

# **USING THE POWER OF GPU TO CRACK COMPLEX PASSWORDS**

## Introduction

In an era of Information technology, the cyber-attacks are real threat to the organization, where an attacker can access the server, application, etc. and can gain full access to the environment by just entering the simple password. The password is what that separates the authenticated user and an attacker (un-authenticated user).

What if the password that is used in your application can easily be cracked.??!

Even if you have used some sort of Encryption method to encrypt your password, do u still think your password is safe...??!

Let's take a look at it.

# Case Scenario 1

## Cracking the MD5 encrypted Password without GPU (With CPU).

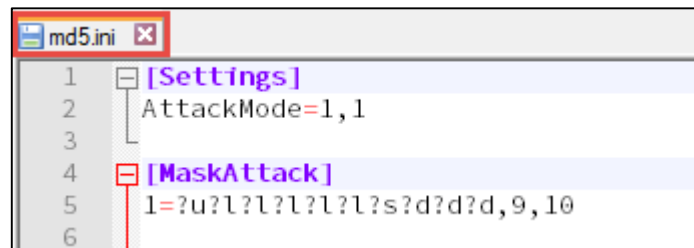
For the cracking purpose, I am using the tool HM (Hash Manager) from Insidepro. It's the fastest password cracking tool for Windows that doesn't support GPU, for more details about the supported algorithms you can visit the site <http://insidepro.com/>.

Hash manager uses the cores present in CPU to crack the encrypted password.

```
Hash Manager v1.2.3 (64-bit), (c)2014-2015 InsidePro Software
Usage: HM <Algorithm> <Configuration File> <Hash List>
```

Before proceeding with password cracking, let's take a look at the arguments which need to be used with this tool.

- 1) **Algorithm** – specifies the encryption algorithm of the passwords which need to be cracked.  
Example: "MD5", "SHA512", "MD5(MD5(\$PASS))".
- 2) **Configuration File**- Name of the INI file with the attack settings to be used during the attack.  
User has to create an INI file which contains attack settings.



**AttackMode** – Types of attack you can perform, there are 4 attack modes available in it.

1 - (brute-force attack), 2 - (mask attack), 3 - (dictionary attack) or 4 - (hybrid attack).

In case you are selecting the 2<sup>nd</sup> option you need to specify the settings for it as well.

**Attack Number** = Character set, Minimum password length, Maximum password length.

**Example settings:**

[MaskAttack]

1=?d,1,12

2=?l?u?d,4,6

3=0123456789abcdef,2,8

**Note: All attacks support the following standard character sets:**

?d – 0123456789

?l – abcdefghijklmnopqrstuvwxyz

?u – ABCDEFGHIJKLMNOPQRSTUVWXYZ

?s – !@#\$%^&\*()~\_-+=\|[]{};:'",.<>/?

So, in the above “ini” file I am using 10-character long password which contains 1 Capital letter alphabet, 4 small letter alphabet, 1 special character and 3 numeric digits.

3) **Hash List**- name of the file that contains the encrypted password.

