WPA2 Cracking Using HashCat

If you remember in the previous part, we learned Speeding up WPA2 Cracking Using Pre-generated PMKs. Which certainly uses CPU as the primary part for the calculations of the PMKs. It surely gives us speed for WPA2 cracking as while using PMKs for cracking we are not performing actual calculations in real-time.

This brings us to some drawbacks of using PMKs, as follows:

- **SSID Specific**. You cannot use PMKs generated for SSID, say “*rootsh3ll”* for another SSID like “*Belkin*“.
- **Case-Sensitive**. Cannot be used even a single letter is up/lower case. Ex: Won’t work for “*Rootsh3ll***“** if PMKs are created for “*rootsh3ll*“.
- **Time used is the same**. As processing power of CPU is same in both cases, the time required for creating PMKs are equal even if you crack using Aircrack or creating PMKs(with GenPMK).
- **Huge HD Space required**. As we are pre-calculating the PMKs and storing them on HD, it requires a lot of space on your HD and that too for a specific SSID. Which is not an option all the time.
- **Less helpful in today’s scenario**. Nowadays routers are being shipped with unique SSID. Ex: *Belkin_04A2* for preventing routers from these kind of attacks or atleast delay the cracking duration.

You might be thinking now that If this is so, then why would I even consider PMKs for WPA2 cracking ?

Well, as I said above this is Less helpful, that means in some cases.

Cases like:

- Simple SSIDs. Ex: MTNL, Airtel, Linksys etc
- Before trying any complex task to crack the PSK, if you have PMKs already stored. Give them a shot
- Mobile numbers are still very common passwords.

Still, even if this gives us speed this method is a bit slow. You don't always have a friend ready to give you a pre-generated PMK file for a specific SSID just when you have captured the handshake, right? yeah, it's very rare!

Here is when you need to stop using your CPU and test the processing power of you GPU.

If you are not aware of using GPUs for WPA2 cracking purposes let me tell you, Yes GPUs can be used for WPA2 cracking password hashes and are being used now from a while.

There are plenty of tools which uses GPU to boost the WPA2 cracking speed and lets you crack in way much lesser time that your CPU would have the job finished.

Tools like:

- Pyrit
- BarsWF
- HashCat
- igHashGPU

How ? Simple! Your CPU has 2,4,8 cores, means parallel computing units where GPUs have them in thousands, if not hundreds.

> NOTE: My GeForce GT 525M have 296 cores, and it is pretty old Graphics card, Speed: ~6000 PMK/s. NVidia Titan X is the Best single graphics card with cracking speed up to 2,096,000 hashes/sec.

Using GPU for Cracking WPA2 Passwords

Being in the scope of the series we will stick to WPA2 cracking with GPU in this chapter. For learning difference between CPU and GPU cracking you can visit the following post I'd previously written on FromDev.com.

CPU vs. GPU Password Hash Cracking – FromDev.com

Tools described above are used for cracking various kinds of passwords.

There are 2 tools used for WPA2 cracking using GPU from the above list

- Pyrit
- HashCat

As the post title suggests we will go with HashCat.

What is HashCat ?

Ocl/CUDA/HashCat is now Open Source. Checkout at GitHub: github.com/hashcat

Hashcat is a self-proclaimed command line based world's fastest password cracker.

It is the world's first and only **GPGPU** based rule engine and available for Linux, OSX, and Windows free-of-cost. It comes in 2 variants

- CPU Based
- GPU Based

There is no difference when passing commands to Hashcat because it automatically uses the best method to crack passwords, either CPU or GPU depending on the Graphics driver you have installed or not.

Hashcat is fast and extremely flexible- to writer made it in such a way that allows distributed cracking. I highly recommend Hashcat over Pyrit for its flexibility.

Why use HashCat at first place ?

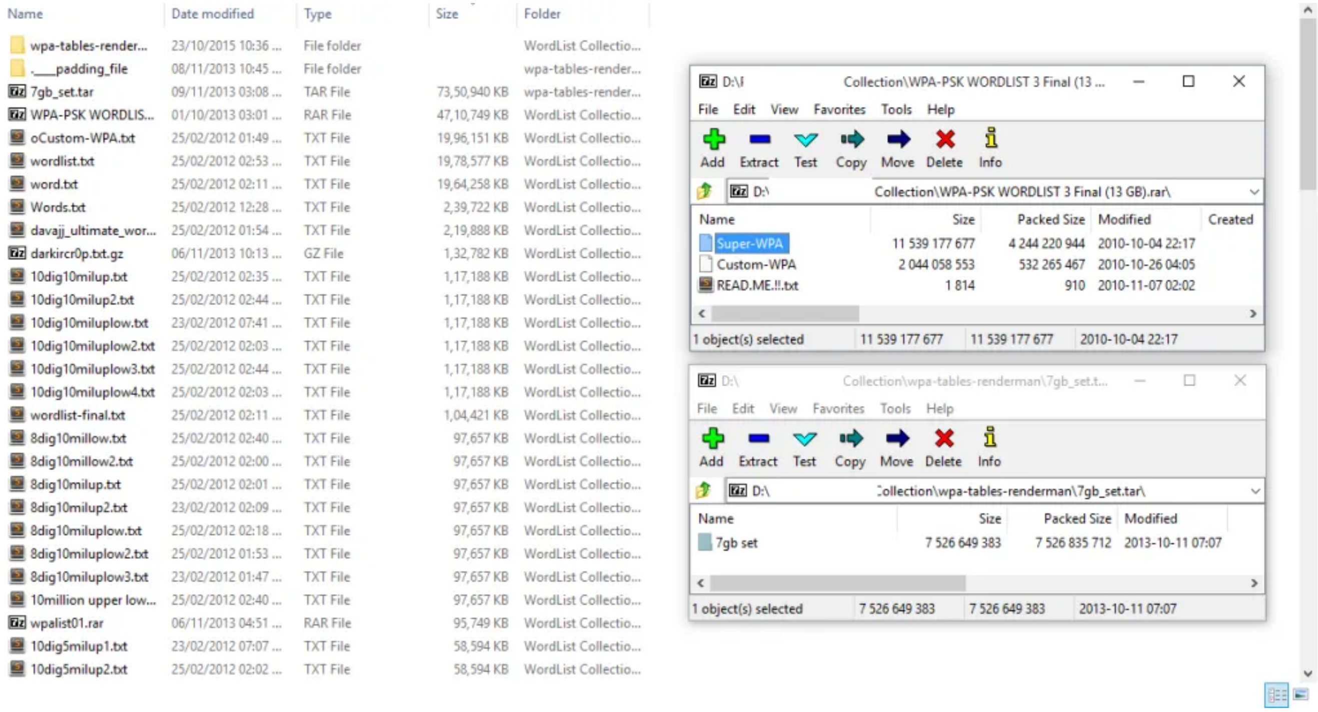
As already told above, because of it's flexibility and vast support of algorithms.

