Releasing the Kracken: Building and Using a GPU Password Cracker

Jonathan Fallone
About Me

Jonathan Fallone
Senior Pen Tester with Knowledge Consulting Group
jonathan.fallone@knowledgecg.com
@Shady_Wushu

Pen Tester, Gamer,
Overall Computer Nerd
Disclaimer

- I do not design or work on any of the software in this presentation
- I do not work for or with any GPU manufacturer
Why GPUs?

- CPU – Good at sequential calculations a few at a time
- GPU – Good at the same calculation (like hashing) done a thousand times at the same time
Why Do You Need a Password Cracker?

- For Pen Testers-
  - For hashes you can’t pass (shadow files, NetNTLM, etc.)
  - For password protected documents (new to Hashcat!)
- For Security Folks-
  - Password Auditing
  - Password Statistics for Security Training Programs
Our Old Password Cracker Kind of Looked Like This…

We Needed An Upgrade.
Some Considerations Before You Begin:

- Money, money, money…
  - What’s your budget
  - How fast do you need to go?
- Space for your monster
  - “Gaming” Style Desktop vs 4U Server
- Maintenance
  - Drivers, patches, new software versions
- Security
  - A system filled with client passwords…
Step 1: Containing the Monster

- Desktops
  - Far less expensive
  - Easier to get parts
  - Doesn’t hold as many cards – 4 max, 3 realistically

- Servers
  - Very Large (4U Normally)
  - Hold far more cards (4 to 8)
  - Very Expensive, but…
  - Often have redundancy built in
Step 2: Fill In the General Bits

- Processor
  - Don’t use anything too good
  - Just keeps the system running

- Memory
  - 8 to 16 GBs

- Hard Drives
  - Enough to hold wordlists
  - RAID 1 is nice, but not necessary
Step 3: (The Best Step...)

![GTX 780Ti Graphics Card](image-url)
What Do I Look At While Picking?

- Cores (Shader Units)
- Clock Speed
- Thermal Design Power (TDP)

But These Don’t Really Tell The Whole Story

- Cracking speed is based on number of instructions it takes to calculate a hash
- Different cards have different instruction sets available
- Different versions of software and different drivers take advantage of instructions in different ways
That Didn’t Help At All…What Do I Pick?

- Use previous benchmarks to estimate cracking speeds
  - Many people post benchmarks online
- If you have to watch a budget, balance cracking speed and cost to get the most for your money
  - Double the price doesn’t always equal double the speed
My Pick

- nVidia 900 Series
  - Bridged the gap to the Radeon cards in terms of speed
  - Low TDP
  - Better Parallelism
  - Better drivers
Other Items to Think About

- Cooling
- Reference Design Cards
  - Usually pull air directly from the back outside of the case rather than through the case interior
  - No overclocking (card wears faster)
- Power
  - Go for overkill – get the largest power supply you can get
Put It All Together And…

Release the kraken!

JOHNHASCHEEZBURGER.COM
Release.....the Kracken!
Software

- Linux (Ubuntu Server)
- SSH
- Video Drivers
- oclHashcat
- hashcat-utils
- Wordlists
Setting It All Up

- Install Linux with minimal options – only SSH if remote access is needed
- Ensure that the system is secured – long passwords, Public-Private Keys for SSH
- Do not use open source video drivers – use only drivers right from AMD or nVidia
KEEP CALM AND POWER ON
There are many, many different strategies and attacks

No one right way

My method:
- Not difficult
- Does not require a lot of work on behalf of the tester
- Since March 2014, cracked 67 percent of all hashes captured

Hashtype: NTLM
Workload: 1024 loops, 256 accel

- Speed.GPU.#1.: 4130.4 MH/s
- Speed.GPU.#2.: 4175.8 MH/s
- Speed.GPU.#3.: 4170.3 MH/s
- Speed.GPU.#4.: 4170.6 MH/s
- Speed.GPU.#*: 16647.1 MH/s