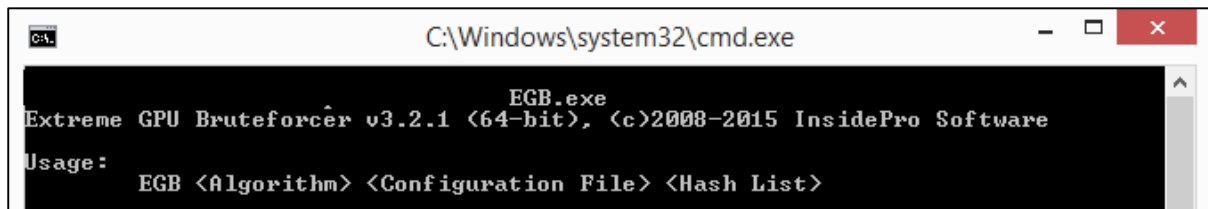
## Case Scenario 2

### Cracking the MD5 encrypted Password with GPU.

#### What is GPU?

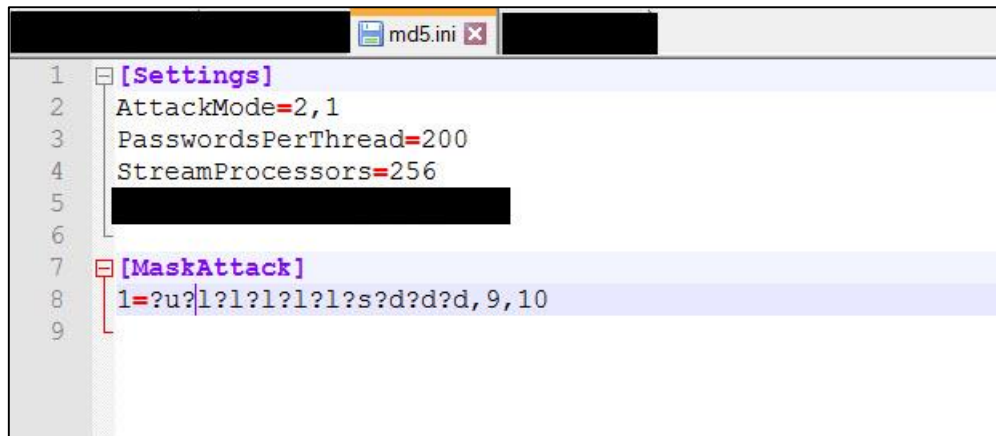
GPU stands for Graphics Processing Unit, it is widely used for 3D Games rendering, but now with **CUDA** technology enabled on some graphics card it allows software developers to use a **CUDA**-enabled graphics processing unit (**GPU**) for general purpose.

For password cracking using GPU, we are going to use EGB (Extreme GPU bruteforcer) which we can download from Insidepro website, the link is mentioned [here](#). EGB harnesses the Power of GPU cores and uses it to crack the password, it's one of the fastest brute forcing tool I have seen. Before moving forward with the cracking part let's take a look at the argument it requires.

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The command prompt displays the following text:

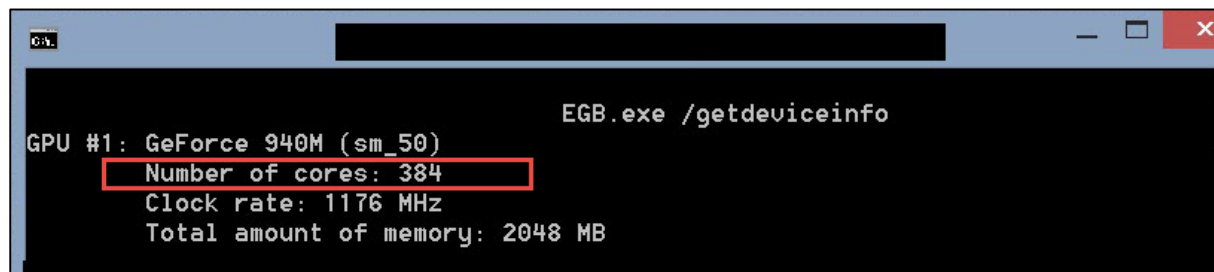
```
EGB.exe
Extreme GPU Bruteforcer v3.2.1 <64-bit>, <c>2008-2015 InsidePro Software
Usage:
    EGB <Algorithm> <Configuration File> <Hash List>
```

As we can see in the above figure the argument required in EGB is same as the argument required in the HM (Hash Manager) , but a slight difference in the INI file (Attack settings files).



```
1 [Settings]
2 AttackMode=2,1
3 PasswordsPerThread=200
4 StreamProcessors=256
5
6
7 [MaskAttack]
8 1=?u?l?l?l?l?s?d?d?d,9,10
9
```

**StreamProcessors** = No. of CUDA Cores in NVIDIA Graphics Cards



```
EGB.exe /getdeviceinfo
GPU #1: GeForce 940M (sm_50)
Number of cores: 384
Clock rate: 1176 MHz
Total amount of memory: 2048 MB
```

I am using NVIDIA GeForce 940M graphics card which has 384 cores in it.

By default, the value of StreamProcessors used in EGB is 128.

**PasswordsPerThread** – Number of passwords to be processed in one thread; by default - 3000 passwords for unsalted hashes, and 100 passwords for salted ones.

