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

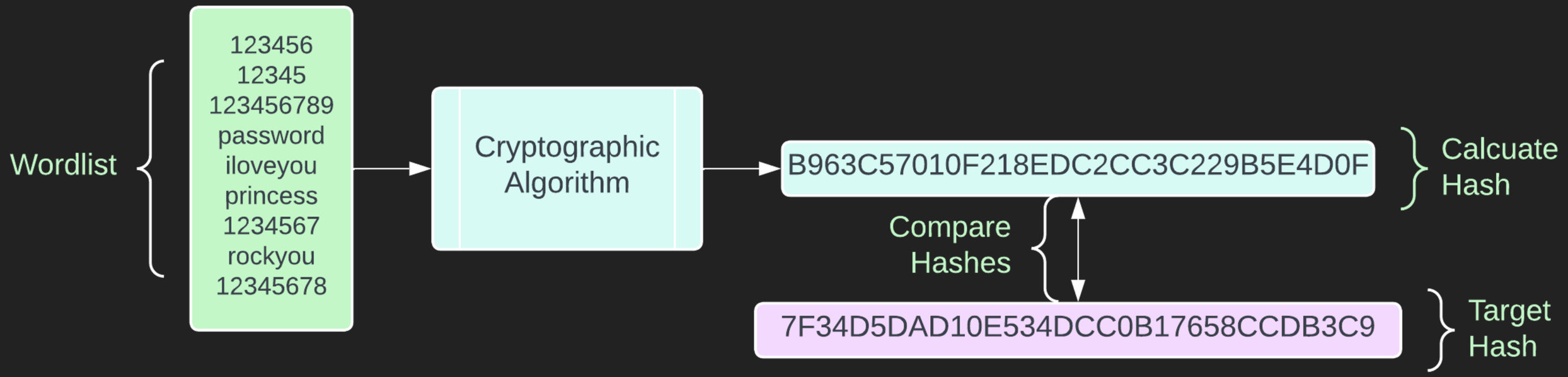
CT

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

KRAKEN TECH

What is Hash Cracking?



OBJECTIVES

1. What is Hashcat?
2. Why is this important?
3. Hashing Tactics
4. Hashcat Brain
5. Practice
6. PWNL3DGER

HC

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

KRAKEN TECH

HC

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)

