COMP6224 Foundations of Cyber Security 2022/23

Coursework on Cyber Attack Analysis and Password Cracking

Student ID: *34309675*

Part 1 – Cyber-Attack Analysis

Task 1.1 – Kill Chain-based Analysis

[*Note that you may need to add further sections, depending on how many phases the attack went through*]

This is a multi-step cyber-attack(3 steps).

Reconnaissance Phase(1)

[*Describe what happened during the "reconnaissance" phase. What information did attackers gather? How? Max 100 words*]

monitor the activities of the targeted developers

Weaponization Phase(1)

[*Describe what happened during the "weaponization" phase. What cyber weapons have been used? How did attackers obtain them? Max 100 words*]

using a spear-phishing email with a malicious attachment

Delivery Phase(1)

[*Describe what happened during the "delivery" phase. How did the attackers deliver the cyber weapon(s) to the intended target? Max 100 words*]

via email

Exploitation Phase(1)

[*Describe what happened during the "exploitation" phase. How have cyber weapons been activated? Max 100 words*]

malicious attachment opened via user deception

Installation Phase(1)

[*Describe what happened during the "installation" phase. How did the attackers gain persistence inside the target? Max 100 words*]

activated a remote access trojan that registered itself as auto start service in the machine

Command and Control Phase(1)

[*Describe what happened during the "command and control" phase. How did the attackers establish a communication channel to control the cyber weapons installed inside the target? What kind of instructions did the attackers send? What types of information did the cyber weapons send back to the attackers? Max 100 words*]

attackers establish a remote connection via HTTPS provided by the trojan

Reconnaissance Phase(2)

figure out how to develop software that made the router firmware vulnerable and the internal procedures to upload updated software to the repository

Weaponization Phase(2)

malicious code (vulnerable firmware update)

Delivery Phase(2)

use developer's machine which is compromised in the first step to push the vulnerable firmware update into the internal repository

Exploitation Phase(2)

source code is used to generate the new firmware with the vulnerability

Installation Phase(2)

the vulnerable firmware is installed in the router XYZ sell and developer's machine has been compromised in the first iteration

Command and Control Phase(2)

no information available

Reconnaissance Phase(3)

find those XYZ routers with vulnerable firmware version 1.2.3 installed in the Internet

Weaponization Phase(3)

cryptominer malware

Delivery Phase(3)

via internet

Exploitation Phase(3)

the vulnerability of firmware is exploited via internet

Installation Phase(3)

Not mentioned, vulnerable firmware was installed in the router in the second step.

Command and Control Phase(3)

Once activated, the cryptominer malware connected to a remote server to register the infection and receive instructions regarding the cryptomining activities to perform

Actions on Objective Phase

[*Describe what happened during the "actions on objective" phase. What did the attacker do to achieve their goals? Max 100 words*]

The routers execute instructions regarding the cryptomining activities to perform; the cryptominer malware propagated over the Internet to infect as many other vulnerable XYZ routers as possible

Task 1.2 – Cyber Actor Analysis

[*Most likely cyber actor profile (or combinations of cyber actor profiles)*]

Cyber Actor Profile:

Cybercriminal

Attack Strategy

[*Discuss how the attack strategy used in this cyber-attack compares to the attack strategies commonly used by this cyber actor profile. Max 100 words*]

Attack strategies commonly followed kill chain model, the attack vectors are Malware, Social Engineering/Email, Botnet and typical attacks include Money theft, personal document ransom, data breaches and DDoS. In this attack the cybercriminals' attack strategy use the kill chain model and attack vectors include 1. phishing email with trojan 2. Malware. This attack aim to steal computing power to gain cryptocurrency and with high level of technical because cybercriminals need to develop trojan and software that made router firmware vulnerable.

Motivations

[*Discuss how the motivations of the attacker in this cyber-attack compares to the motivations that generally characterise this cyber actor profile. Max 100 words*]

Cybercriminals generally interested in illegal profit, in this cyber-attack, cybercriminals use infected routers to perform cryptomining activities to gain profit.

