

Hack the Hash

Unveiling Password Cracking Strategies



* * * * *



Evan Hosinski

OnDefend | KrakenTech LLC

WHO AM I?

Evan Hosinski

- Tradecraft Engineer
- Pentester
- Social Engineering SME
- Password Hardening SME
- Malware Developer
- Educator
- Researcher
 - Phishing Infrastructure
 - Hashcracking
- Public Speaker



 ehosinski@krakensec.tech

 <https://krakensec.tech>

 <https://linkedin.com/in/evanhosinski>

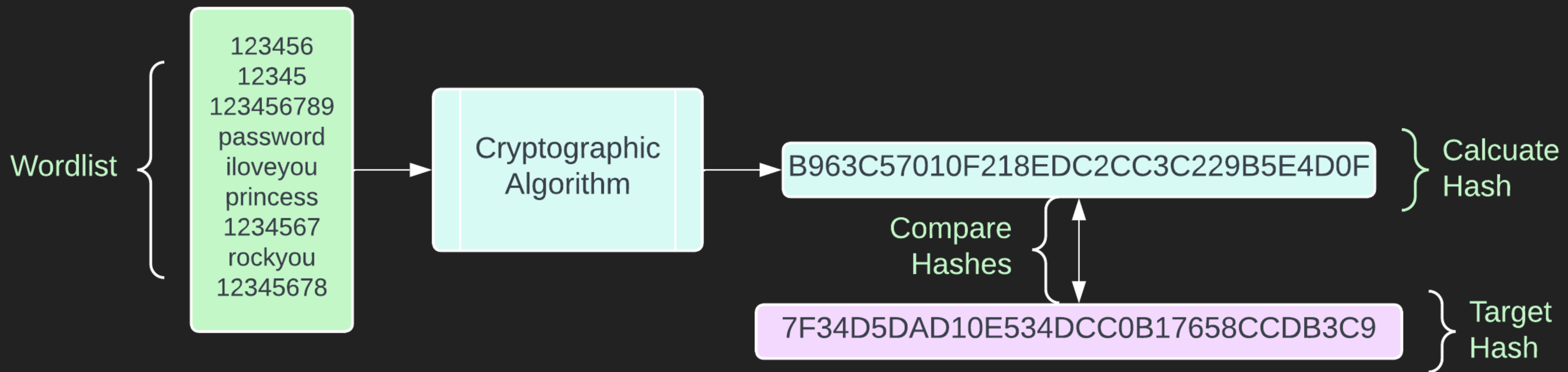
Common Terms

- **MH/s, GH/s, TH/s, H/s**
 - Million, Billion, Trillion hashes per second (Hashing Power)
- **Hash**
 - A fixed-length string created by a cryptographic function
- **Brute-force**
 - Attempting all possible combinations of characters
- **Wordlist**
 - A predefined list of password candidates
- **Mask Attack**
 - A customized brute-force attack with constraints
- **Pre-Processor**
 - Program that massages input data, optimizing it for more efficient, targeted attacks

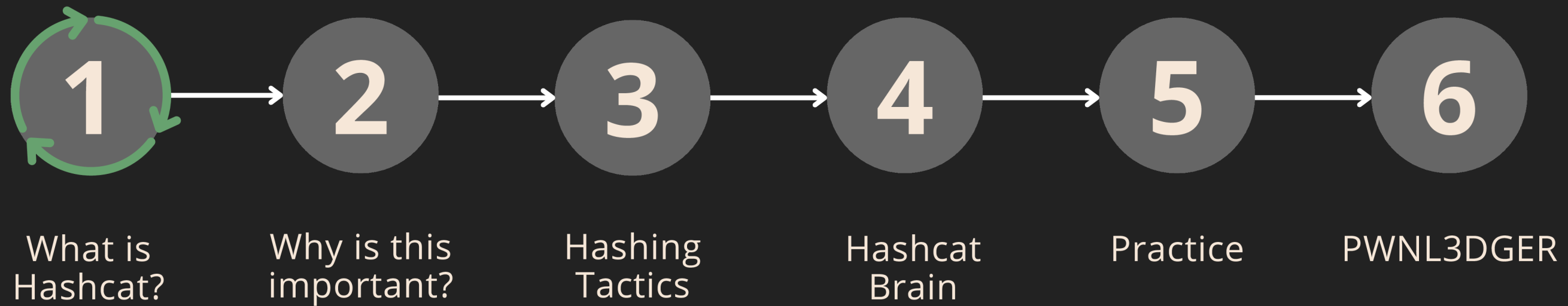




What is Hash Cracking?



OBJECTIVES





What is Hashcat?



Utilize GPU acceleration and CUDA Cores for faster hash cracking



Brain - Distributed password-cracking synchronization server



Common hashing algorithms



Brute force + masks + rules



Analyze results and refine cracking strategies





Flags



Flag	Description
-r/--rules-file	The location of the rules folder/file to use
-d/--backend-devices	The id(s) of the devices to use
-i, --increment-min, --increment-max	increment, minimum increment, maximum increment
--show/--identify	show all matches or show viable modes
--remove	remove matched hashes
--left	show remaining hashes
--session	Name the session (used for resuming)
-O	Optimize Kernel (limited password length ~15 characters)





