?1?1?1?1 ?1?1
```

md5.txt = List of Hash's to be cracked

-m = Type of hash

-n = Workload tuning

--gpu-devices = OCL devices to use

--gpu-loops = Workload fine-tuning if -n is not precise enough

?1?1?1?1 = Left mask

?1?1 = right mask





Lets do a quick demo  
of the brute force  
attack





# Hybrid Example Attack

```
oclHashcat64.bin md5.txt -m 0 -n 160  
--gpu-devices=1,2,3,4,5,6,7,8  
--gpu-loops=1024  
-1 ?1?d?s?u $DICT_FILE ?1?1
```

This attack will append a-z A-Z  
0-9 and all special characters to the end  
of every word in the dictionary





Lets do a quick demo  
of the hybrid attack





# Fingerprint Attack

- Fingerprinting is using common patterns users use to create passwords
- Common patterns like adding a 1 or a date to a password are no longer safe so users are creating more complex patterns
- Using a dictionary expander we can create all the patterns possible from a given wordlist





# Basic analysis of the attack

1. Bruteforce the list of hash's with a simple 5 or 6 character attack which will give us a small password list to begin with
2. Remove the hash portion of the list leaving us with a small dictionary file
3. Run the dictionary file through the expander
4. Now we use Oclhashcat's Combination engine with our new dictionary file on the left mask and the right mask which will increase our chances of success even more.
5. This will give us a second set of cracked passwords to work.
6. We now run that list through the expander and then repeat the Combination attack
7. The attack can be repeated using these steps until no more passwords are found.





Lets do a demo of the  
fingerprint attack





# Password fingerprinting tips and tricks

- Fingerprinting attack is designed for use with GPUs
- Fingerprint attacks can be automated
- You can use your own wordlists as well
- Be careful not use use huge wordlists with the expander
- Build your own pattern dictionary
- Limit the lengths of the patterns





## What's The best Place to get a Wordlist?

- <http://www.skullsecurity.org/wiki/index.php/Passwords>
- <http://hashcrack.blogspot.com/p/wordlist-downloads.html>





This password crap sucks, I  
hate doing it...

Let us do the work for you. We have a  
online password cracking service at  
[tools.question-defense.com](http://tools.question-defense.com)

Which supports

WPA, ntlm, md5, md4, sha1 and rar

We are currently adding more  
algorithms and much more speed





# What have we learned?

Treat your password  
like your  
toothbrush. Do not  
let anyone else use  
it and get a new one  
every 6 months





# Questions?