KRAKEN TECH

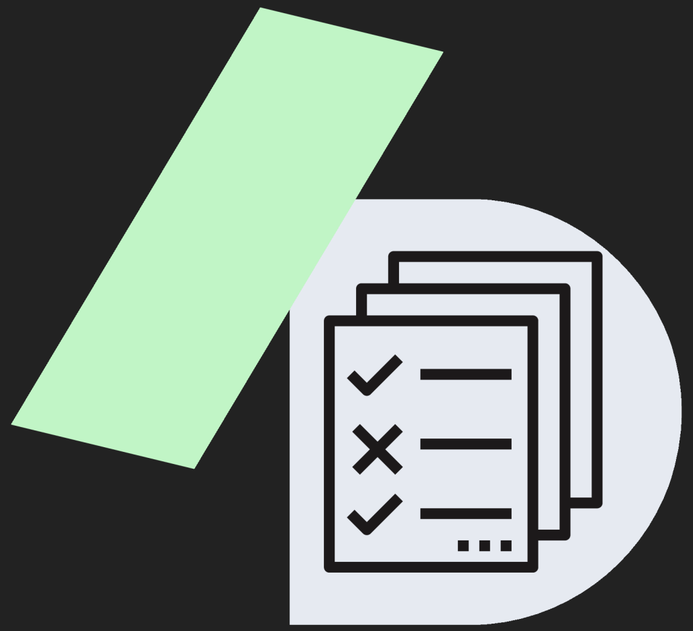
HC

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



KRAKEN TECH

HC

Masks

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

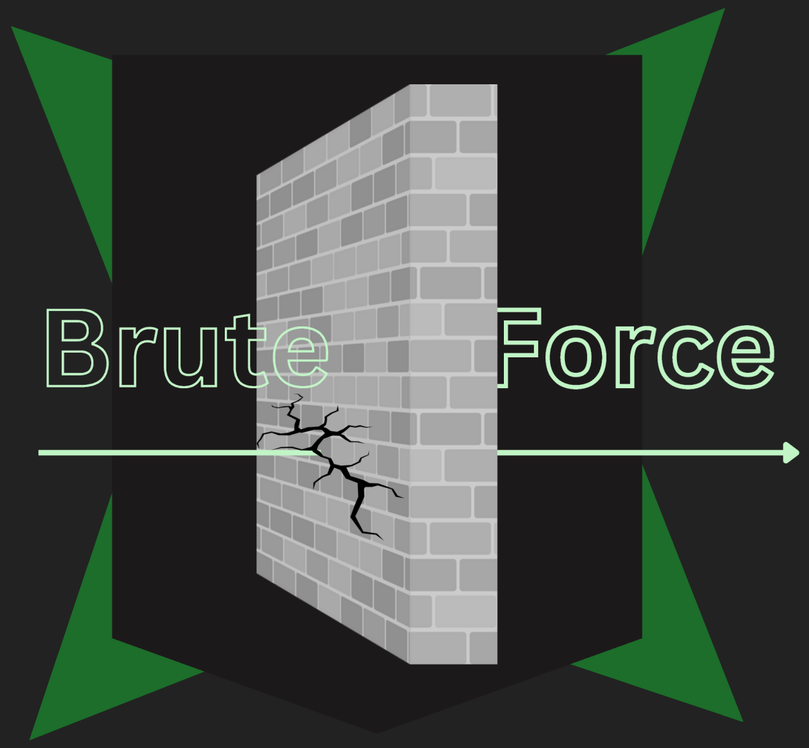
?u?l?l?l?l?l
?l?l?l?l?l?l?d
grrcon?s
?u?l?l?l?l?l?d?d?s
grrcon?d?d?s

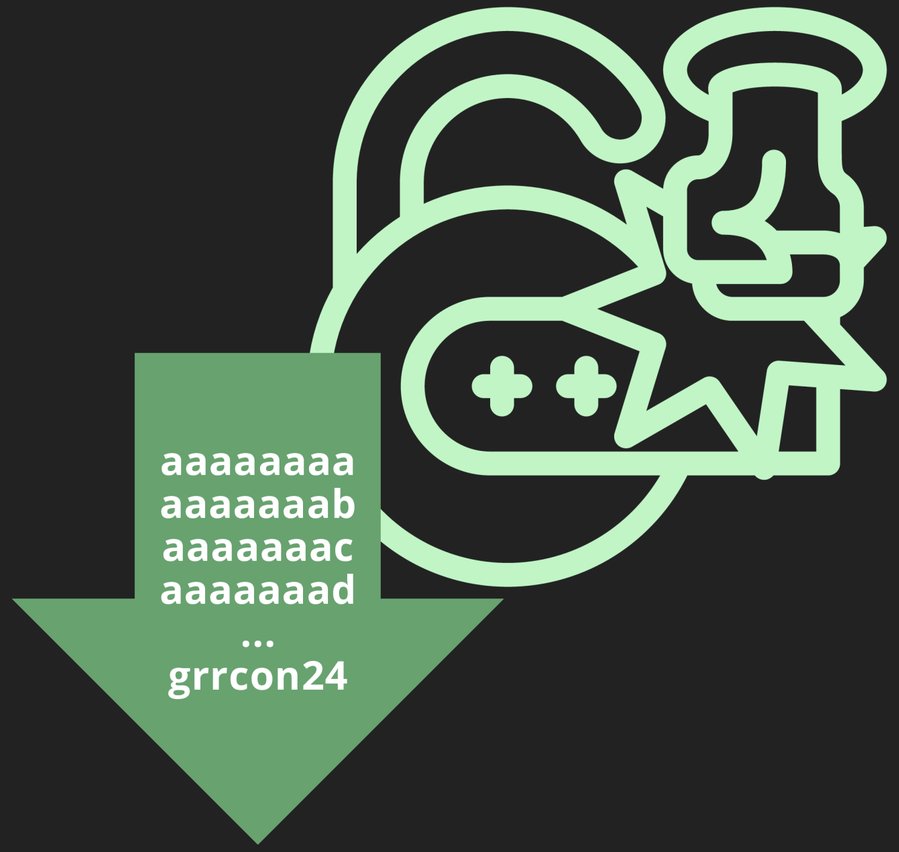
hashcat -a 3 -m 1000 hashes.txt grrcon?d?d