Rules	
c	Capitalize the first letter
\$1	Append '1' to the end
^!	Prepend '!' to the front
c\$1	Capitalize the first letter, append '1'
^!c	Prepend '!', capitalize the first letter
c\$1d	Capitalize first letter, append '1', then duplicate the word

hashcat -a 0 -m 1000 hashes.txt wordlist.txt -r example.rule

grrcon
Grrcon
grrcon1
!grrcon
Grrcon1
!Grrcon
Grrcon1Grrcon1

Rules





Masks

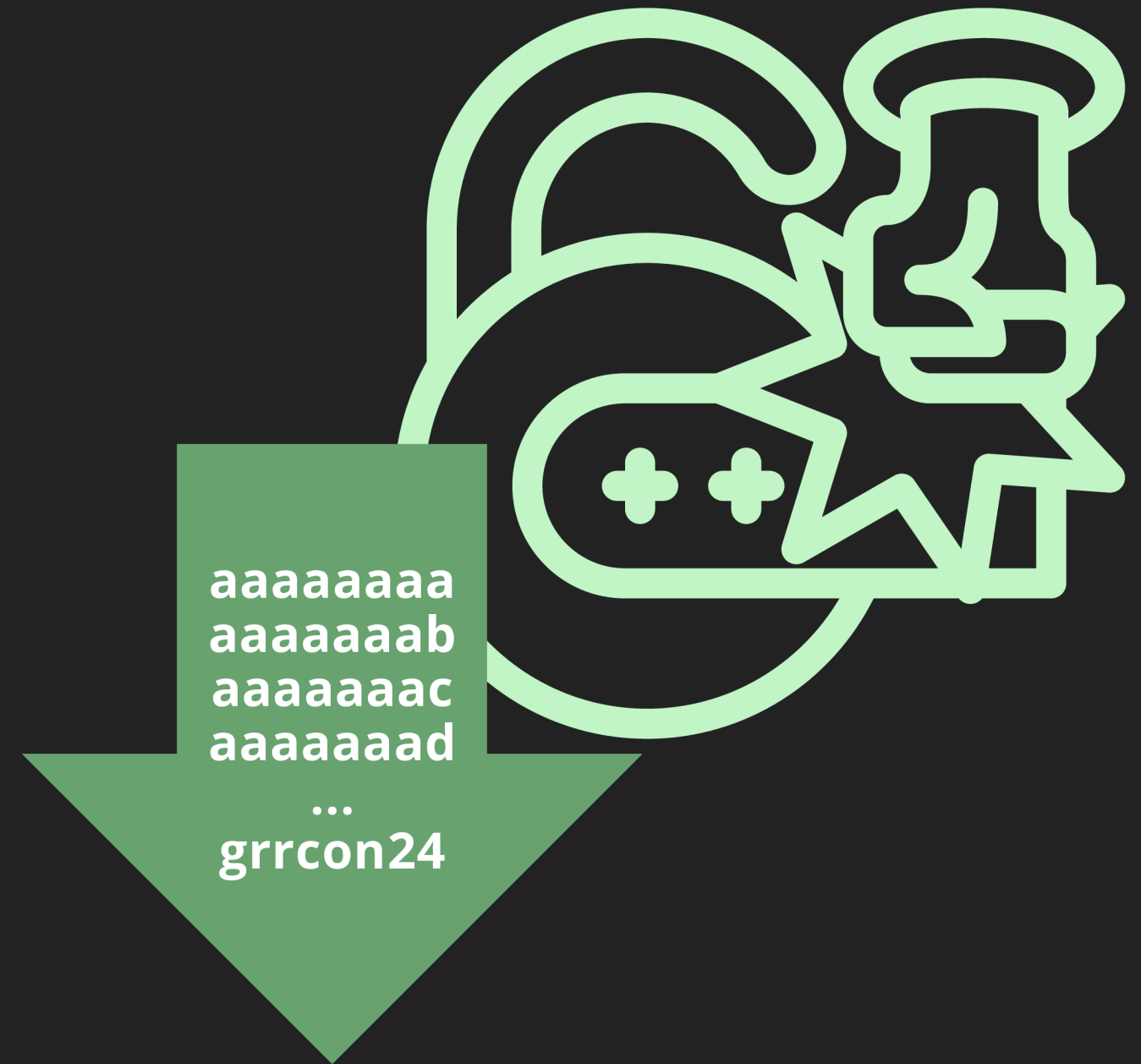
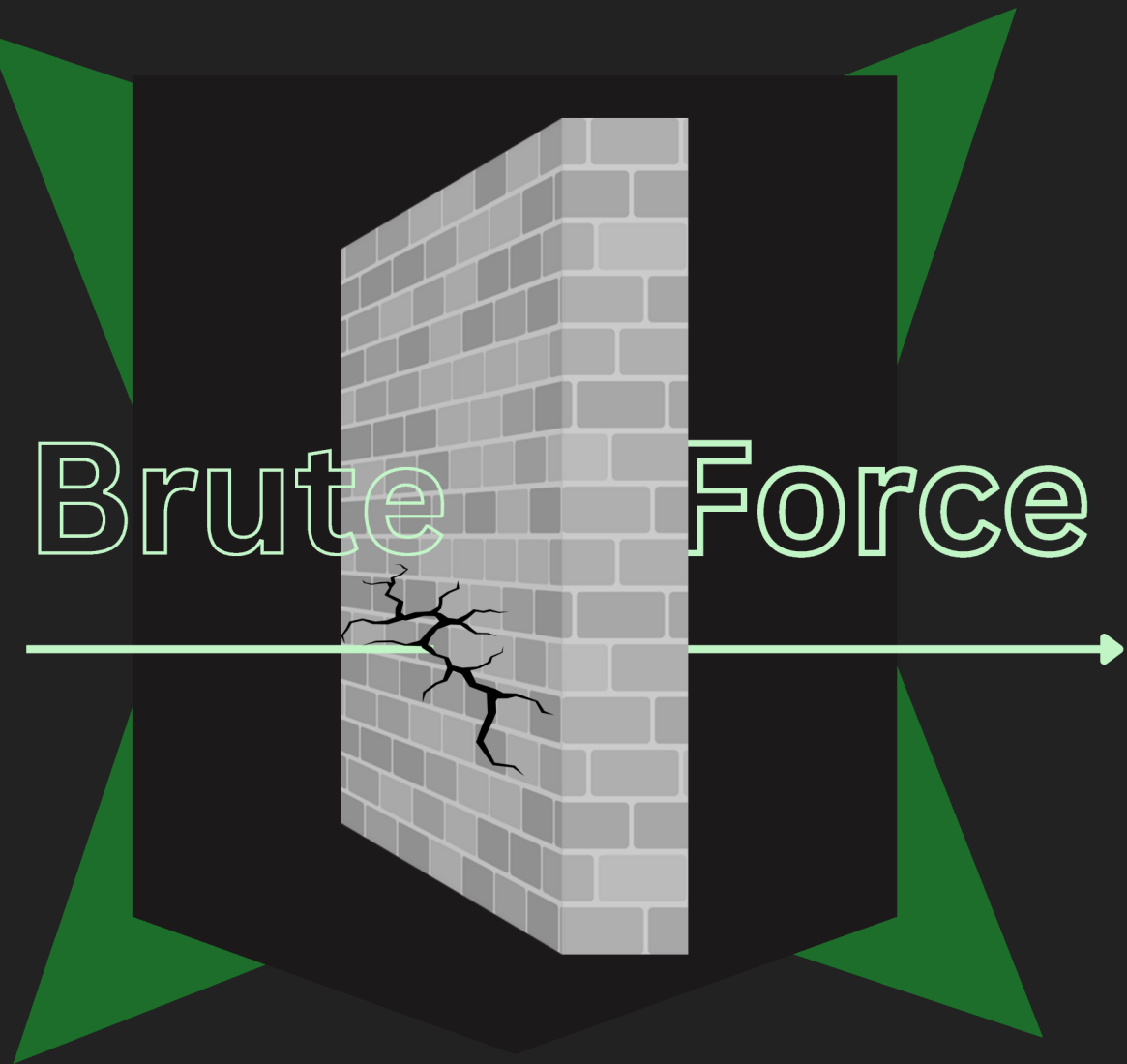
Masks	
?d	Digit
?u	Uppercase Letter
?l	Lowercase Letter
?s	Special character (e.g., !, @, #)