Part 2 – Password Cracking

Task 2.1 - Dictionary-based cracking of passwords

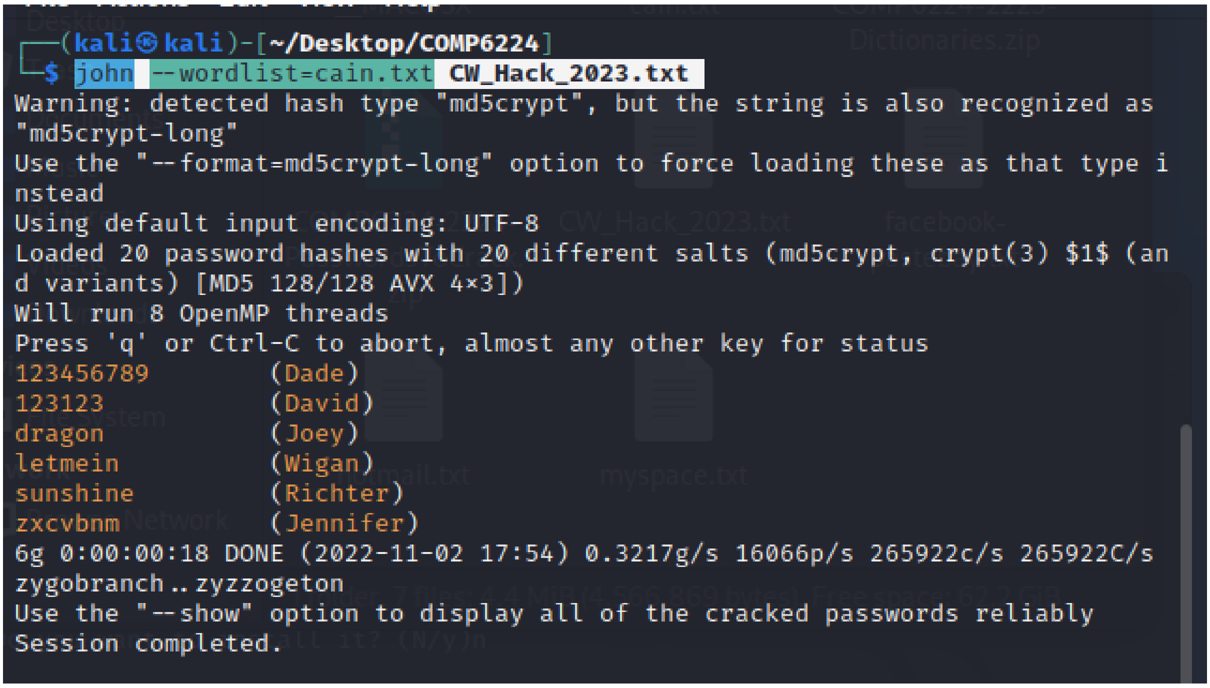
[*Up to 3 dictionaries, up to 20 passwords overall; you will need to add further hash/password entries*]

Dictionary #1

Name of the file: cain.txt

File source: https://secure.ecs.soton.ac.uk/noteswiki/images/COMP6224-222©3-Dictionaries.zip