But why Hashcat when I just want to WPA2 cracking most of the times ?

If you have used or haven't used Pyrit yet, let me tell you one thing. Pyrit wasthe fastest WPA2 cracker available in its early times but it uses dictionary or wordlist to crack the passwords even if you use PMKs or directly run the cracker you need to have a large amount of dictionaries to test the validity of the hash.

Also Hashcat has been outperforming Pyrit for many years now. and if you are still using Pyrit, Time for switching to Hashcat is now!

For storing hashes you need a lot of disk space. As you can see in the image below, there is a few wordlists that almost take >25 GB on the disk(Extracted), and it take more than 2-3 days to run through them all even with GPU.



Name	Date modified	Type	Size	Folder
wpa-tables-render...	23/10/2015 10:36 ...	File folder		WordList Collectio...
.___padding_file	08/11/2013 10:45 ...	File folder		wpa-tables-render...
7gb_set.tar	09/11/2013 03:08 ...	TAR File	73,50,940 KB	wpa-tables-render...
WPA-PSK WORDLIS...	01/10/2013 03:01 ...	RAR File	47,10,749 KB	WordList Collectio...
oCustom-WPA.txt	25/02/2012 01:49 ...	TXT File	19,96,151 KB	WordList Collectio...
wordlist.txt	25/02/2012 02:53 ...	TXT File	19,78,577 KB	WordList Collectio...
word.txt	25/02/2012 02:11 ...	TXT File	19,64,258 KB	WordList Collectio...
Words.txt	25/02/2012 12:28 ...	TXT File	2,39,722 KB	WordList Collectio...
davajj_ultimate_wor...	25/02/2012 01:54 ...	TXT File	2,19,888 KB	WordList Collectio...
darkircr0p.txt.gz	06/11/2013 10:13 ...	GZ File	1,32,782 KB	WordList Collectio...
10dig10milup.txt	25/02/2012 02:35 ...	TXT File	1,17,188 KB	WordList Collectio...
10dig10milup2.txt	25/02/2012 02:44 ...	TXT File	1,17,188 KB	WordList Collectio...
10dig10miluplow.txt	23/02/2012 07:41 ...	TXT File	1,17,188 KB	WordList Collectio...
10dig10miluplow2.txt	25/02/2012 02:03 ...	TXT File	1,17,188 KB	WordList Collectio...
10dig10miluplow3.txt	25/02/2012 02:44 ...	TXT File	1,17,188 KB	WordList Collectio...
10dig10miluplow4.txt	25/02/2012 02:03 ...	TXT File	1,17,188 KB	WordList Collectio...
wordlist-final.txt	25/02/2012 02:11 ...	TXT File	1,04,421 KB	WordList Collectio...
8dig10millow.txt	25/02/2012 02:40 ...	TXT File	97,657 KB	WordList Collectio...
8dig10millow2.txt	25/02/2012 02:00 ...	TXT File	97,657 KB	WordList Collectio...
8dig10milup.txt	25/02/2012 02:01 ...	TXT File	97,657 KB	WordList Collectio...
8dig10milup2.txt	23/02/2012 02:09 ...	TXT File	97,657 KB	WordList Collectio...
8dig10miluplow.txt	25/02/2012 02:18 ...	TXT File	97,657 KB	WordList Collectio...
8dig10miluplow2.txt	25/02/2012 01:53 ...	TXT File	97,657 KB	WordList Collectio...
8dig10miluplow3.txt	23/02/2012 01:47 ...	TXT File	97,657 KB	WordList Collectio...
10million upper low...	25/02/2012 02:40 ...	TXT File	97,657 KB	WordList Collectio...
wpalist01.rar	06/11/2013 04:51 ...	RAR File	95,749 KB	WordList Collectio...
10dig5milup1.txt	23/02/2012 07:07 ...	TXT File	58,594 KB	WordList Collectio...
10dig5milup2.txt	25/02/2012 02:02 ...	TXT File	58,594 KB	WordList Collectio...

D:\ Collection\WPA-PSK WORDLIST 3 Final (13 ...

D:\ Collection\WPA-PSK WORDLIST 3 Final (13 GB).rar\

Name	Size	Packed Size	Modified	Created
Super-WPA	11 539 177 677	4 244 220 944	2010-10-04 22:17	
Custom-WPA	2 044 058 553	532 265 467	2010-10-26 04:05	
READ.ME.!!.txt	1 814	910	2010-11-07 02:02	

1 object(s) selected | 11 539 177 677 | 11 539 177 677 | 2010-10-04 22:17

D:\ Collection\wpa-tables-renderman\7gb_set.t...

D:\ :ollection\wpa-tables-renderman\7gb_set.tar\

Name	Size	Packed Size	Modified
7gb set	7 526 649 383	7 526 835 712	2013-10-11 07:07

1 object(s) selected | 7 526 649 383 | 7 526 649 383 | 2013-10-11 07:07

You can download some useful wordlists here.

But most of the times there are some pattern(default passwords) we like to test for validity. Patterns like:

- Mobile number
- Date of Birth
- Default password patterns like "**56324FHe**"
- 10 digit default password by ISP
- and so on

Here is when We have to leave Pyrit with it's dictionaries and get our hands-on with HashCat.

HashCat have a brilliant feature called mask-attack, which allows us to create user-defined patterns to test for password validity and you know what the best thing is ? It requires 0 Bytes on your hard drive.

How ?

Before we go through this we need to understand that in some cases we need **Wordlists**. Its only when we are 100% certain that it has some kind of pattern we can use this type of attack. So of you know a certain ISP has 10 random numbers and only a few letters, you could do it to save space on your HD.

WPA2 cracking is a tedious task and uses maximum power of the system when we use Hashcat for the purpose and sometimes it needs to take down the load from the system to switch tasks. hashcat stands best here for it's remarkable feature.

- It supports **pause**/**resume** while cracking
- Supports **sessions** and **restore**

We will see this feature in this tutorial. Keep reading.

Supported Attack types

- Dictionary based attack

- Brute-force/Mask attack
- Hybrid dict + mask
- Hybrid mask + dict
- Permutation attack
- Rule-based attack
- Toggle-case attack

These are too name a few. Hashcat supports way too many algorithms to get your hash cracked.

> NOTE: Traditional Brute-force attack is outdated and is replaced by Mask attack in Hashcat. Wee will see later in this post in details about this.

Variants

As told above Hashcat comes in 2 vaiants:

1. Hashcat -A CPU based password cracker
2. Oclhashcat/CudaHashcat – GPU accelerated tool

Setting up the Lab

Installing Graphics driver

You have basically 2 choices

1. Install graphics driver in Kali Linux directly, i.e your Pentesting distro.
2. Install graphics driver in Windows or OSX.

I have Kali Sana installed in my Virtual machine and unfortunately no virtual machine supports using graphics card or GPU acceleration inside the virtual OS. So I'll be sticking with Hashcat on windows. You can still do the same task with exact same commands on Kali Linux(or any Linux OS) or OSX with properly installed proprietary drivers.