?u?l?l?l?l?
?l?l?l?l?l?d
grrcon?s
?u?l?l?l?l?d?d?s
grrcon?d?d?s

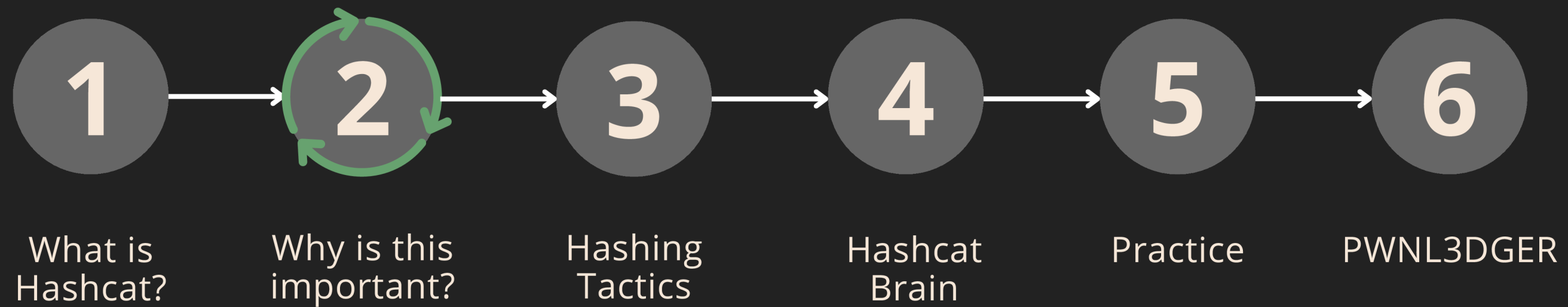
hashcat -a 3 -m 1000 hashes.txt grrcon?d?d