Command (also include a screenshot): john --wordlist=cain.txt CW_Hack_2023.txt



```
┌──(kali㉿kali)-[~/Desktop/COMP6224]
└─$ john --wordlist=cain.txt CW_Hack_2023.txt
Warning: detected hash type "md5crypt", but the string is also recognized as
"md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type i
nstead
Using default input encoding: UTF-8
Loaded 20 password hashes with 20 different salts (md5crypt, crypt(3) $1$ (an
d variants) [MD5 128/128 AVX 4×3])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
123456789        (Dade)
123123           (David)
dragon           (Joey)
letmein          (Wigan)
sunshine         (Richter)
zxcvbnm          (Jennifer)
6g 0:00:00:18 DONE (2022-11-02 17:54) 0.3217g/s 16066p/s 265922c/s 265922C/s
zygobranch..zyzzogeton
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Cracked passwords

- $1$DDY0yMZV$3nfvqmRjl6lvyD5TDA5aQ.
- 123456789
- $1$NVqFKDAJ$q7.GKuLp.81NvWTTdz7Mk/
- 123123
- $1$YTmCElPR$Dz.fVWaiJ6WRYa4WuBSSK.
- dragon
- $1$ogsh4He8$iDMirR6H.iwndmp1x49PB/
- letmein

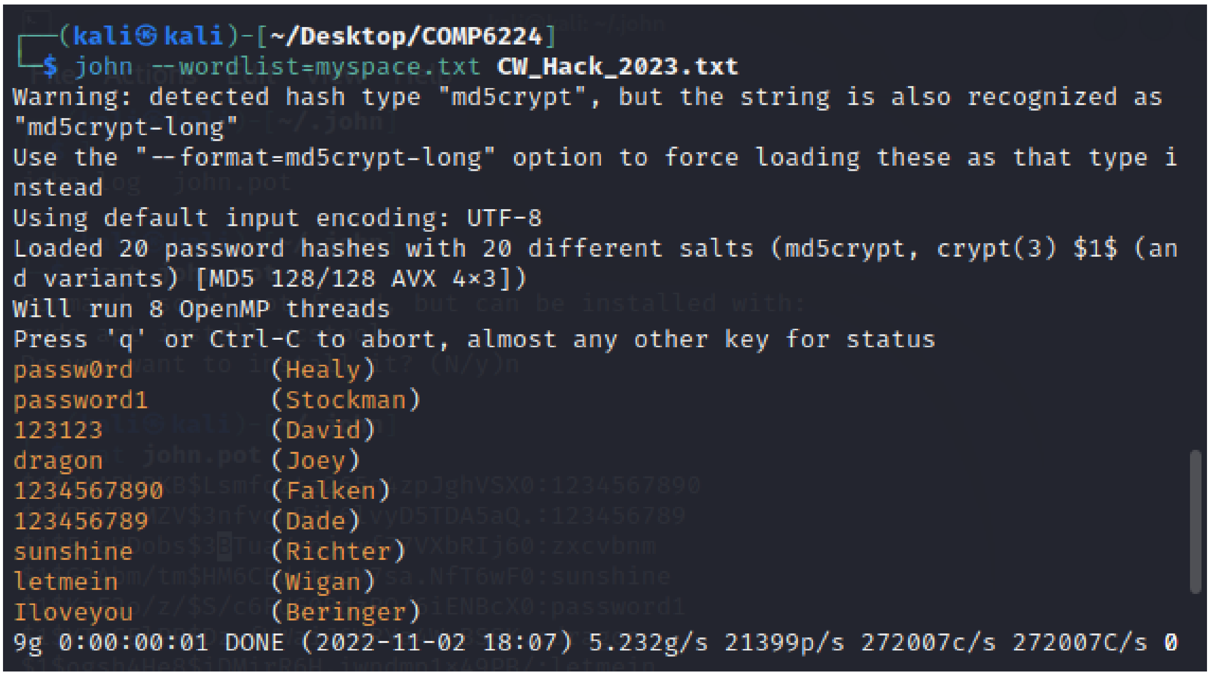
- $1$G2Abm/tm$HM6CEdctwcM7sa.NfT6wF0
- sunshine
- $1$E4cHDobs$3BTuaVxojwwf77VXbRIj60
- zxcvbnm

Dictionary #2

Name of the file: myspace.txt

File source: https://secure.ecs.soton.ac.uk/noteswiki/images/COMP6224-222©3-Dictionaries.zip

Command (also include a screenshot): john --wordlist=myspace.txt CW_Hack_2023.txt