I haven't written any article on how to install graphics drier in Kali Linux as BlackmoreOps already have a great article on same. so you can follow the links and try installing the same on your version of Kali.

NVidia Users:

- Install proprietary NVIDIA driver on Kali Linux – NVIDIA Accelerated Linux Graphics Driver
- Install NVIDIA driver kernel Module CUDA and Pyrit on Kali Linux – CUDA, Pyrit and Cpyrit-cuda

AMD Users:

- Install AMD ATI proprietary fglrx driver in Kali Linux 1.x/2.x
- Install AMD APP SDK in Kali Linux
- Install CAL++ in Kali Linux

Download HashCat

You can download Hashcat from it's official website: http://hashcat.net/

File is highly compressed using 7z compression. So make sure you have atleast 1 GB before extracting the downloaded file.

You can use 7zip extractor to decompress the .7z file. Download it here: http://www.7-zip.org/download.html

P.S: It is free of use and better than WinRAR.

Cleanup your cap file using wpaclean

Next step will be converting the `.cap` file to a format cudaHashcat or oclHashcat or Hashcat on Kali Linux will understand.

Here's how to do it:

To convert your **`.cap`** files manually in Kali Linux, use the following command

```
wpaclean <out.cap> <in.cap>
```

Please note that the `wpaclean` options are the wrong way round. < `out.cap` > < `in.cap` > instead of < `in.cap` > < `out.cap` > which may cause some confusion.

Convert .cap file to .hccapx file

Now assuming that you have installed appropriate graphics driver for the selected OS, moving on to the nest step. We need to convert the previously captured handshake i.e .cap file to a format that hashcat could understand and it is .hccapx file format.

Nothing difficult or time taking. Command to convert .cap to .hccapx goes like this

- aircrack-ng -j <output.hccapx> <path/to/.cap file>

Here output.hccapx is the output filename with .hccapx file format and input.cap is the handshake originally captured.

Log in to Kali Linux, open Terminal and type:

```
aircrack-ng -j "rootsh3ll-01.hccapx"  "rootsh3ll-01.cap"
```

Note: rootsh3ll-01.cap is located on Desktop. Check location of your .cap file.

Now we have .hccapx file, installed graphics driver and downloaded hashcat. Let's begin the cracking.

Cracking WPA2 Passwords using Hashcat

We will cover the following topics:

- WPA2 Cracking with Dictionary attack using Hashcat.
- WPA2 Cracking with Mask attack using Hashcat.

- WPA2 Cracking with Hybrid attack using Hashcat.
- WPA2 Cracking Pause/resume in Hashcat (One of the best features)
- WPA2 Cracking save sessions and restore.

WPA2 dictionary attack using Hashcat

Open cmd and direct it to Hashcat directory, copy .hccapx file and wordlists and simply type in cmd

```
cudaHashcat64.exe -m 2500 rootsh3ll-01.hccapx  wordlist.txt wordlist2.txt
```

Here I have NVidia's graphics card so I use CudaHashcat command followed by 64, as I am using Windows 10 64-bit version. yours will depend on graphics card you are using and Windows version(32/64).

cudaHashcat64.exe – The program, In the same folder theres a cudaHashcat32.exe for 32 bit OS and cudaHashcat32.bin / cudaHashcat64.bin for Linux. *oclHashcat*.exe* for AMD graphics card.

-m 2500 = The specific hashtype. 2500 means WPA/WPA2.

In case you forget the WPA2 code for Hashcat.

Windows CMD: **cudaHashcat64.exe –help | find "WPA"**

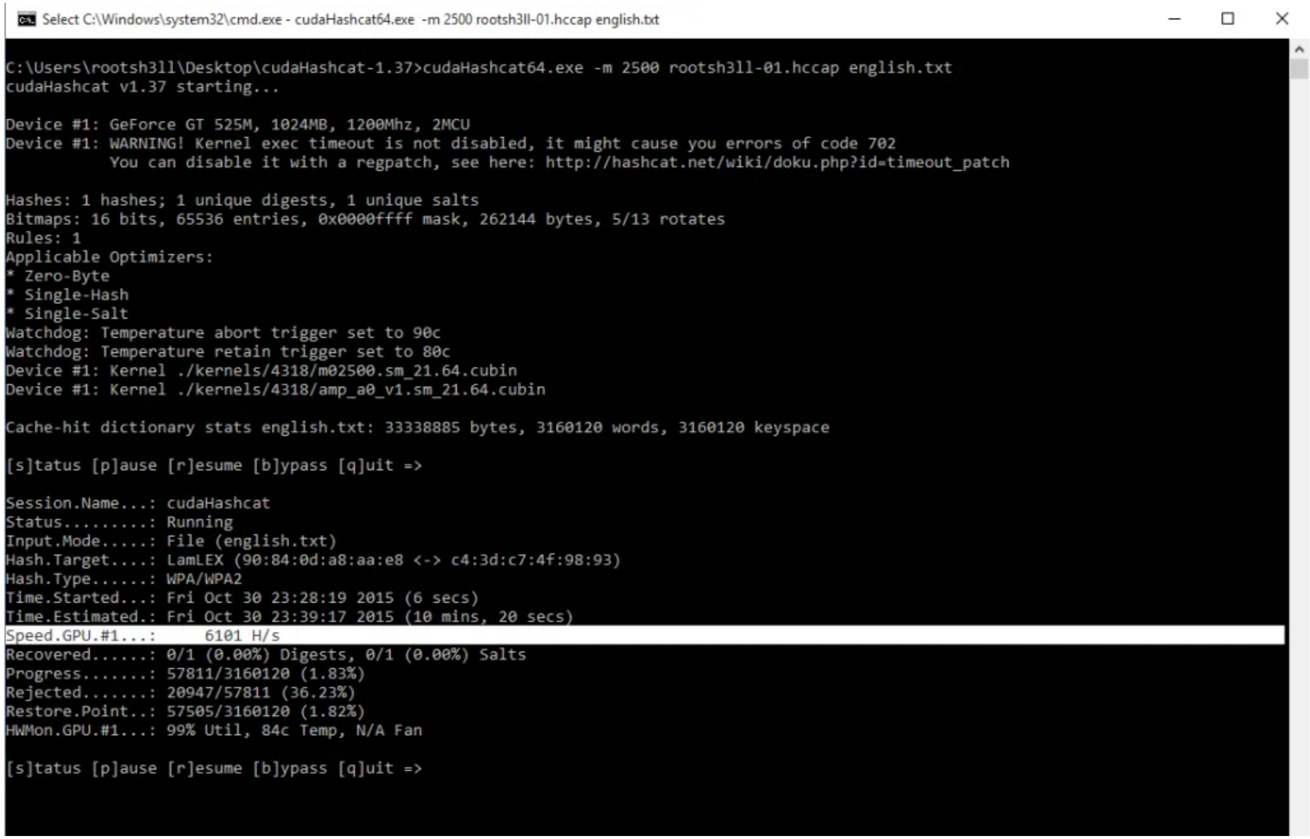
Linux Terminal:**cudaHashcat64.bin –help | grep "WPA"**

It will show you the line containing “WPA” and corresponding code.