grrcon00
grrcon10
grrcon11
...
grrcon24





OBJECTIVES





Performance Increase

```
* Device #2: NVIDIA GeForce RTX 3090, skipped
* Device #1: NVIDIA GeForce RTX 3090, 23385/24147 MB, 82MCU

Benchmark relevant options:
=====
* --optimized-kernel-enable

* Hash-Mode 0 (MD5)
Speed.#1.....: 70538.3 MH/s (18.94ms) @ Accel:1024 Loops:512 Thr:256 Vec:1

* Hash-Mode 100 (SHA1)
Speed.#1.....: 22475.0 MH/s (60.43ms) @ Accel:64 Loops:1024 Thr:256 Vec:1

* Hash-Mode 1400 (SHA2-256)
Speed.#1.....: 9693.8 MH/s (70.26ms) @ Accel:8 Loops:1024 Thr:256 Vec:1

* Hash-Mode 1700 (SHA2-512)
Speed.#1.....: 3231.6 MH/s (52.56ms) @ Accel:4 Loops:1024 Thr:256 Vec:1

* Hash-Mode 22000 (WPA-PBKDF2-PMKID+EAPO) [Iterations: 4095]
Speed.#1.....: 1122.3 kH/s (73.36ms) @ Accel:8 Loops:1024 Thr:256 Vec:1

* Hash-Mode 1000 (NTLM)
Speed.#1.....: 127.3 GH/s (20.72ms) @ Accel:128 Loops:1024 Thr:256 Vec:1

* Hash-Mode 3000 (LM)
Speed.#1.....: 66070.5 MH/s (20.16ms) @ Accel:512 Loops:1024 Thr:256 Vec:1

* Hash-Mode 5500 (NetNTLMv1 / NetNTLMv1+ESS)
Speed.#1.....: 70206.1 MH/s (77.32ms) @ Accel:256 Loops:1024 Thr:256 Vec:2

* Hash-Mode 5600 (NetNTLMv2)
Speed.#1.....: 5028.3 MH/s (67.7ms) @ Accel:128 Loops:1024 Thr:256 Vec:1

* Hash-Mode 1500 (decrypt, DES (Unix))
Speed.#1.....: 2729.7 MH/s (62.2ms) @ Accel:128 Loops:1024 Thr:256 Vec:1

* Hash-Mode 500 (md5crypt, MD5 (Unix))
Speed.#1.....: 27335.0 kH/s (83.11ms) @ Accel:128 Loops:1000 Thr:256 Vec:1

* Hash-Mode 3200 (bcrypt $2*$, Blowfish (Unix)) [Iterations: 32]
Speed.#1.....: 83306 H/s (70.32ms) @ Accel:256 Loops:32 Thr:24 Vec:1
```