grrcon00
grrcon10
grrcon11
...
grrcon24

KRAKEN TECH

HC





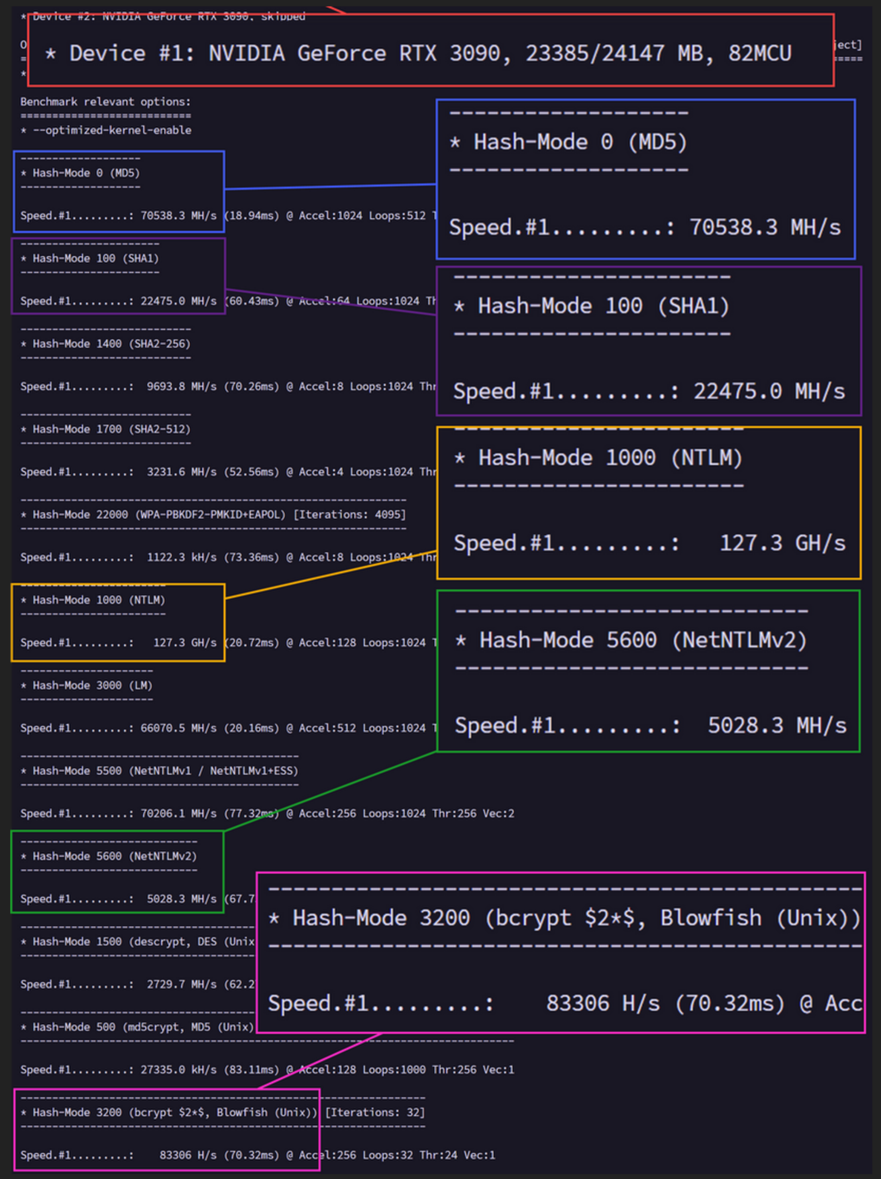
KRAKEN TECH

OBJECTIVES

1. What is Hashcat?
2. Why is this important?
3. Hashing Tactics
4. Hashcat Brain
5. Practice
6. PWNL3DGER

GPU++

Performance Increase



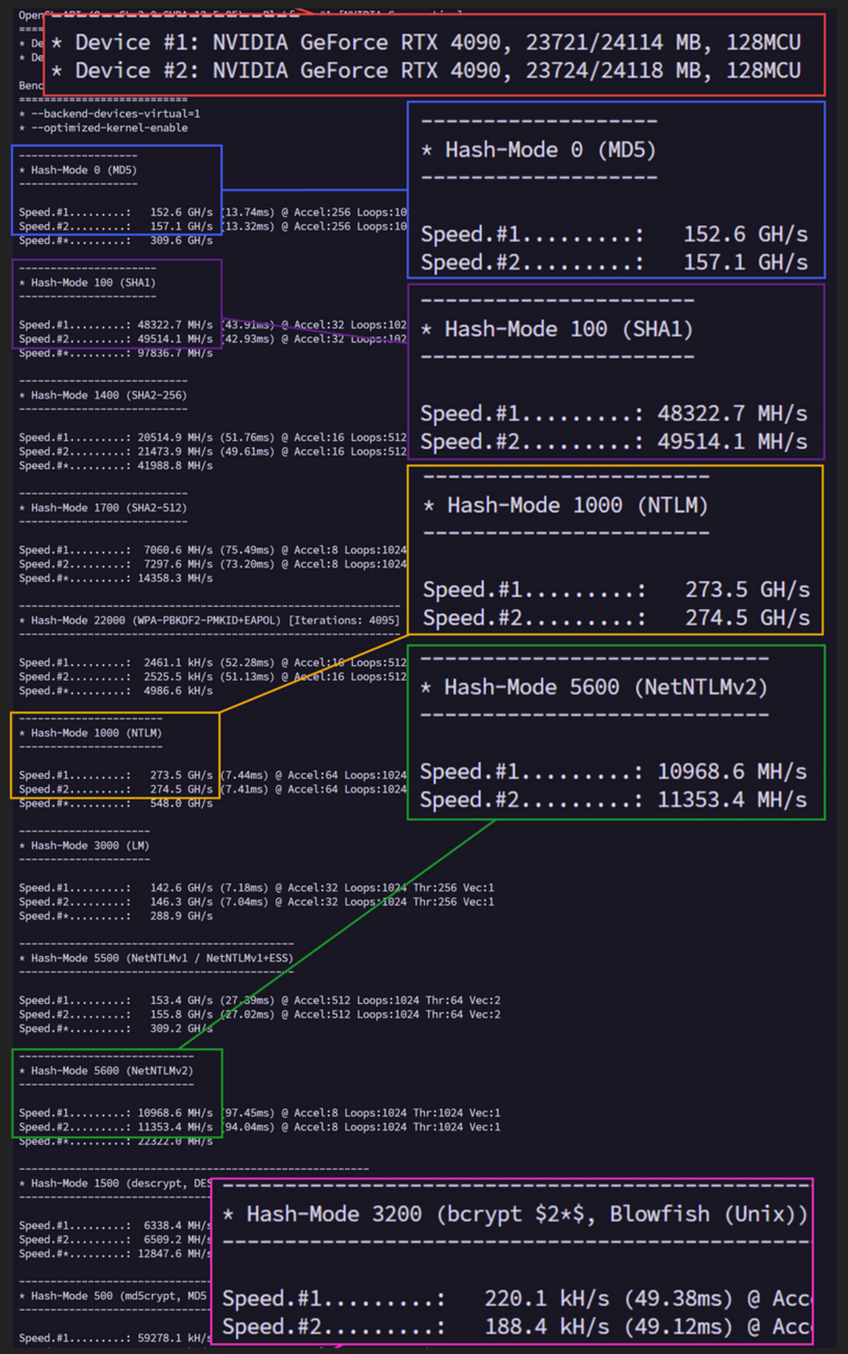
MD5: 122.72%

SHA1: 120.31%

NTLM: 115.63%

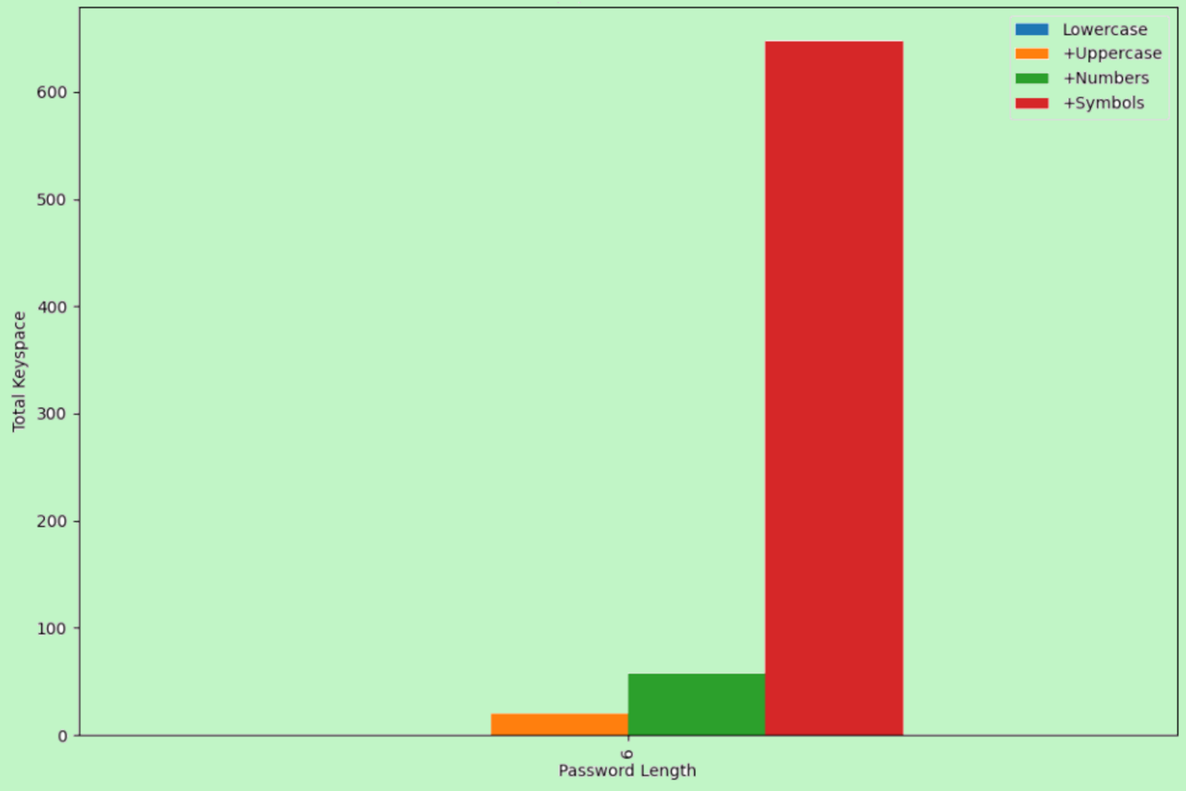
NetNTLMv2: 125.79%

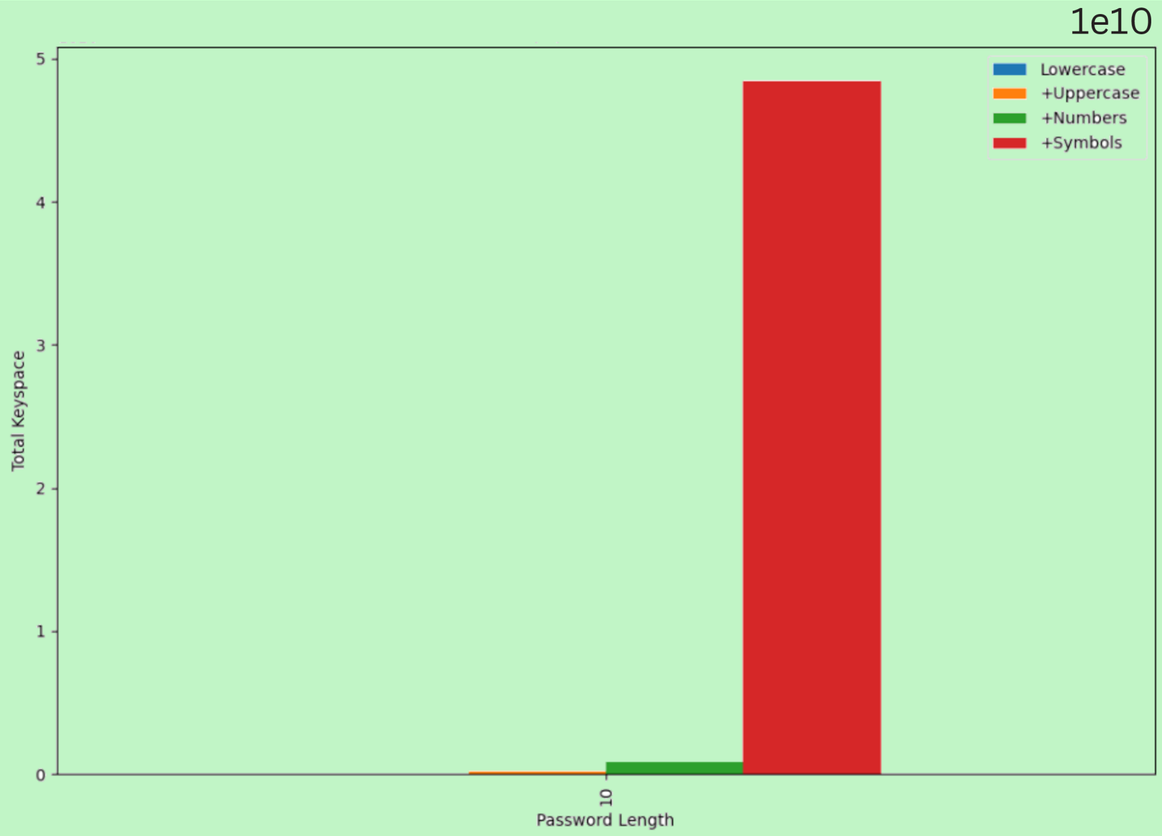
bcrypt: 164.21%



Complexity Matters

Keyspace in Billions





Time to Crack

RTX 4090

VIII

- ?l
  - NTLM: 0 Hours
  - NTLMv2: 0 Hours
  - SHA1: 0 Hours
  - SHA256: 0 Hours
  - SHA512: 0 Hours
  - MD5: 0 Hours
  - BCRYPT: 0 Hours
- ?l + ?u
  - NTLM: 0 Hours
  - NTLMv2: 0 Hours
  - SHA1: 0 Hours
  - SHA256: 0 Hours