Hashtype: DCC, mscash
Workload: 1024 loops, 256 accel

- Speed.GPU.#1.: 1224.5 MH/s
- Speed.GPU.#2.: 1238.6 MH/s
- Speed.GPU.#3.: 1237.7 MH/s
- Speed.GPU.#4.: 1236.9 MH/s
- Speed.GPU.#*: 4937.8 MH/s
Efficient Cracking

- Begin with fast attacks
- Take advantage of the fact that most users are ignorant of what makes a strong password (or choose to ignore the rules!)
- Then use the passwords that you cracked to help crack others!
  - Users often follow similar patterns
  - The organization often requires certain rules that make passwords similar
- Even once you move to brute force, you can configure rules and statistics to make it more efficient
Step 1: Easy Brute Force

- Take care of all the passwords you can brute force in no time.

- `cudaHashcat64.bin -a 3 -m 1000 -i /path/to/hash ?a?a?a?a?a?a`
  - `-a`: attack type
  - `-m`: hash type
  - `-i`: Increment Mode – starts at 1 character, goes up through the length of the mask
  - `?a?a?a?a?a?a`: Mask of 6 characters, with the “all” character set in each position
Step 2: Username

- Many users *still* include their username in their password
- Modifying your username list with rules files give even more possibilities
- Use the list of usernames captured with the password
- `cudahashcat64.bin -a 0 -m 1000 --rules-file=rules/d3ad0ne.rule /path/to/hash /path/to/to/userlist`
Step 3: Dictionaries

- `cudaHashcat64.bin -a 0 -m 1000 --rules-file =rules/d3ad0ne.rule --loopback /path/to/_hashes /path/to/dictionaries/`

- Use a variety of dictionaries
  - Rockyou
  - English Dictionaries
  - Passphrase list
  - Numerous other lists

- Use rules files to extend the wordlists
  - These can greatly increase crack time
Step 4: The Fingerprint Attack

- Use the passwords already cracked and create every possible combination of characters up to 7 characters, which we will then combine, which creates wordlist of words 2 to 14 characters.
- This uses the expander tool, found in the hashcat-utils
  - `awk < hashcat.pot -F: '{print $2}' > outfile`
  - `expander.bin < /path/to/outfile > expanded.txt`
  - `cudaHashcat64.bin -a 1 -m 1000 /path/to/ hashes /path/to/expanded.txt /path/to/expanded.txt`
From There…

- Markov Attacks – statistically based brute force
- Custom Wordlists – Create new wordlists based on patterns or topics in the cracked password list
- Straight Brute Force – fast hashes like NTLM are highly susceptible
Our Cracking Stats Since March

<table>
<thead>
<tr>
<th></th>
<th>Total Hashes</th>
<th>Cracked</th>
<th>Uncracked</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTLM</td>
<td>25087</td>
<td>16912</td>
<td>8175</td>
<td>67.4</td>
</tr>
<tr>
<td>SHA1</td>
<td>10</td>
<td>6</td>
<td>4</td>
<td>60.0</td>
</tr>
<tr>
<td>NetNTLMv2</td>
<td>30</td>
<td>13</td>
<td>17</td>
<td>43.3</td>
</tr>
<tr>
<td>Salted MD5</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>50.0</td>
</tr>
<tr>
<td>bcrypt/Blowfish</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>100.0</td>
</tr>
<tr>
<td>MS Cache v2</td>
<td>188</td>
<td>44</td>
<td>144</td>
<td>23.4</td>
</tr>
<tr>
<td>MS Cache v1</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>100.0</td>
</tr>
</tbody>
</table>
Thanks to:

- Jens Steube aka Atom – Brilliant designer of oclHashcat
- Jeremi Gosney aka epixoip – Team Hashcat member and hardware guru, answers all my questions on the Hashcat forum
- Chris Duffy aka Funk and Wagnall – pushed me to create this presentation
- Andrew Whitaker aka The Godfather – advice, support, and letting me build the Kracken!