Handshake-01.hccap = The converted *.cap file.

wordlist.txt wordlist2.txt= The wordlists, you can add as many wordlists as you want. To simplify it a bit, every wordlist you make should be saved in the CudaHashcat folder.

After executing the command you should see a similar output:



Select C:\Windows\system32\cmd.exe - cudaHashcat64.exe -m 2500 rootsh3ll-01.hccap english.txt

```
C:\Users\rootsh3ll\Desktop\cudaHashcat-1.37>cudaHashcat64.exe -m 2500 rootsh3ll-01.hccap english.txt
cudaHashcat v1.37 starting...

Device #1: GeForce GT 525M, 1024MB, 1200Mhz, 2MCU
Device #1: WARNING! Kernel exec timeout is not disabled, it might cause you errors of code 702
           You can disable it with a regpatch, see here: http://hashcat.net/wiki/doku.php?id=timeout_patch

Hashes: 1 hashes; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
Applicable Optimizers:
* Zero-Byte
* Single-Hash
* Single-Salt
Watchdog: Temperature abort trigger set to 90c
Watchdog: Temperature retain trigger set to 80c
Device #1: Kernel ./kernels/4318/m02500.sm_21.64.cubin
Device #1: Kernel ./kernels/4318/amp_a0_v1.sm_21.64.cubin

Cache-hit dictionary stats english.txt: 33338885 bytes, 3160120 words, 3160120 keyspace

[s]tatus [p]ause [r]esume [b]ypass [q]uit =>

Session.Name...: cudaHashcat
Status.........: Running
Input.Mode.....: File (english.txt)
Hash.Target....: LamLEX (90:84:0d:a8:aa:e8 <-> c4:3d:c7:4f:98:93)
Hash.Type......: WPA/WPA2
Time.Started...: Fri Oct 30 23:28:19 2015 (6 secs)
Time.Estimated.: Fri Oct 30 23:39:17 2015 (10 mins, 20 secs)
Speed.GPU.#1...:     6101 H/s
Recovered......: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.......: 57811/3160120 (1.83%)
Rejected.......: 20947/57811 (36.23%)
Restore.Point..: 57505/3160120 (1.82%)
HWMon.GPU.#1...: 99% Util, 84c Temp, N/A Fan

[s]tatus [p]ause [r]esume [b]ypass [q]uit =>
```

Wait for Hashcat to finish the task. You can pass multiple wordlists at once so that Hashcat will keep on testing next wordlist until the password is matched.

WPA2 Mask attack using Hashcat

As told earlier, Mask attack is a replacement of the traditional Brute-force attack in Hashcat for better and faster results.

let’s have a look at what Mask attack really is.

In Terminal/cmd type:

- cudaHashcat64.exe -m 2500 <rootsh3ll-01.hccapx> -a 3 ?d?l?u?d?d?d?u?d?s?a

-a 3 is the Attack mode, custom-character set (Mask attack)

?d?l?u?d?d?d?u?d?s?a is the character-set we passed to Hashcat. Let's understand it in a bit of detail that

- What is a character set in Hashcat ?
- Why it is useful ?

What is a character set in Hashcat ?

?d ?l ?u ?d ?d ?d ?u ?d ?s ?a = 10 letters and digits long WPA key. Can be 8-63 char long.

The above text string is called the "Mask". Every pair we used in the above examples will translate into the corresponding character that can be an Alphabet/Digit/Special character.

For remembering, just see the character used to describe the charset

?d: For digits

?s: For Special characters

?u: For Uppercase alphabets

?l: For Lowercase alphabets

?a: all of the above.

Simple! isn’t it ?

Here is the actual character set which tells exactly about what characters are included in the list:

```
?l = abcdefghijklmnopqrstuvwxyz
?u = ABCDEFGHIJKLMNOPQRSTUVWXYZ
?d = 0123456789
?s = «space»!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
?a = ?l?u?d?s
```

Here are a few examples of how the PSK would look like when passed a specific Mask.

PSK = ?d?l?u?d?d?d?u?d?s?a

0aC575G2/@
9zG432H0*K
8sA111W1$4
3wD001Q5+z

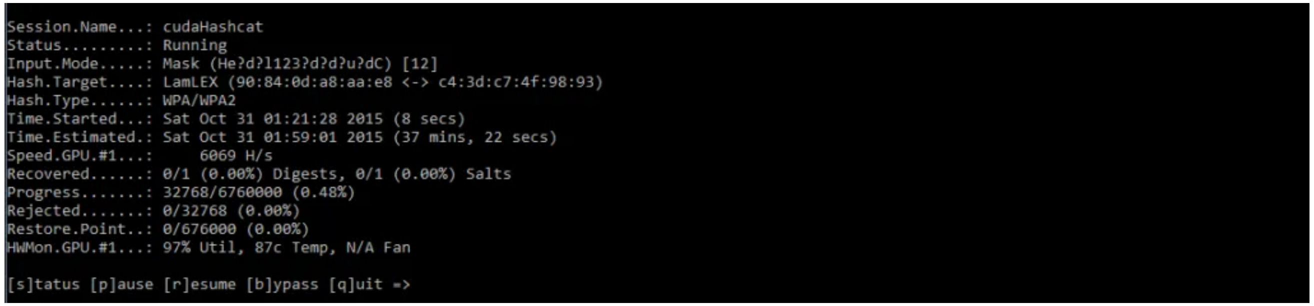
So now you should have a good understanding of the mask attack, right ?

Let’s dig a bit deeper now.

Mixing Mask attack with Custom characters.

Let's say, we somehow came to know a part of the password. So, it would be better if we put that part in the attack and randomize the remaining part in Hashcat, isn't it ?

Sure! it is very simple. Just put the desired characters in the place and rest with the Mask.



```
Session.Name...: cudaHashcat
Status.........: Running
Input.Mode.....: Mask (He?d?l123?d?d?u?dC) [12]
Hash.Target....: LamLEX (90:84:0d:a8:aa:e8 <-> c4:3d:c7:4f:98:93)
Hash.Type......: WPA/WPA2
Time.Started...: Sat Oct 31 01:21:28 2015 (8 secs)
Time.Estimated.: Sat Oct 31 01:59:01 2015 (37 mins, 22 secs)
Speed.GPU.#1...:     6069 H/s
Recovered......: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.......: 32768/6760000 (0.48%)
Rejected.......: 0/32768 (0.00%)
Restore.Point..: 0/676000 (0.00%)
HWMon.GPU.#1...: 97% Util, 87c Temp, N/A Fan

[s]tatus [p]ause [r]esume [b]ypass [q]uit =>
```

He ?d ?l **123** ?d ?d ?u ?d **C** is the custom Mask attack we have used. Here assuming that I know the first 2 characters of the original password then setting the 2nd and third character as digit and lowercase letter followed by "123" and then "?d ?d ?u ?d" and finally ending with "C" as I knew already.