```
┌──(kali㉿kali)-[~/Desktop/COMP6224]
└─$ john --wordlist=myspace.txt CW_Hack_2023.txt
Warning: detected hash type "md5crypt", but the string is also recognized as
"md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type i
nstead
Using default input encoding: UTF-8
Loaded 20 password hashes with 20 different salts (md5crypt, crypt(3) $1$ (an
d variants) [MD5 128/128 AVX 4×3])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
passw0rd         (Healy)
password1        (Stockman)
123123           (David)
dragon           (Joey)
1234567890       (Falken)
123456789        (Dade)
sunshine         (Richter)
letmein          (Wigan)
Iloveyou         (Beringer)
9g 0:00:00:01 DONE (2022-11-02 18:07) 5.232g/s 21399p/s 272007c/s 272007C/s 0
```

Cracked passwords

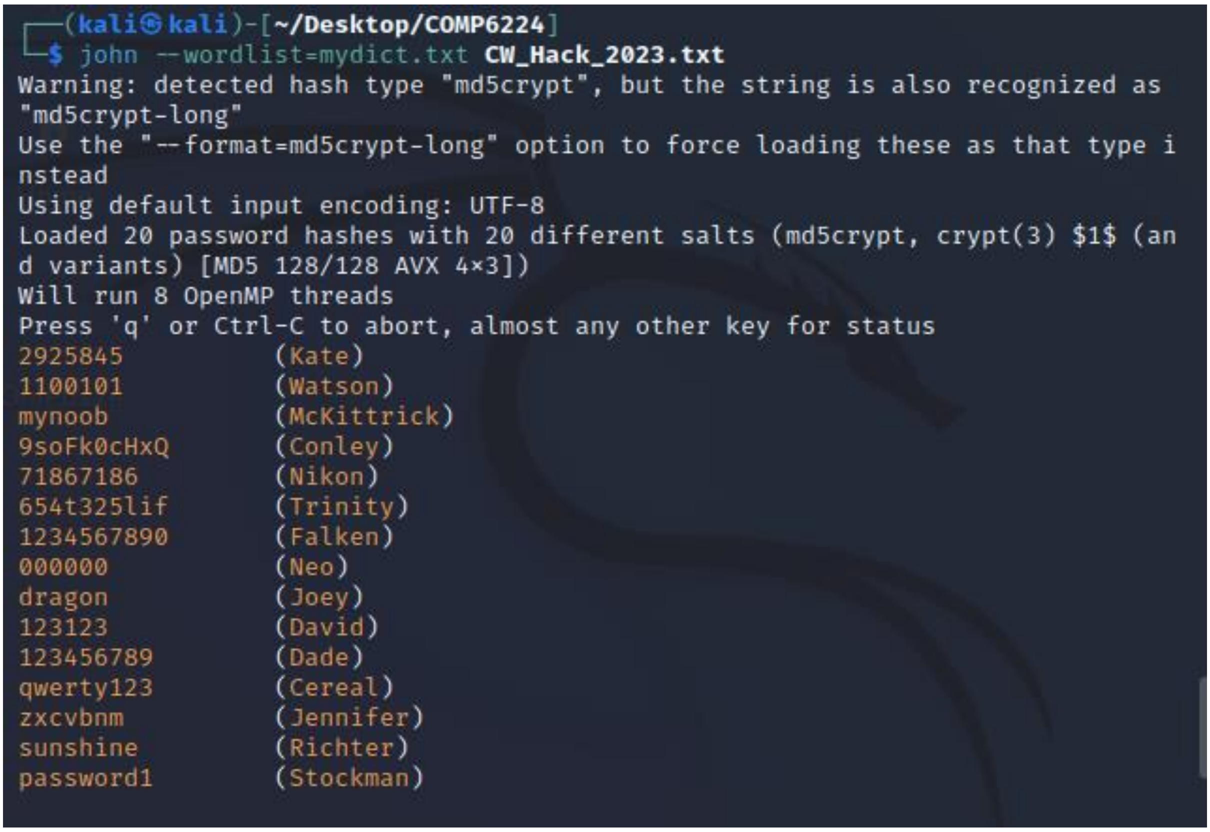
- $1$JnAeQVME$QBtFn2xTUSUGrT5aWBsA80
- passw0rd
- $1$KzF2o/z/$S/c6FUG0BdzRQ/6iENBcX0
- password1
- $1$NVqFKDAJ$q7.GKuLp.81NvWTTdz7Mk/
- 123123
- $1$YTmCElPR$Dz.fVWaiJ6WRYa4WuBSSK.
- dragon
- $1$2AABkRKB$Lsmfqzta265n4zpJghVSX0
- 1234567890
- $1$DDY0yMZV$3nfvqmRjl6lvyD5TDA5aQ.
- 123456789
- $1$G2Abm/tm$HM6CEdctwcM7sa.NfT6wF0
- sunshine
- $1$ogsh4He8$iDMirR6H.iwndmp1x49PB/
- letmein
- $1$EpTiLUre$8vQ6HYyKoG6HP5LdiRLrg0
- Iloveyou

Dictionary #3

Name of the file:

File source: [https://github.com/duyet/bruteforce-database,rockyou.txt, and](https://github.com/duyet/bruteforce-database) my dictionary

Command (also include a screenshot): $ john --wordlist=mydict.txt CW_Hack_2023.txt