MD5: 122.72%

SHA1: 120.31%

NTLM: 115.63%

NetNTLMv2: 125.79%

bcrypt: 164.21%

```
* Device #1: NVIDIA GeForce RTX 4090, 23721/24114 MB, 128MCU
* Device #2: NVIDIA GeForce RTX 4090, 23724/24118 MB, 128MCU

* Hash-Mode 0 (MD5)
Speed.#1.....: 152.6 GH/s (13.74ms) @ Accel:256 Loops:1024 Thr:256 Vec:1
Speed.#2.....: 157.1 GH/s (13.32ms) @ Accel:256 Loops:1024 Thr:256 Vec:1
Speed.#*.....: 309.6 GH/s

* Hash-Mode 100 (SHA1)
Speed.#1.....: 48322.7 MH/s (43.91ms) @ Accel:32 Loops:1024 Thr:256 Vec:1
Speed.#2.....: 49514.1 MH/s (42.93ms) @ Accel:32 Loops:1024 Thr:256 Vec:1
Speed.#*.....: 97836.7 MH/s

* Hash-Mode 1400 (SHA2-256)
Speed.#1.....: 20514.9 MH/s (51.76ms) @ Accel:16 Loops:512 Thr:256 Vec:1
Speed.#2.....: 21473.9 MH/s (49.61ms) @ Accel:16 Loops:512 Thr:256 Vec:1
Speed.#*.....: 41988.8 MH/s

* Hash-Mode 1700 (SHA2-512)
Speed.#1.....: 7060.6 MH/s (75.49ms) @ Accel:8 Loops:1024 Thr:256 Vec:1
Speed.#2.....: 7297.6 MH/s (73.20ms) @ Accel:8 Loops:1024 Thr:256 Vec:1
Speed.#*.....: 14358.3 MH/s

* Hash-Mode 22000 (WPA-PBKDF2-PMKID+EAPO) [Iterations: 4095]
Speed.#1.....: 2461.1 kH/s (52.28ms) @ Accel:16 Loops:512 Thr:256 Vec:1
Speed.#2.....: 2525.5 kH/s (51.13ms) @ Accel:16 Loops:512 Thr:256 Vec:1
Speed.#*.....: 4986.6 kH/s

* Hash-Mode 1000 (NTLM)
Speed.#1.....: 273.5 GH/s (7.44ms) @ Accel:64 Loops:1024 Thr:256 Vec:1
Speed.#2.....: 274.5 GH/s (7.41ms) @ Accel:64 Loops:1024 Thr:256 Vec:1
Speed.#*.....: 548.0 GH/s

* Hash-Mode 3000 (LM)
Speed.#1.....: 142.6 GH/s (7.18ms) @ Accel:32 Loops:1024 Thr:256 Vec:1
Speed.#2.....: 146.3 GH/s (7.04ms) @ Accel:32 Loops:1024 Thr:256 Vec:1
Speed.#*.....: 288.9 GH/s

* Hash-Mode 5500 (NetNTLMv1 / NetNTLMv1+ESS)
Speed.#1.....: 153.4 GH/s (27.39ms) @ Accel:512 Loops:1024 Thr:64 Vec:2
Speed.#2.....: 155.8 GH/s (27.02ms) @ Accel:512 Loops:1024 Thr:64 Vec:2
Speed.#*.....: 309.2 GH/s

* Hash-Mode 5600 (NetNTLMv2)
Speed.#1.....: 10968.6 MH/s (97.45ms) @ Accel:8 Loops:1024 Thr:1024 Vec:1
Speed.#2.....: 11353.4 MH/s (94.04ms) @ Accel:8 Loops:1024 Thr:1024 Vec:1
Speed.#*.....: 22322.0 MH/s

* Hash-Mode 1500 (decrypt, DES (Unix))
Speed.#1.....: 6338.4 MH/s (62.2ms) @ Accel:128 Loops:1024 Thr:256 Vec:1
Speed.#2.....: 6509.2 MH/s (62.2ms) @ Accel:128 Loops:1024 Thr:256 Vec:1
Speed.#*.....: 12847.6 MH/s

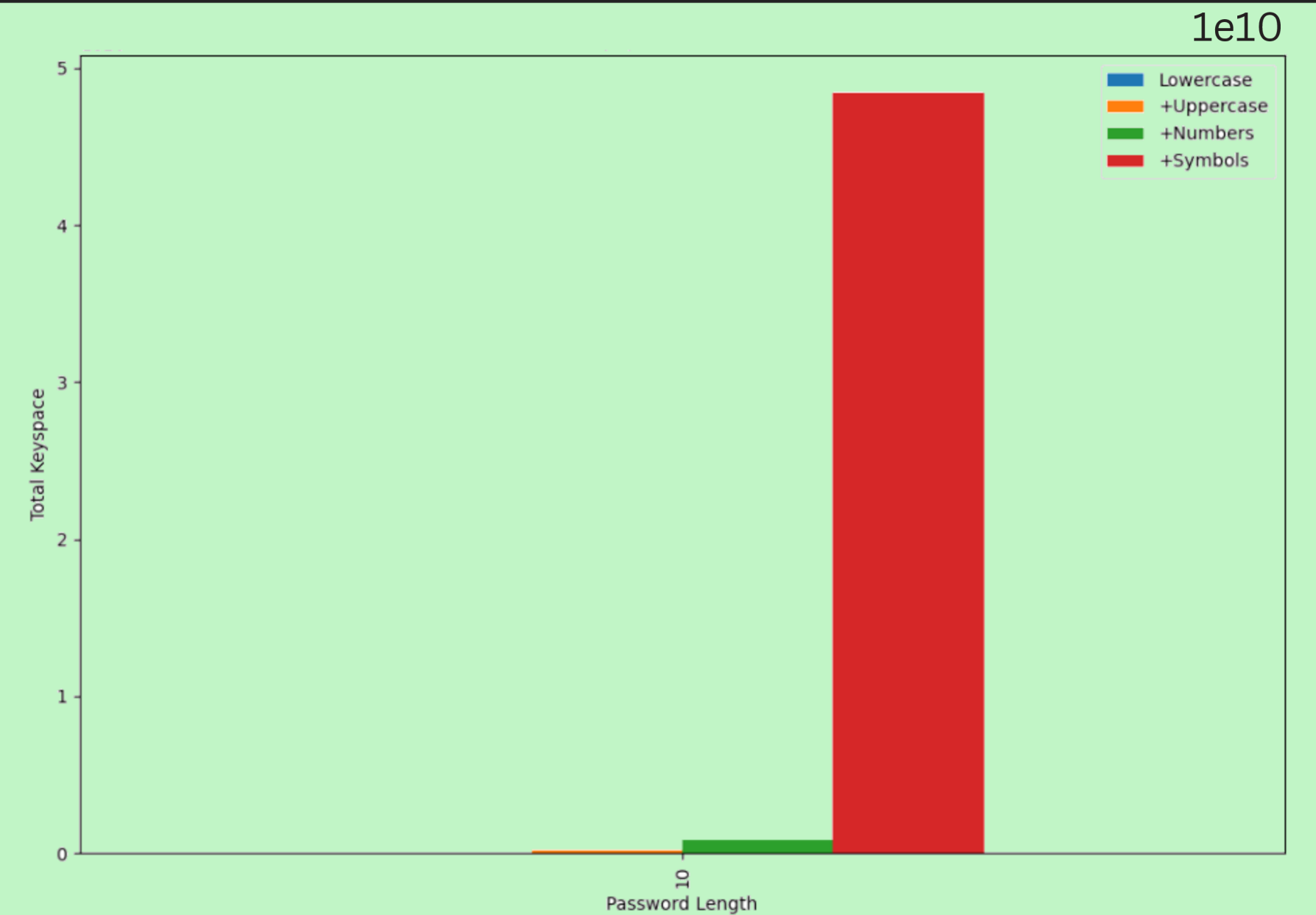
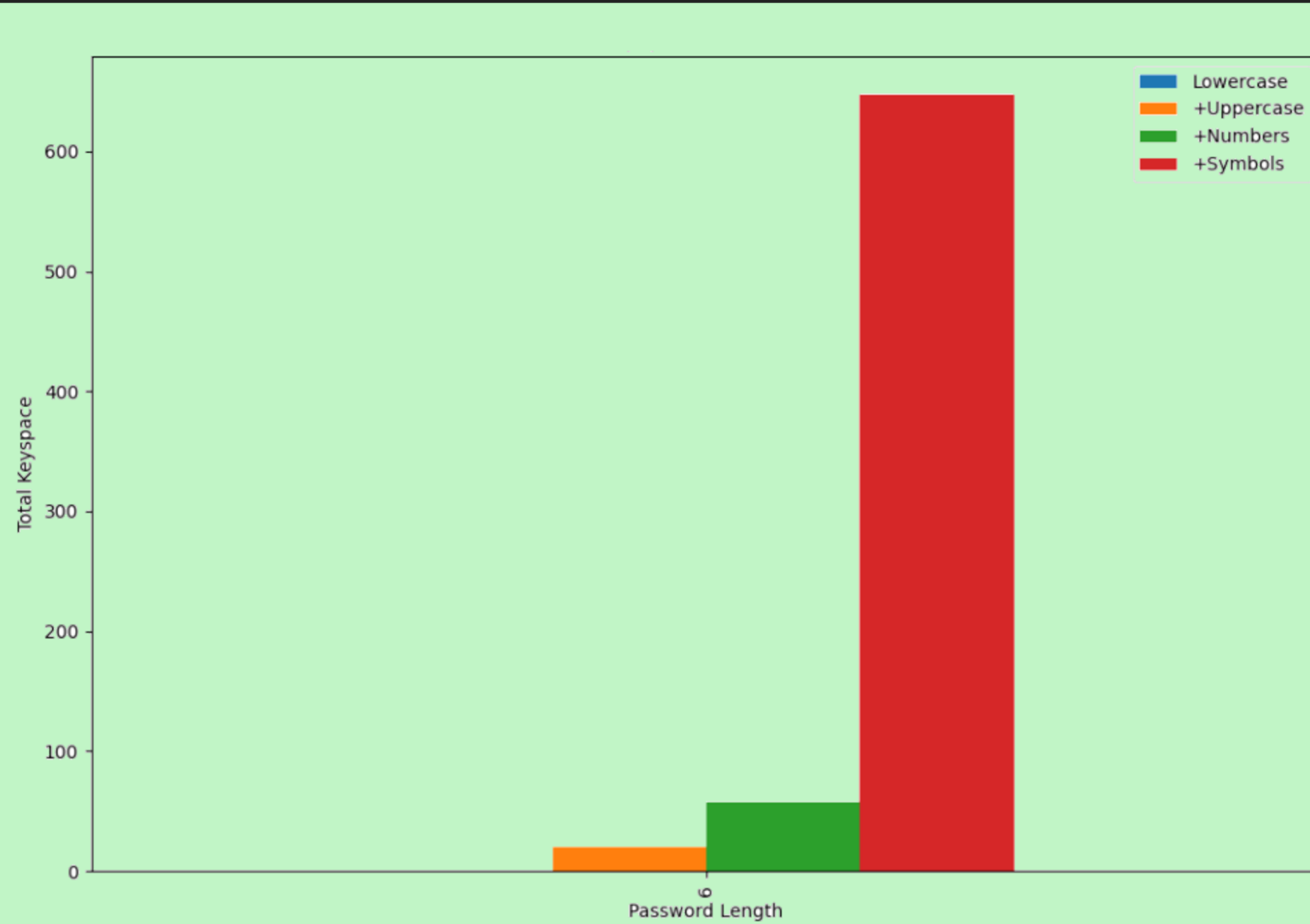
* Hash-Mode 500 (md5crypt, MD5 (Unix))
Speed.#1.....: 59278.1 kH/s (83.11ms) @ Accel:128 Loops:1000 Thr:256 Vec:1

* Hash-Mode 3200 (bcrypt $2*$, Blowfish (Unix))
Speed.#1.....: 220.1 kH/s (49.38ms) @ Accel:256 Loops:32 Thr:24 Vec:1
Speed.#2.....: 188.4 kH/s (49.12ms) @ Accel:256 Loops:32 Thr:24 Vec:1
Speed.#*.....: 83306 H/s (70.32ms) @ Accel:256 Loops:32 Thr:24 Vec:1
```