Example: -

**EGB.exe "MD5" MD5.ini Hashes.txt**

**NOTE: - If you are planning to buy the NVIDIA Graphics card for Password Cracking make sure to check, whether that graphics cards Support the CUDA technology.**

List of Graphics card (NVIDIA) that supports CUDA technology.

<https://developer.nvidia.com/cuda-gpus>

## Let's Move on to the main part that is Password Cracking.

The Password which we will crack is "**Abcedf@123**", it meets the complexity requirement as it has:

- 1 Capital letter
- 1 special character
- 3 numeric digits
- total length is 10 characters.

MD5 of **Abcedf@123** = "**3c2b6733705ad2e1ded0fcd79981c7c2**"

I am using the **Mask attack** for the password cracking as I have limited time and resource for it.

Mask attack is preferred over bruteforce attack as it requires less time to crack the password. All you need to have is an idea of how the password was set i.e. the complexity. In this case I knew that the 1<sup>st</sup> character was capital then 5 small letters followed by 1 special and 3 numeric digits respectively.

Our main goal over here is to see the time required to crack the password even if password meets the complexity required.

### 1) Without GPU

**Processor:** - Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz (4 CPUs), ~1.7GHz

**RAM:** - 6064MB RAM

**OS:** - Windows 8.1 Pro 64-bit

Let's start then.

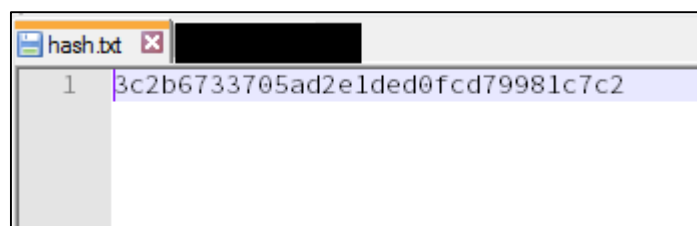
**Steps to Reproduce:** -

- 1) Create INI File.



```
1 [Settings]
2 AttackMode=2,1
3
4 [MaskAttack]
5 1=?u?l?l?l?l?s?d?d?d,9,10
6
```

- 2) Create Hash List



```
1 3c2b6733705ad2e1ded0fcd79981c7c2
```

3) Supply all the arguments to the application, and run it.

```
C:\Windows\system32\cmd.exe - HM.exe "MD5" md5.ini hash.txt
HM.exe "MD5" md5.ini hash.txt
Algorithm: MD5 (AUX2)
Hashes loaded: 1
Threads runned: 4
Passwords found: 0 : 31.1M p/s : ?u?l?l?l?l?s?d?d,9,10 <0.011%> : 04:04:10:
```

As we can see in above figure the HM (Hash Manager) is brute forcing the password at the speed of 31 million passwords per sec, which is excellent as it's doesn't have any GPU.

But, on the downside, it is consuming resources of the CPU. It consumes 90% of CPU usage, so in order to crack the password, you need to close all the other processes.

Name	Status	CPU	Memory	Disk	Network
Background processes (46)					
		0%	0.4 MB	0 MB/s	0 Mbps
		0%	3.7 MB	0 MB/s	0 Mbps
		0%	2.8 MB	0 MB/s	0 Mbps
		0%	11.5 MB	0 MB/s	0 Mbps
		0%	2.0 MB	0 MB/s	0 Mbps
		0%	0.5 MB	0 MB/s	0 Mbps
		0%	0.1 MB	0 MB/s	0 Mbps
HM.exe		98.0%	0.3 MB	0 MB/s	0 Mbps
		0%	2.4 MB	0 MB/s	0 Mbps

So, I guess, If I am in a hurry to crack the password but I also don't have a GPU for it, I guess I need to rely on HM to crack it for me.