```
┌──(kali㉿kali)-[~/Desktop/COMP6224]
└─$ john --wordlist=mydict.txt CW_Hack_2023.txt
Warning: detected hash type "md5crypt", but the string is also recognized as
"md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type i
nstead
Using default input encoding: UTF-8
Loaded 20 password hashes with 20 different salts (md5crypt, crypt(3) $1$ (an
d variants) [MD5 128/128 AVX 4×3])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
2925845          (Kate)
1100101          (Watson)
mynoob           (McKittrick)
9soFk0cHxQ       (Conley)
71867186         (Nikon)
654t325lif       (Trinity)
1234567890       (Falken)
000000           (Neo)
dragon           (Joey)
123123           (David)
123456789        (Dade)
qwerty123        (Cereal)
zxcvbnm          (Jennifer)
sunshine         (Richter)
password1        (Stockman)
```

Cracked passwords

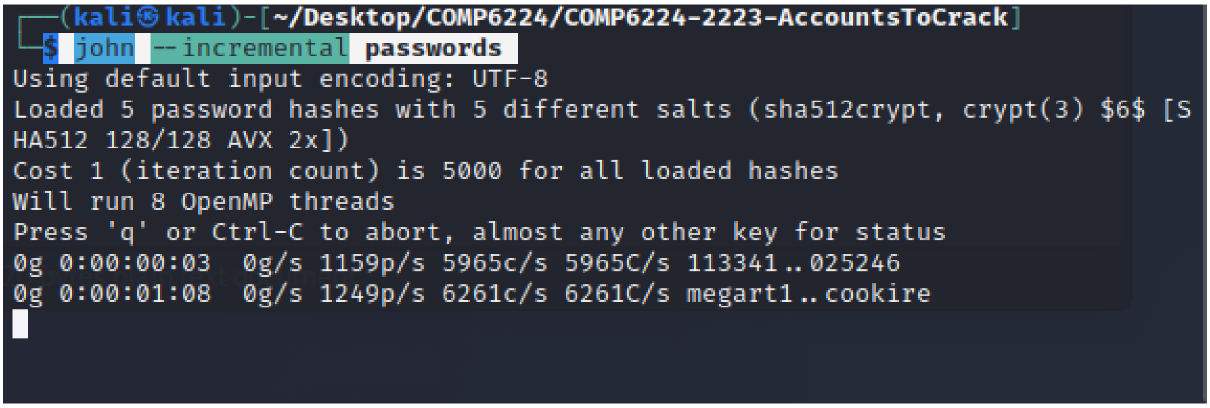
- $1$VjdzN9p6$GHsIVVgKAj49QakLdV9aJ0
- 000000
- $1$Kx/qvi5/$LP..XjweHe1gWJeXrpbdh/
- 1100101
- $1$NVqFKDAJ$q7.GKuLp.81NvWTTdz7Mk/
- 123123
- $1$2AABkRKB$Lsmfqzta265n4zpJghVSX0
- 1234567890
- $1$DDY0yMZV$3nfvqmRjl6lvyD5TDA5aQ.
- 123456789
- $1$k8GN3C7Y$p9MODWBc6YjCPt1e9VYHG1
- 71867186
- $1$YTmCElPR$Dz.fVWaiJ6WRYa4WuBSSK.
- dragon
- $1$EpTiLUre$8vQ6HYyKoG6HP5LdiRLrg0
- Iloveyou
- $1$ogsh4He8$iDMirR6H.iwndmp1x49PB/
- letmein
- $1$JnAeQVME$QBtFn2xTUSUGrT5aWBsA80
- passw0rd
- $1$KzF2o/z/$S/c6FUG0BdzRQ/6iENBcX0
- password1

- $1$At1h0BZQ$H1p4TXq7AjEoYlmu.PUuJ.
- qazwsx
- $1$ydYZLxdN$.AI/jI36bqRtUwNeR5Ocr1
- qazwsx
- $1$hAr5LZNv$12/M0Nzo.IgVnGIlLPYKn0
- qwerty123
- $1$G2Abm/tm$HM6CEdctwcM7sa.NfT6wF0
- sunshine
- $1$E4cHDobs$3BTuaVxojwwf77VXbRIj60
- zxcvbnm
- $1$PcDTObtW$PivNVYnv3DioRK/4SJJo3/
- 9soFk0cHxQ
- $1$zHExthhZ$DPd5aFaKrA1yV09DI3zLM.
- 654t325lif
- $1$AXM0vqt.$Pr.jDvYV41.zzTs0pDHQr/
- 2925845
- $1$MfDcSNNn$wNoMvVNygSEe0NBE6E9Db/
- mynoob

Task 2.2 - Password cracking of Linux accounts

Brute force

Command (also include a screenshot): $ john --incremental passwords