Complexity Matters

Keyspace in Billions





Time to Crack

RTX 4090

VIII

X

XII

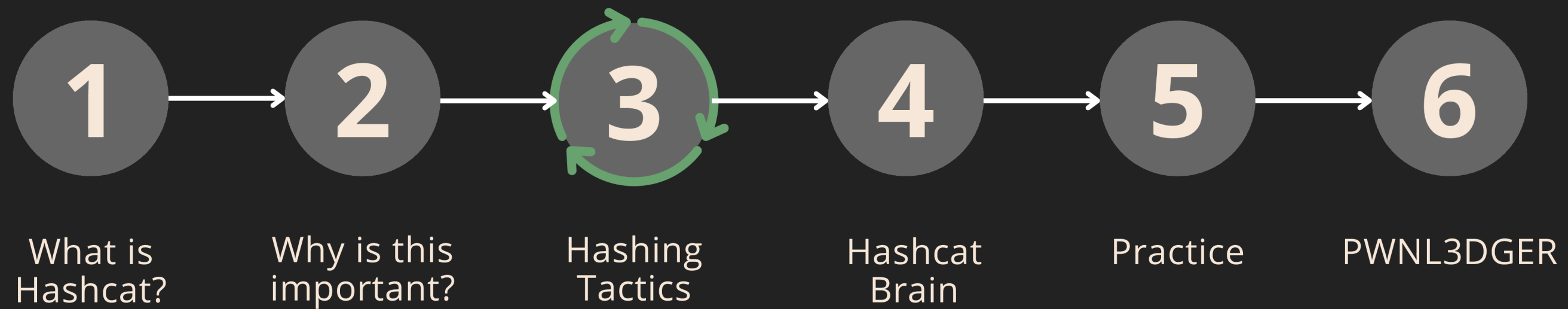
- | | |
|--|---|
| <ul style="list-style-type: none">• ?l<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 0 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 0 Hours◦ MD5: 0 Hours◦ BCrypt: 0 Hours• ?l + ?u<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 0 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 0 Hours◦ MD5: 0 Hours◦ BCrypt: 0 Hours | <ul style="list-style-type: none">• ?l + ?u + ?d<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 0 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 0 Hours◦ MD5: 0 Hours◦ BCrypt: 2 Hours• ?l + ?u + ?d + ?s<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 0 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 0 Hours◦ MD5: 0 Hours◦ BCrypt: 39 Hours |
|--|---|