## 2) With GPU

**Processor:** - Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz (4 CPUs), ~2.2GHz

**RAM:** - 4018MB RAM

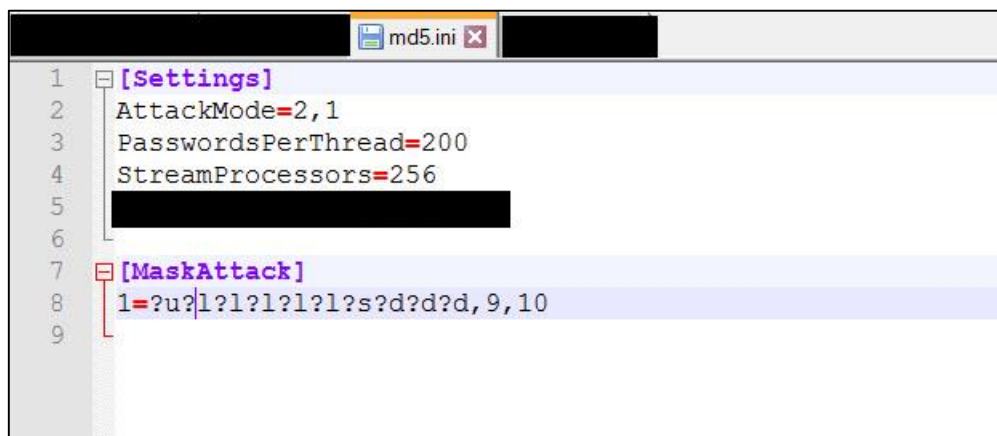
**OS:** - Windows 8.1 Single Language 64-bit (6.3, Build 9600)

**Graphics Card:** - 2048MB

Let's start then.

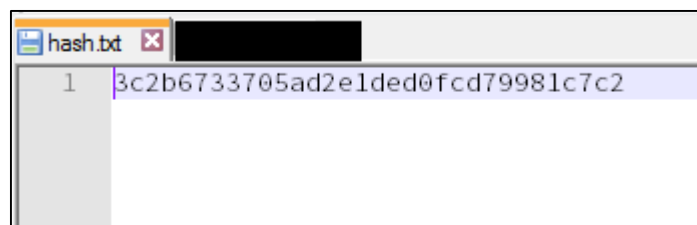
**Steps to Reproduce:** -

- 1) Create INI File.



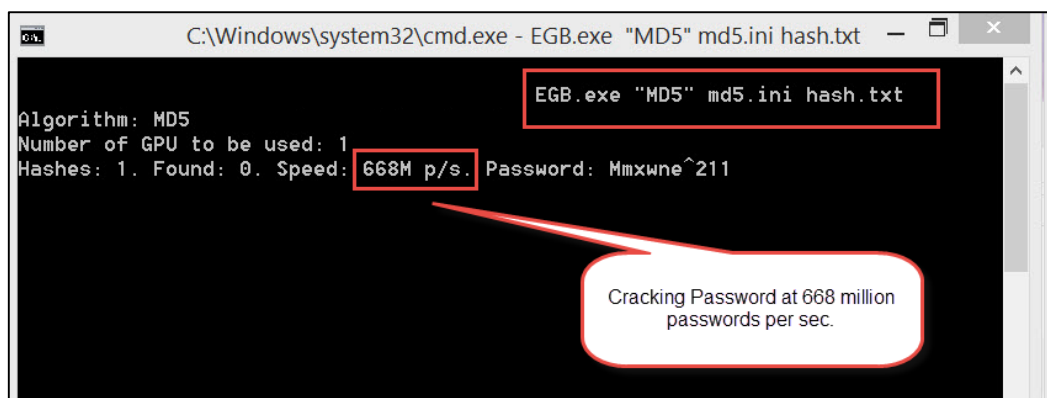
```
1 [Settings]
2 AttackMode=2,1
3 PasswordsPerThread=200
4 StreamProcessors=256
5
6
7 [MaskAttack]
8 1=?u?l?l?l?l?l?s?d?d?d,9,10
9
```

- 2) Create Hash List



```
1 3c2b6733705ad2e1ded0fcd79981c7c2
```

- 3) Supply all the arguments to the application, and run it.