```
┌──(kali㉿kali)-[~/Desktop/COMP6224/COMP6224-2223-AccountsToCrack]
└─$ john --incremental passwords
Using default input encoding: UTF-8
Loaded 5 password hashes with 5 different salts (sha512crypt, crypt(3) $6$ [S
HA512 128/128 AVX 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:03  0g/s 1159p/s 5965c/s 5965C/s 113341..025246
0g 0:00:01:08  0g/s 1249p/s 6261c/s 6261C/s megart1..cookire
```

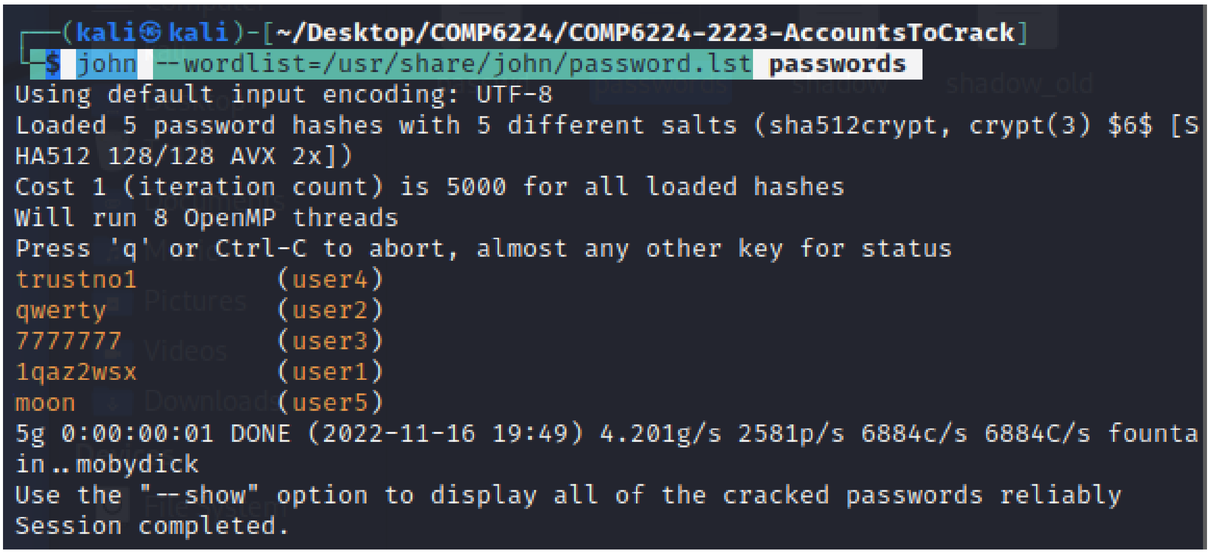
Cracked passwords

- user5
- moon

Dictionary

Command (also include a screenshot): —$ john --wordlist=/usr/share/john/password.lst

passwords



```
┌──(kali㉿kali)-[~/Desktop/COMP6224/COMP6224-2223-AccountsToCrack]
└─$ john --wordlist=/usr/share/john/password.lst passwords
Using default input encoding: UTF-8
Loaded 5 password hashes with 5 different salts (sha512crypt, crypt(3) $6$ [S
HA512 128/128 AVX 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
trustno1         (user4)
qwerty           (user2)
7777777          (user3)
1qaz2wsx         (user1)
moon             (user5)
5g 0:00:00:01 DONE (2022-11-16 19:49) 4.201g/s 2581p/s 6884c/s 6884C/s founta
in..mobydick
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Cracked passwords

- user1
- 1qaz2wsx
- user2
- qwerty
- user3
- 7777777
- user4
- trustno1
- user5
- moon

Result Comparison [max 200 words]