  - SHA512: 0 Hours
  - MD5: 0 Hours
  - BCRYPT: 0 Hours
- ?l + ?u + ?d
  - 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
  - NTLM: 0 Hours
  - NTLMv2: 0 Hours
  - SHA1: 0 Hours
  - SHA256: 0 Hours
  - SHA512: 0 Hours
  - MD5: 0 Hours
  - BCRYPT: 39 Hours

X

- ?l
  - NTLM: 0 Hours
  - NTLMv2: 0 Hours
  - SHA1: 0 Hours
  - SHA256: 0 Hours
  - SHA512: 0 Hours
  - MD5: 0 Hours
  - BCRYPT: 1 Hours
- ?l + ?u
  - NTLM: 0 Hours
  - NTLMv2: 0 Hours
  - SHA1: 0 Hours
  - SHA256: 0 Hours
  - SHA512: 0 Hours
  - MD5: 0 Hours
  - BCRYPT: 1004 Hours
- ?l + ?u + ?d
  - 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
  - NTLM: 0 Hours
  - NTLMv2: 0 Hours
  - SHA1: 0 Hours
  - SHA256: 0 Hours
  - SHA512: 0 Hours
  - MD5: 0 Hours
  - BCRYPT: 336099 Hours

XII

- ?l
  - NTLM: 0 Hours
