Cloud Based Password Cracking

GPU-Based Attacks using someone else’s hardware

Traditional Password Cracking

Ten years ago, cracking passwords was difficult:

- Use a GPU cluster in an on-premise data center
- Use a video card on a desktop/laptop
- Use a CPU (yuck)

Most people had access to < 1024 processing cores, with 1024 being a huge, expensive number.

Password Cracking Today

Today, harnessing the power of cloud computing, it is possible to crack passwords on a **6144-core** cluster for **$2.60 per hour**.

Or if confidentiality is not paramount, hosted cloud based solutions exist which take care of all the technical parts of password cracking.

Hosted Solutions

Cloudcracker (https://cloudcracker.com)

Pros:

- Don’t have to build any sort of wordlist
- Dead simple
- No technical expertise required

Cons:

- Cleartext of hash matches are used and stored internally as per CloudCracker Terms of Service

Cloud solutions

With Amazon Web Services (AWS), it is possible spin up a 6144 core GPU cluster in about 30 seconds and run it for $2.60 an hour.

Pros:

- Far less likelihood of third party involvement with results
- Cheaper per hash
- More flexibility as instances can be spun up on demand

Cons:

- Technical expertise in systems administration / password cracking required.

AWS Instance Types

- The Beast - **g2.8xlarge**
  - Comes with 4 GPUs - each having 1536 cuda cores.
  - Each GPU has 4GB of VRAM
  - Limit of two running at any one time
  - 32 vCPUs + 32GB of RAM + 240 SSD storage
- Also available: **g2.2xlarge**
  - Comes with 1 GPU - which has 1536 cuda cores.
  - GPU has 4GB of VRAM
  - $0.65 an hour
  - 8 vCPUs + 15GB of RAM + 60GB SSD storage

Setup commands

1. `$ sudo apt-get update`
2. `$ sudo apt-get install -y build-essential dkms linux-source p7zip linux-headers-$(uname -r)`
3. `$ wget http://developer.download.nvidia.com/compute/cuda/7.5/Prod/local_installers/cuda_7.5.18_linux.run`
4. `$ chmod +x cuda_7.5.18_linux.run`
5. `$ sudo mv cuda_7.5.18_linux.run /mnt`
6. `$ sudo /mnt/cuda_7.5.18_linux.run`
7. `$ sudo apt-get install -y linux-image-extra-virtual`
8. `$ sudo reboot`
9. `$ sudo apt-get install -y linux-headers-$(uname -r)`

Setup Commands (Continued)

1.
   ```
   $ wget http://us.download.nvidia.com/XFree86/Linux-x86_64/340.93/NVIDIA-Linux-x86_64-340.93.run
   ```

2.
   ```
   $ chmod +x NVIDIA-Linux-x86_64-340.93.run
   ```

3.
   ```
   $ sudo ./NVIDIA-Linux-x86_64-340.93.run
   ```

4.
   ```
   $ wget http://hashcat.net/files/cudaHashcat-1.37.7z
   ```

5.
   ```
   $ p7zip -d cudaHashcat-1.37.7z
   ```

Hashcat vs oclHashcat

GPU-enabled version: oclHashcat - Linux/Windows:

http://hashcat.net/oclhashcat/

(Original) Non-GPU-enabled version: Hashcat - Linux/Windows/Mac OS X:

http://hashcat.net/hashcat/

oclHashcat supports up to 128 GPUs!

oclHashcat - Command parameters

```
$ ./cudaHashcat64.bin -m 100 -a 6 test-sha1.txt rockyou.txt ?a?a
```

1. `-m 100` = The type of hash to break
   a. 100 is SHA1
   b. 0 is MD5
2. `-a 6` = The type of attack to attempt
   a. 6 is a dictionary with mask attack
   b. 3 is a brute force with mask attack
3. `test-sha1.txt` = The file containing the hash input to break.
4. `rockyou.txt` = A wordlist to hash against
5. `?a?a` = A suffix mask to make variations on entries in the wordlist

And the fun begins

Seven character password with five lowercase alphabet characters and two lowercase / uppercase / numeric / special characters:

Hashcat

```
$ ./hashcat-cli64.app -m 100 -a 3 test-sha1.txt "?l?l?l?l?l?a?a"
```

~48 Minutes

---

oclHashcat - 6144 cores

```
$ ./cudaHashcat64.bin -m 100 -a 3 test-sha1.txt ?l?l?l?l?l?a?a
```

~46 seconds

Eight Character Password - Lower case

Eight character password with six lowercase alphabet characters and two lowercase/uppercase/numeric/special characters:

Hashcat

```
$ ./hashcat-cli64.app -m 100 -a 3 test-sha1.txt "?l?l?l?l?l?l?a?a"
```

~21 Hours

oclHashcat - 6144 cores

```
$ ./cudaHashcat64.bin -m 100 -a 3 test-sha1.txt ?l?l?l?l?l?l?a?a
```

~20 Minutes

Eight Character Passwords - Upper and lower

Eight character password with six lowercase/uppercase alphabet characters and two lowercase / uppercase / numeric / special characters:

Hashcat

```
$ ./hashcat-cli64.app -m 100 -a 3 test-sha1.txt -1 “?l?u”
"?1?1?1?1?1?1?a?a"
```

~56 Days

oclHashcat - 6144 cores

```
$ ./cudaHashcat64.bin -m 100 -a 3 test-sha1.txt -1 ?l?u ?1?1?1?1?1?1?a?a
```

~21 Hours

oclHashcat - Dictionary Attacks

oclHashcat can take a wordlist and a mask for mutations on that wordlist

```
$ ./cudaHashcat64.bin -m 100 -a 6 test-sha1.txt rockyou.txt ?a?a
```

Where `rockyou.txt` is a dictionary of 14,344,392 words

`?a?a` is a mask to try two characters after each word in the dictionary

oclHashcat can handle the load ***in under 2 minutes***.

`?a?a?a` = ***2 hours 50 minutes***

`?a?a?a?a` = ***11 days 11 hours***

oclHashcat - Benchmarks

Hashcat

```
$ ./hashcat-cli64.app -b
```

oclHashcat - 6144 cores

```
$ ./cudaHashcat64.bin -b
```

Questions?

- https://github.com/nstarke
- nick@alephvoid.com