What we have actually done is that we have simply placed the characters in the exact position we knew and Masked the unknown characters, hence leaving it on to Hashcat to test further.

Here is one more example for the same:

Let's say password is "Hi123World" and I just know the "Hi123" part of the password, and remaining are lowercase letters. Assuming length of password to be 10.

So I would simply use the command below

```
cudaHashcat64.exe -m 2500 <handshake.hccap> -a 3 Hi123?u?u?u?u?u
```

Where ?u will be replaced by uppercase letters, one by one till the password is matched or the possibilities are exhausted.

Moving on even further with Mask attack i.r the Hybrid attack.

In hybrid attack what we actually do is we don't pass any specific string to hashcat manually, but automate it by passing a wordlist to Hashcat.

Hashcat picks up words one by one and test them to the every password possible by the Mask defined.

Example:

- cudaHashcat64.exe -m 2500 handshake.hccapx -a 1 password.txt ?d?l?d?l

-a 1 : The hybrid attack
password.txt : wordlist
?d?l?d?l = Mask (4 letters and numbers)

The wordlist contains 4 words.

```
carlos

bigfoot
```

```
guest
onion
```

Now it will use the words and combine it with the defined Mask and output should be this:

carlos2e1c

bigfoot0h1d

guest5p4a

onion1h1h

It is cool that you can even reverse the order of the mask, means you can simply put the mask before the text file. Hashcat will bruteforce the passwords like this:

7a2ecarlos

8j3abigfoot

0t3wguest

6a5jonion

You getting the idea now, right ?

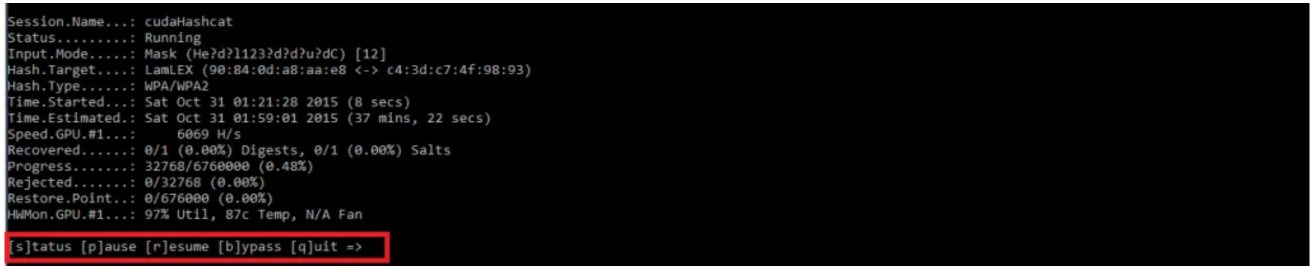
Using so many dictionary at one, using long Masks or Hybrid+Masks takes a long time for the task to complete. It is not possible for everyone every time to keep the system on and not use for personal work and the Hashcat developers understands this problem very well. So, they came up with a brilliant solution which no other password recovery tool offers built-in at this moment. That is the Pause/Resume feature

WPA2 Cracking Pause/resume in Hashcat (One of the best features)

This feature can be used anywhere in Hashcat. It isn't just limited to WPA2 cracking. Even if you are cracking md5, SHA1, OSX, wordpress hashes. As soon as the process is in running state you can pause/resume the process at any moment.

Just press [**p**] to pause the execution and continue your work.

To resume press [**r**]. All the commands are just at the end of the output while task execution. See image below



```
Session.Name...: cudaHashcat
Status.........: Running
Input.Mode.....: Mask (He?d?l123?d?d?u?dC) [12]
Hash.Target....: LamLEX (90:84:0d:a8:aa:e8 <-> c4:3d:c7:4f:98:93)
Hash.Type......: WPA/WPA2
Time.Started...: Sat Oct 31 01:21:28 2015 (8 secs)
Time.Estimated.: Sat Oct 31 01:59:01 2015 (37 mins, 22 secs)
Speed.GPU.#1...:     6069 H/s
Recovered......: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.......: 32768/6760000 (0.48%)
Rejected.......: 0/32768 (0.00%)
Restore.Point..: 0/676000 (0.00%)
HWMon.GPU.#1...: 97% Util, 87c Temp, N/A Fan

[s]tatus [p]ause [r]esume [b]ypass [q]uit =>
```

You might sometimes feel this feature as a limitation as you still have to keep the system awake, so that the process doesn't gets cleared away from the memory.

And we have a solution for that too. Create session!

WPA2 Cracking save Sessions and Restore.

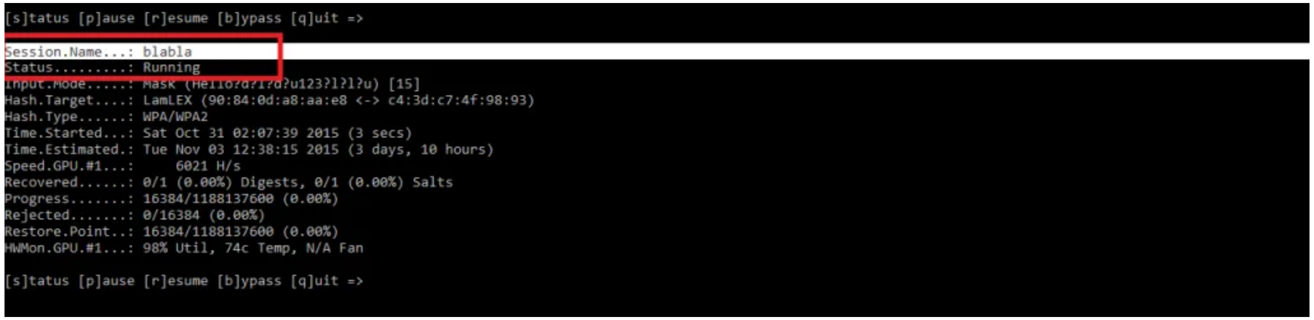
Creating and restoring sessions with hashcat is Extremely Easy.

Just add –session at the end of the command you want to run followed by the session name.