Brute force check a large number of possible keys, so it costs much time. In this case, I ran the brute force for 10min with only one password cracked(User5 moon, which only has 4 characters), but dictionary cracked all the passwords in 1 second. Dictionary attack check the password with most possibility of success and less time consuming than brute force. The success of brute force cracking and the cracking time depend on the complexity of the password, while the dictionary attack depends on whether the password is in the dictionary

Task 2.3 - Password analysis

[*Up to 25 passwords, up to 4 weaknesses per password*]

Password #1

Password: 000000

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- contains only numbers without letters and special characters
- password entropy is low (many repeated characters)

Password #2

Password: 1100101

Weaknesses

- too short
- password entropy is low (many repeated characters)

- contains only numbers without letters and special characters
- This password is contained in publicly available password cracking dictionaries

Password #3

Password: 123123

Weaknesses

- too short
- very common password and easy to guess
- contains only numbers without letters and special characters
- This password is contained in publicly available password cracking dictionaries

Password #4

Password: 1234567890

Weaknesses

- very common password and easy to guess
- contains only numbers without letters and special characters
- This password is contained in publicly available password cracking dictionaries
- passwords follow the rules of keyboard input

Password #5

Password: 123456789

Weaknesses

- too short
- contains only numbers without letters and special characters
- very common password and easy to guess
- This password is contained in publicly available password cracking dictionaries

Password #6

Password: 71867186

Weaknesses

- too short
- contains only numbers without letters and special characters
- This password is contained in publicly available password cracking dictionaries
- password entropy is low (many repeated characters)

Password #7

Password: dragon

Weaknesses

- too short
- only include lowercase letters without numbers and uppercase letters and special characters
- This password is contained in publicly available password cracking dictionaries
- very common password and easy to guess

Password #8

Password: Iloveyou

Weaknesses

- too short
- only include letters without numbers and special characters
- This password is contained in publicly available password cracking dictionaries
- contains common words and high frequency of use

Password #9

Password: letmein

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- only include letters without numbers and special characters
- contains common words and high frequency of use

Password #10

Password: passw0rd

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- contains common words and high frequency of use
- only include lowercase letters and numbers without uppercase letters and special characters

Password #11

Password: password1

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- contains common words and high frequency of use
- only include lowercase letters and numbers without uppercase letters and special characters

Password #12

Password: qazwsx

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- This password does not contain any numbers, making it more vulnerable to brute force attacks
- This password does not contain any special characters, making it more vulnerable to brute force attacks

Password #13

Password: qazwsx

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- This password does not contain any numbers, making it more vulnerable to brute force attacks
- This password does not contain any special characters, making it more vulnerable to brute force attacks

Password #14

Password: qwerty123

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- only include lowercase letters and numbers without uppercase letters and special characters

- This password does not contain any special characters, making it more vulnerable to brute force attacks

Password #15

Password: sunshine

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- only include letters without numbers and special characters
- contains common words and high frequency of use

Password #16

Password: zxcvbnm

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- This password does not contain any numbers, making it more vulnerable to brute force attacks
- This password does not contain any special characters, making it more vulnerable to brute force attacks

Password #17

Password: 9soFk0cHxQ

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- contains common words and high frequency of use(soFK0c)
- only include letters and numbers without special characters

Password #18

Password: 654t325lif

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- only include lowercase letters and numbers without uppercase letters and special characters
- This password does not contain any special characters, making it more vulnerable to brute force attacks

Password #19

Password: 2925845

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- contains only numbers without letters and special characters
- This password does not contain any special characters, making it more vulnerable to brute force attacks

Password #20

Password: mynoob

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short

- only include letters without numbers and special characters
- contains common words and high frequency of use

Password #21

Password: 1qaz2wsx

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- This password does not contain any uppercase letters, making it more vulnerable to brute force attacks
- This password does not contain any special characters, making it more vulnerable to brute force attacks

Password #22

Password: qwerty

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- This password does not contain any numbers, making it more vulnerable to brute force attacks
- This password does not contain any special characters, making it more vulnerable to brute force attacks

Password #23

Password: 7777777

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- contains only numbers without letters and special characters
- password entropy is low (many repeated characters)

Password #24

Password: trustno1

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- contains common words and high frequency of use
- only include lowercase letters and numbers without uppercase letters and special characters

Password #25

Password: moon

Weaknesses

- This password is contained in publicly available password cracking dictionaries
- too short
- only include letters without numbers and special characters
- contains common words and high frequency of use