- | | |
|---|--|
| <ul style="list-style-type: none">• ?l<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 0 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 0 Hours◦ MD5: 0 Hours◦ BCrypt: 1 Hours• ?l + ?u<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 0 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 0 Hours◦ MD5: 0 Hours◦ BCrypt: 1004 Hours | <ul style="list-style-type: none">• ?l + ?u + ?d<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 0 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 0 Hours◦ MD5: 0 Hours◦ BCrypt: 5828 Hours• ?l + ?u + ?d + ?s<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 0 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 0 Hours◦ MD5: 0 Hours◦ BCrypt: 336099 Hours |
|---|--|

- | | |
|--|--|
| <ul style="list-style-type: none">• ?l<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 0 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 0 Hours◦ MD5: 0 Hours◦ BCrypt: 663 Hours• ?l + ?u<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 0 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 0 Hours◦ MD5: 0 Hours◦ BCrypt: 2714424 Hours | <ul style="list-style-type: none">• ?l + ?u + ?d<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 0 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 0 Hours◦ MD5: 0 Hours◦ BCrypt: 22404630 Hours• ?l + ?u + ?d + ?s<ul style="list-style-type: none">◦ NTLM: 0 Hours◦ NTLMv2: 1 Hours◦ SHA1: 0 Hours◦ SHA256: 0 Hours◦ SHA512: 2 Hours◦ MD5: 0 Hours◦ BCrypt: 2906918732 Hours |
|--|--|



OBJECTIVES





Fast Hashing

1

1-8 Character Incremental Brute-Force

```
hashcat -O -m 1000 -a 3 ?a?a?a?a?a?a?a -i --increment-min 1 --increment-max 8 -d1,2 -w 3 --session sess1
```

2

Wordlist + Rule Attack

```
hashcat -O -m 1000 -a 0 -r rules/OneRuleToRuleThemStill.rule wordlists/rockyou.txt -d1,2 -w 3 --session sess1
```

3

Wordlist + Mask Attack

```
hashcat -m 1000 -a 6 wordlists/rockyou.txt ?a?a?a -i --increment-min 1 --increment-max 3 -d1,2 -w 3 --session sess1
```

4

Mask + Wordlist Attack

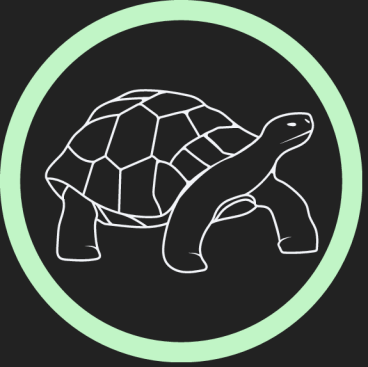
```
hashcat -m 1000 -a 7 ?a?a wordlists/rockyou.txt -d1,2 -w 3 --session sess1
```

5

Mask Attack

```
hashcat -m 1000 -a 3 masks/custom/10-char-masks -d1,2 -w 3 --session sess1
```





Slow Hashing

Computationally Expensive

1

Wordlist + Rule Attack

```
hashcat -O -m 1000 -a 0 -r rules/OneRuleToRuleThemStill.rule wordlists/rockyou.txt -d1,2 -w 3 --session sess1
```

2

Contextual Keywords + Mask Attack

```
hashcat -m 1000 -a 6 ClientName?a?a?d?d?d?d -i --increment-min 9 --increment-max 15 -d1,2 -w 3 --session sess1
```

3

Preprocessor Wordlist

```
hashcat -m 1000 -a 6 wordlists/rockyou.txt ?a?a?a -i --increment-min 1 --increment-max 3 -d1,2 -w 3 --session sess1
```

4

1-8 Character Incremental Brute-Force

```
hashcat -O -m 1000 -a 3 ?a?a?a?a?a?a?a?a -i --increment-min 1 --increment-max 8 -d1,2 -w 3 --session sess1
```

5

Mask Attack

```
hashcat -m 1000 -a 3 masks/custom/10-char-masks -d1,2 -w 3 --session sess1
```





ENGAGEMENT CYCLE



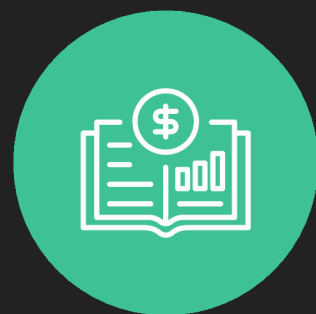
Generate Potential Keywords List

Specific to client names, words extracted from OSINT or iother sources/crackeded passwords. Can include a preprocessor



Mask Attack

Perform a mask attack utilizing masks developed from the potfile and/or other sources.



Targeted Brute Force Attack

Perform a brute force attack (1-7 characters, 8 if its fast) (mode 3 with increments)



Generate Potfile Wordlist and Masks

~/.local/share/hashcat/hashcat.potfile



Wordlist + Rule Attack

Utilize wordlists in a combinator, or rule + wordlist attack (modes 0/1)