Example:

```
cudaHashcat64.exe -m 2500 rootsh3ll-01.hccapx -a 3 Hello?d?l?d?u123?l?l?u --session=blabla
```

Here I named the session “blabla”. You can see in the image below that Hashcat has saved the session with the same name i.e blabla and running.



```
[s]tatus [p]ause [r]esume [b]ypass [q]uit =>

Session.Name...: blabla
Status.........: Running
Input.Mode.....: Mask (Hello?d?l?d?u123?l?l?u) [15]
Hash.Target....: LamLEX (90:84:0d:a8:aa:e8 <-> c4:3d:c7:4f:98:93)
Hash.Type......: WPA/WPA2
Time.Started...: Sat Oct 31 02:07:39 2015 (3 secs)
Time.Estimated.: Tue Nov 03 12:38:15 2015 (3 days, 10 hours)
Speed.GPU.#1...:     6021 H/s
Recovered......: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.......: 16384/1188137600 (0.00%)
Rejected.......: 0/16384 (0.00%)
Restore.Point..: 16384/1188137600 (0.00%)
HWMon.GPU.#1...: 98% Util, 74c Temp, N/A Fan

[s]tatus [p]ause [r]esume [b]ypass [q]uit =>
```

Now you can simply press [**q**] close cmd, ShutDown System, comeback after a holiday and turn on the system and resume the session. That easy!

NOTE: Once execution is completed session will be deleted.

How to restore ?

Above command – "–restore". Here it goes:

```
cudaHashcat64.exe -m 2500 rootsh3ll-01.hccapx -a 3 Hello?d?l?d?u123?l?l?u --session=blabla --
```

Hashcat will now check in its working directory for any session previously created and simply resume the Cracking process.

Simple enough ? Yes it is.

This is all for Hashcat. Hope you understand it well and performed it along. No need to be sad if you don't have enough money to purchase those expensive Graphics cards for this purpose you can still try cracking the passwords at high speeds using the clouds. You just have to pay accordingly.

Cloud for Cracking WPA2-PSK

You can even leverage cloud for the same purpose. You just have to pay for the service you use as it requires a lot of money, electricity to keep the system up and running and keeping it fast at the same time.

A Website that provide the similar service is http://cloudcracker.com/ (Discontinued)

They charge $17 for 300 Million words in 20 minutes. Which means 250,000 PMK/Second. Sounds nice! isnt it ?

Well this is a service so they surely have their part of profit. If you are at a shortage of money you can try even cheaper service.

Don't worry this cheap is actually better than the expensive if you are able to do it accordingly.

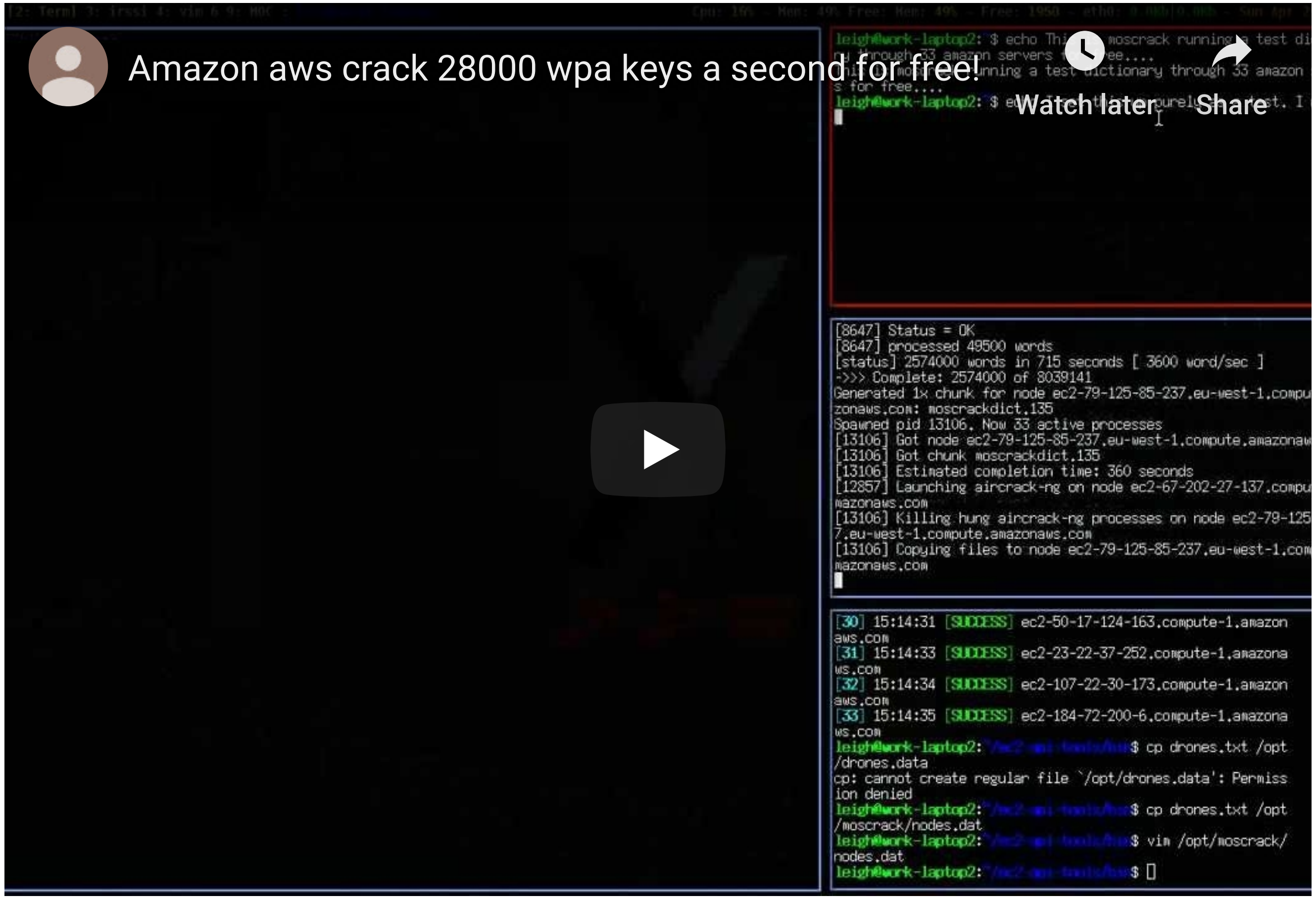
That is Amazon Elastic Computing 2(EC2) or AWS (Amazon Web Services). Here you need to do all the things manually after logging into the remote host that yo0u have purchased.

You have to install the tools and dependencies accordingly and give commands to the master server to perform the cracking. You can also create multiple instances to distribute the load and raise the WPA2 cracking speed. Price will change accordingly.

But in short let me tell you if you are willing to do this Super Interesting stuff, it will cost you maximum of $1 an hour for even greater speeds than cloudcracker.

Here is a video to help you understand better the concept of load distribution and command the master server. Hope you are getting the concept.