  - NTLMv2: 0 Hours
  - SHA1: 0 Hours
  - SHA256: 0 Hours
  - SHA512: 0 Hours
  - MD5: 0 Hours
  - BCRYPT: 663 Hours
- ?l + ?u
  - NTLM: 0 Hours
  - NTLMv2: 0 Hours
  - SHA1: 0 Hours
  - SHA256: 0 Hours
  - SHA512: 0 Hours
  - MD5: 0 Hours
  - BCRYPT: 2714424 Hours
- ?l + ?u + ?d
  - 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
  - NTLM: 0 Hours
  - NTLMv2: 1 Hours
  - SHA1: 0 Hours
  - SHA256: 0 Hours
  - SHA512: 2 Hours
  - MD5: 0 Hours
  - BCRYPT: 2906918732 Hours

OBJECTIVES

1. What is Hashcat?
2. Why is this important?
3. Hashing Tactics
4. Hashcat Brain
5. Practice
6. PWNL3DGER

Fast Hashing

1. 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
```

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 utilziing 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

~/.locall/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

1. What is Hashcat?
2. Why is this important?
3. Hashing Tactics
4. Hashcat Brain
5. Practice
6. PWNL3DGER

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?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

1. What is Hashcat?
2. Why is this important?
3. Hashing Tactics
4. Hashcat Brain
5. Practice
6. PWNL3DGER

Practicing Hash Cracking Techniques

Have I Been Pwned Hashes

https://github.com/HaveIBeenPwned/PwnedPasswordsDownloader

OBJECTIVES

1. What is Hashcat?
2. Why is this important?
3. Hashing Tactics
4. Hashcat Brain
5. Practice
6. PWNL3DGER

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

KRAKEN TECH

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