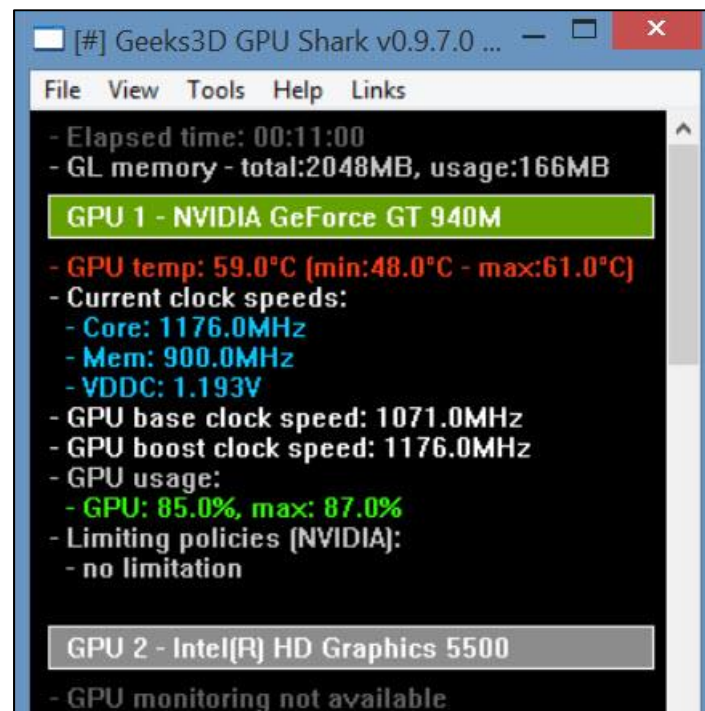


```
C:\Windows\system32\cmd.exe - EGB.exe "MD5" md5.ini hash.txt
EGB.exe "MD5" md5.ini hash.txt
Algorithm: MD5
Number of GPU to be used: 1
Hashes: 1. Found: 0. Speed: 668M p/s. Password: Mmxwne^211
```

Cracking Password at 668 million passwords per sec.



As we can see in above figure, the password is being cracked at an average speed of 668 million passwords per sec. This is when my Graphics Card is just having 384 Cores in it. You can imagine the power of [GeForce GTX TITAN X](#) graphics card which is having 3072 cores in it.



This Is a real time performance output of GPU when it is being used during password cracking.

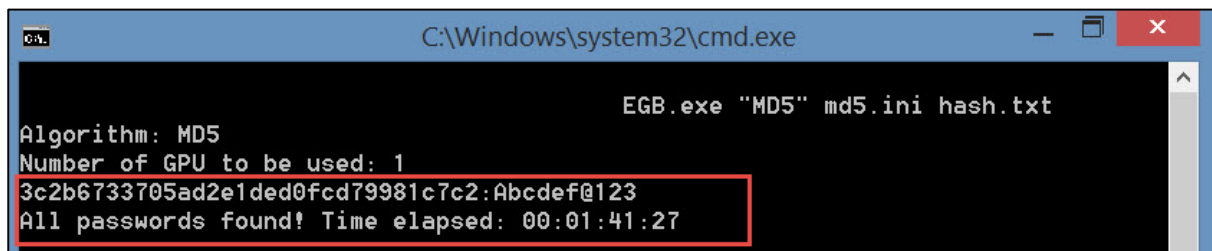
# Comparison

## Without GPU

As the processor was being over utilized and I was unable to perform any other actions on the system, I was forced to abort the password cracking process.

## With GPU

As we can see, with GPU, the password was cracked within 2 hours. Also, it offloaded the performance pressure from the CPU cores.



```
C:\Windows\system32\cmd.exe

EGB.exe "MD5" md5.ini hash.txt

Algorithm: MD5
Number of GPU to be used: 1
3c2b6733705ad2e1ded0fcd79981c7c2:Abcdef@123
All passwords found! Time elapsed: 00:01:41:27
```

The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe". The command "EGB.exe 'MD5' md5.ini hash.txt" has been executed. The output displays the algorithm as MD5, the number of GPUs used as 1, and the cracked password "3c2b6733705ad2e1ded0fcd79981c7c2:Abcdef@123". The final line indicates that all passwords were found and the time elapsed was 00:01:41:27. The password and the final status line are highlighted with a red rectangular box.

## TOOLS USED

**HM**

<http://insidepro.com/download/HM.zip>

**EGB**

<http://insidepro.com/download/EGB.zip>

## REFERENCE

<http://insidepro.com/>

<https://www.question-defense.com/2010/08/15/automated-password-cracking-use-oclhashcat-to-launch-a-fingerprint-attack>