Here is one more for you to see the WPA2 cracking process running on Amazon EC2, It’s an old video but worth watch and understand the concept.



```
[8647] Status = OK
[8647] processed 49500 words
[status] 2574000 words in 715 seconds [ 3600 word/sec ]
->>> Complete: 2574000 of 8039141
Generated 1x chunk for node ec2-79-125-85-237.eu-west-1.compu
zonaws.com: moscrackdict.135
Spawned pid 13106. Now 33 active processes
[13106] Got node ec2-79-125-85-237.eu-west-1.compute.amazonaw
[13106] Got chunk moscrackdict.135
[13106] Estimated completion time: 360 seconds
[12857] Launching aircrack-ng on node ec2-67-202-27-137.compu
mazonaws.com
[13106] Killing hung aircrack-ng processes on node ec2-79-125
7.eu-west-1.compute.amazonaws.com
[13106] Copying files to node ec2-79-125-85-237.eu-west-1.com
mazonaws.com
```

```
[30] 15:14:31 [SUCCESS] ec2-50-17-124-163.compute-1.amazon
aws.com
[31] 15:14:33 [SUCCESS] ec2-23-22-37-252.compute-1.amazona
ws.com
[32] 15:14:34 [SUCCESS] ec2-107-22-30-173.compute-1.amazon
aws.com
[33] 15:14:35 [SUCCESS] ec2-184-72-200-6.compute-1.amazona
ws.com
leigh@work-laptop2:[illegible]$ cp drones.txt /opt
/drones.data
cp: cannot create regular file `/opt/drones.data': Permiss
ion denied
leigh@work-laptop2:[illegible]$ cp drones.txt /opt
/moscrack/nodes.dat
leigh@work-laptop2:[illegible]$ vim /opt/moscrack/
nodes.dat
leigh@work-laptop2:[illegible]$ 
```

Forgot to tell you one good news. Amazon EC2 is FREE for first year. It will just ask you for the credit/debit card info as a validation proof. But don't worry no extra penny will be deducted until you extend to new plan.

So I would encourage you to do some research on this specific topic after getting over of Hashcat. It is the real Fun believe me!

If you love all this crazy stuff You will love that too.

I discuss all of them in my free WiFi pentesting and Security eBook. Follow the link below to learn more

Download All 10 Chapters of WiFi Pentesting and Security Book...

READ DESCRIPTION

PDF version contains all of the content and resources found in the web-based guide

Hope this was helpful enough!

Keep Learning.

Hashcat Rootsh3ll Wi-Fi Security and Pentesting Series rwsps Wi-Fi pentesting windows

24 Comments rootsh3ll.com 1 Login

♡ Recommend Tweet Share Sort by Newest

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Donna R. • 3 years ago

What is the correct syntax for a hashcat dictionary attack while *NOT* using my GPU? (I'm trying to NOT mess up yet another machine trying to force-install non-compatible GPU drivers)

drivers.)

```
# Only use my CPU:
> hashcat -m 2500 my.hccap mydict.txt
```

Always fails:

```
hashcat (v3.10) starting...
OpenCL Platform #1: Mesa, skipped! No OpenCL compatible devices found
ERROR: No devices found/left
```

2 ⌃ | ⌄ • Reply • Share ›

Hardeep Singh Mod ➦ Donna R. • 3 years ago

You should use Hashcat Legacy. it uses CPU only for cracking and does not require additional drivers.

for more help have a look at **GPU device not found, why?** section of the link: https://hashcat.net/wiki/do...

⌃ | ⌄ • Reply • Share ›

Almasihu akbar • 3 years ago

Hashcat sounds good , they should of made it easier to use , like aircrack. I cant find the cracked password file. I tried cat hashcat.pot
No directory

⌃ | ⌄ • Reply • Share ›

Hardeep Singh Mod ➦ Almasihu akbar • 3 years ago

aircrack is dedicated for WEP/WPA/2 password cracking only, so comes with minimal options whereas Hashcat provides almost every encryption method for cracking with so much of flexibility. I think it is already easiest possible cracker, considering the power and flexibility it gives.

For listing cracked password try `ls *.pot`

⌃ | ⌄ • Reply • Share ›

Almasihu akbar ➦ Hardeep Singh • 3 years ago

Thanks , I've just started using hashcat today -all day..Just got my video card and wow hashcat runs fast with gpu..

One question bro..
Is their a way of knowing the password lenght ..i opened a cap file with wireshark. And i was browsing through the file and saw (key) lenght 16 digits.
Im not sure ..
Any input would be appreciated.

^ | ∨ • Reply • Share ›

Hardeep Singh Mod ➦ Almasihu akbar • 3 years ago • edited

Nope. There's no way to get the key length. it is because the PBKDF2 function in the WPA2 algorithm always comes out with a fixed size PSK that's 32 bytes (256 bits) and often displayed as 64 hex characters.

For better clarity you can use `wpa_passphrase` utility.
Syntax: `wpa_passphrase <ssid> <passphrase>`
Output:

```
network={
ssid="rootsh3ll"
#psk="iamrootsh3ll"
psk=1f4b02fe4c82f4e0262e6097e7bad1f19283b6687f084f73331db86c62498b4
}
```

Notice the PSK in output. It is a 64 HexaDecimal string. As 1 HexaDecimal represents a nibble(4 bits), 64 hex characters represents 256-bits of PSK.

The passphrase must lie within range of 8-63 characters, as per WPA2 requirement. No matter passphrase is 8 or 63 characters the PSK will always be 256 bit key.
Which simply does not allow us to check the length of the passphrase

Which simply does not allow us to check the length of the passphrase from the Hash

NOTE: Encrypted string can be decrypted, Encoded string can be decoded, But a Hash can never be de-hashed. It's Irreversible.

^ | ⌄ • Reply • Share ›

Almasihu akbar ➦ Hardeep Singh • 3 years ago

Thanks brother , for your time.

Can you help me with a command..

I want to run this on hashcat
Example: crunch 12 12 0123456789abcdefghijklmnopqrtuvwxyz

Im testing a att router which has a 12 length password
Random letters and numbers..
How can i run a similar command on hashcat without creating a huge wordlist

^ | ⌄ • Reply • Share ›

Hardeep Singh Mod ➦ Almasihu akbar • 3 years ago • edited

Just replace the numbers with a **%** and alphabets with **@**
Command:

```
crunch 12 12 -t %%%%%%@@@@@@
```

Sample output:
000000aaaaaa
000000aaaaab
000000aaaaac
000000aaaaad
000000aaaaae
000000aaaaaf

NOTE: Make sure total characters in the -t arguments lie within the

provided range.. i.e 12 in this case. Ofcourse you can make combinations like:
%%@@@@@@@@@@ = 00aaaaaaaaaa
%%%%%@@@@@@@ = 00000aaaaaaa
%%%@@@%%%@@@ = 000aaa000aaa

Have fun ;)

⌃ | ⌄ • Reply • Share ›

Eric • 3 years ago

Hi,
Thanks for a thorough tutorial.
I'd like to ask: If I want hashcat to try words beginning from 6 chars up to 8 chars (?a?a?a?a?a?a - ?a?a?a?a?a?a?a?a).
How to do that?