Wordlist + Mask Attack

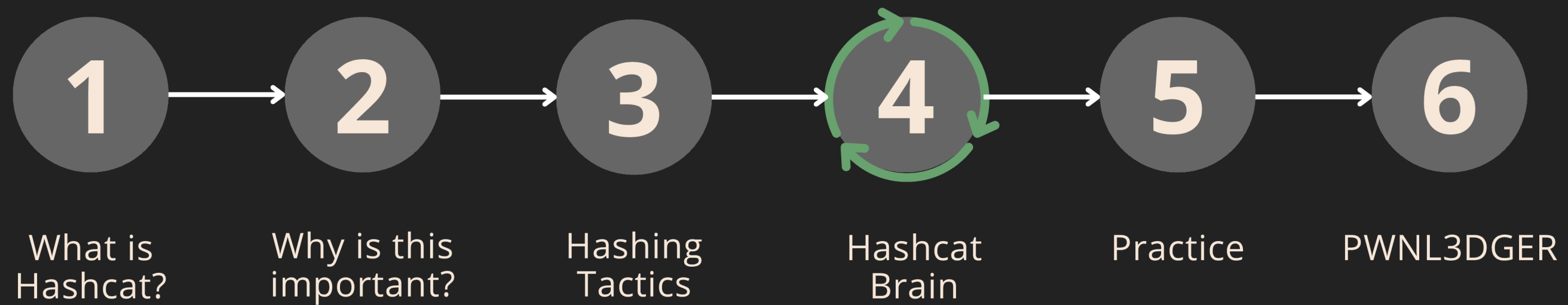
Utilize wordlists in a Wordlist + Mask Attack (modes 6/7)



<https://github.com/Ar1ste1a/Potfile>



OBJECTIVES





Hashcat Brain



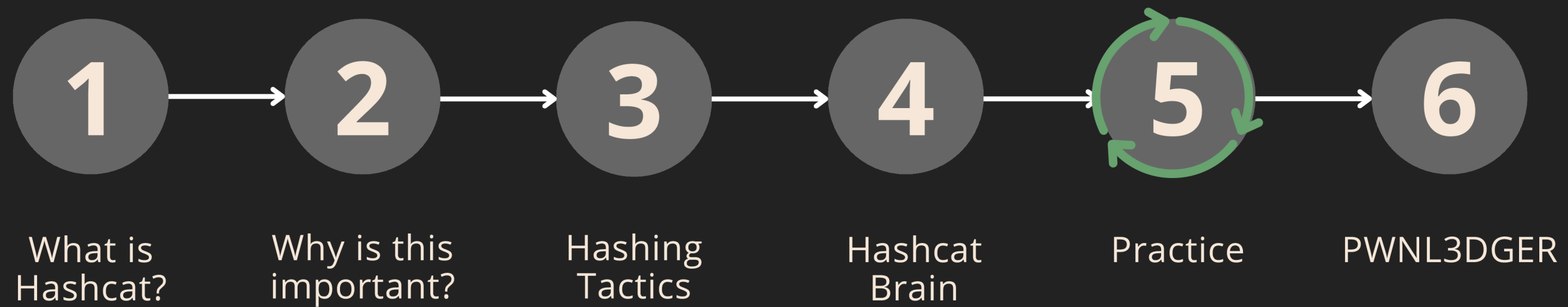
<https://github.com/Ar1ste1a/BrainDocker>

```
hashcat -O --brain-client --brain-client-features=3 --brain-host=<IP Address> --brain-port=1991 --brain-password='HashCracking!' -m 0 -a 3  
ntlm.hashes ?a?a?a?a?a?a
```

```
hashcat -O --brain-client --brain-client-features=3 --brain-host=<IP Address> --brain-port=1991 --brain-password='HashCracking!' -m 1000  
-a 0 ntlm.hashes wordlists/rockyou.txt -r rules/d3ad0ne.rule
```



OBJECTIVES





Practicing Hash Cracking Techniques

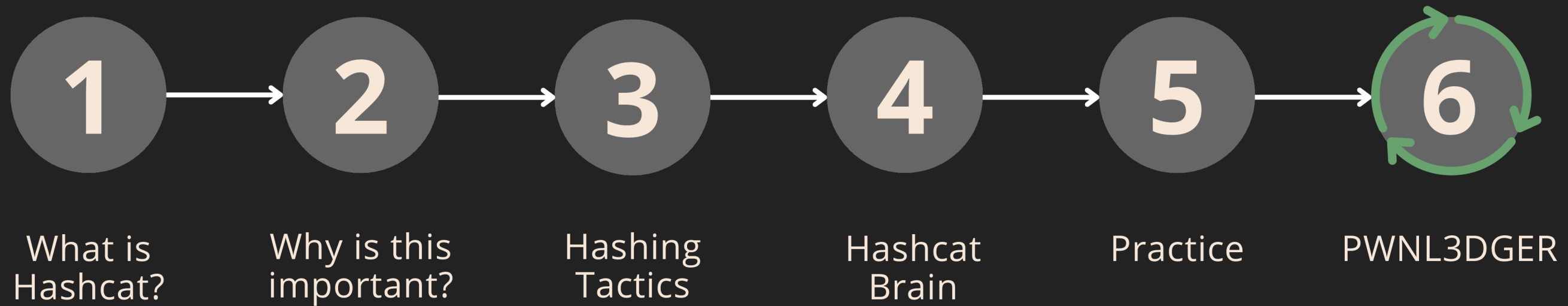
Have I Been Pwned Hashes



<https://github.com/HaveIBeenPwned/PwnedPasswordsDownloader>



OBJECTIVES



Takeaways



Defense



1

Hash cracking is only getting faster

2

Complexity + length matters

3

Implement MFA



Offense



4

Think like a (lazy) human

5

Practice makes perfect

6

Analyze your data



PWNL3DG3R



Hack the Hash: Unveiling Password Cracking Strategies

Thank you!

Questions?



 ehosinski@krakensec.tech

 <https://krakensec.tech>

 <https://linkedin.com/in/evanhosinski>

R I G