1 ⌃ | ⌄ • Reply • Share ›

Hardeep Singh Mod ➦ Eric • 3 years ago

Just add " **-i --increment-min=6 --increment-max=8** " at the very end of your command.
-i is to enable the increment switch and to automate the command for you to increase character testing limit.

Remember. WPA2 requires atleast 8 characters.

⌃ | ⌄ • Reply • Share ›

Eric ➦ Hardeep Singh • 3 years ago • edited

Thanks,
Yeah, I know WPA2 requires 8 chars minimum.
I just needed the exact syntax.

One more question:
If I want to test 8 up to 10 chars, do I need to write ten **?a** in the command?

If I want to test 8 up to 10 chars, do I need to write ten ?a in the command?

1 ^ | ⌄ • Reply • Share ›

Hardeep Singh Mod ➦ Eric • 3 years ago

All I can say is Test, Test, and Test!
;)

^ | ⌄ • Reply • Share ›

Michael • 3 years ago

Hi great read! But i just wanted to correct you on one of there commands for the Hi123 where you say the rest is lower case but you used ?u where is actually should be ?l

1 ^ | ⌄ • Reply • Share ›

Hardeep Singh Mod ➦ Michael • 3 years ago

Fixed!
Thanks for pointing out Michael. :)

^ | ⌄ • Reply • Share ›

paul_pjl • 3 years ago

Good write-up. Appreciate the thorough approach.

One comment/question. In all of my studying of the statistical probabilities of brute-forcing a WPA2 handshake, it seems that unless you are certain the passphrase contains specific characters, using a mask attack is of little value. In fact, from my own personal experience, there is little to no benefit in using wordlists or dictionaries whatsoever. I think I have been successful around 5% of the time when using dictionaries/wordlists. In one case a few years ago, I wasted significant time using a 27GB wordlist on a handshake without success.

This has led me to the belief that for purposes of efficiency, and to avoid duplicative efforts with multiple wordlists (which may contain common words between them), the most reliable method is to just pipe a wordlist from Crunch...For instance, a Crunch generated list of all words containing 8 characters (say lower-case alpha and numeric). I realize this is a very large wordlist, but through the use of AWS high performance AMIs, it can be done.

That said, if using simply a Crunch generated list, do you still believe oclHashcat performs better than Pyrit-Cuda? I have not used Hashcat, but assumed that other than the mask-attack or other word-manipulation features, there is no difference in speed, and further, pyrit has the database functionality where the PMKs can be pre-generated. Do you have experience showing hashcat is indeed faster than Pyrit for just plain brute-forcing/testing of each hash against the handshake?

1 ^ | ˅ • Reply • Share ›

Hardeep Singh Mod ➦ paul_pjl • 3 years ago

Hey Paul,

Thanks for the appreciation Paul!

Talking about the Wordlist/dictionary based attack, they are most likely to be successful when combined with a well conducted Social Engineering and Reconnaissance.
In simple terms you just have to do the right thing to get the job done and to know that right thing you can use social engineering to get your direction.
Although success isn't guaranteed but chances increases by multiple folds.

I cracked a WPA2 of an Institute by creating a wordlist adding possible combinations of the name of the Institute itself, some numbers like 1234567890, 1..9, 098..1 etc and the password was cracked within seconds as wordlist was ridiculously small and the password was written on their board itself. It was a mobile number.
It is definitely not the best way(none is). But can be very helpful and save a lot of money,resources and of course time. Even if you are on AWS AMIs.

As you told using any random dictionary is mostly a waste of time, effort and money. Agree!

see more

^ | ˅ • Reply • Share ›

CodersSquads • 3 years ago

CodersSquads • 3 years ago

Nice tutorial.

I guess I've found a mistake. You say that -a 1 is the parameter for hybrid attack, but Hashcat documentation says it's -a 6. In my tests with oclhashcat, -a 6 is the correct parameter.

^ | ˅ • Reply • Share ›

Hardeep Singh Mod ➦ CodersSquads • 3 years ago

Nice Found. Fixed!

^ | ˅ • Reply • Share ›

shanky • 4 years ago

what will be the command for hash cat
case 1-total 8 characters with any combination
case2-total 9 characters with any combination
case 3-total 10 characters with any combination

^ | ˅ • Reply • Share ›

rootsh3ll ➦ shanky • 4 years ago

cudaHashcat64.exe -m 2500 <handshake.hccap> -a 3 ?a?a?a?a?a?a?a?a # For 8 random chars.

Simply put 9 times **?a** for 9 chars and 10 times **?a** for 10 random characters.

Ex: **cudaHashcat64.exe -m 2500 <handshake.hccap> -a 3 ?a?a?a?a?a?a?a?a?a**

^ | ˅ • Reply • Share ›

shanky • 4 years ago

I ran hashcat in windows 7 64 bit with rockyou.txt for 5 hours . In the end no password shown just starting and stooping time .What should I do next ?

^ | ˅ • Reply • Share ›

rootsh3ll ➦ shanky • 4 years ago

That means that rockyou.txt doesn't contains the password used to secure the AP. You should try Mask attack next which will cover the phone numbers. As many people use their mobile numbers as the WPA2 passwords it is worth giving a try.
Not only it will be completed fairly quick and also you can lately jump on to next dictionary with a confirmation that no mobile number is being used as the passphrase.

1 ^ | ⌄ • Reply • Share ›

amback • 4 years ago

is there a way i can calculate how many h/s or pks i can get on a gpu before buying one?, i heard you "sort of" can using the core count, and the Clock Speed but have found nothing. for example say i want to get the amd fury x and before buying it want to calculate how many psk/s i can get on aircrack just by using the gpu specification. is it possible?

also very nice guide and thank you

^ | ⌄ • Reply • Share ›

rootsh3ll ➦ amback • 4 years ago

Thanks Amback!
You can simply search google with this query: "[Graphics card name] hashcat benchmark" and this is what I got for AMD Fury X.

But you might like to have a look at Jeremi M Gosney's Post on NVidia Titan X cudaHashcat Benchmarks. Of course it is +**$350** expensive than the AMD Fury X but it outperform AMD's Fastest Graphics Card by a staggering +60% of WPA2 cracking speed and +85% when overclocked and that's a big deal on a Single GPU which makes it the best card for WPA2 cracking at this moment.
Benchmarks goes like,
WPA2 Cracking speed for:
AMD Fury X: ~**163,100** Keys/Second
NVidia Titan X: ~**279,200** Keys/Second

If you are on a budget of $**999**, I would definitely suggest you to purchase NVidia Titan X for WPA2 cracking purposes. Why ? because it gives you WPA2 cracking speed

X for WPA2 cracking purposes. Why ? because it gives you WPA2 cracking speed almost equals to AMD's Top 2 graphics card combined. Here's how

WPA2 cracking speed for Fury X + Radeon HD 6990 i.e **163100 + 154300 = 317400**

see more

^ | ⌄ • Reply • Share ›

✉ Subscribe   Add Disqus to your site   Disqus' Privacy Policy   DISQUS