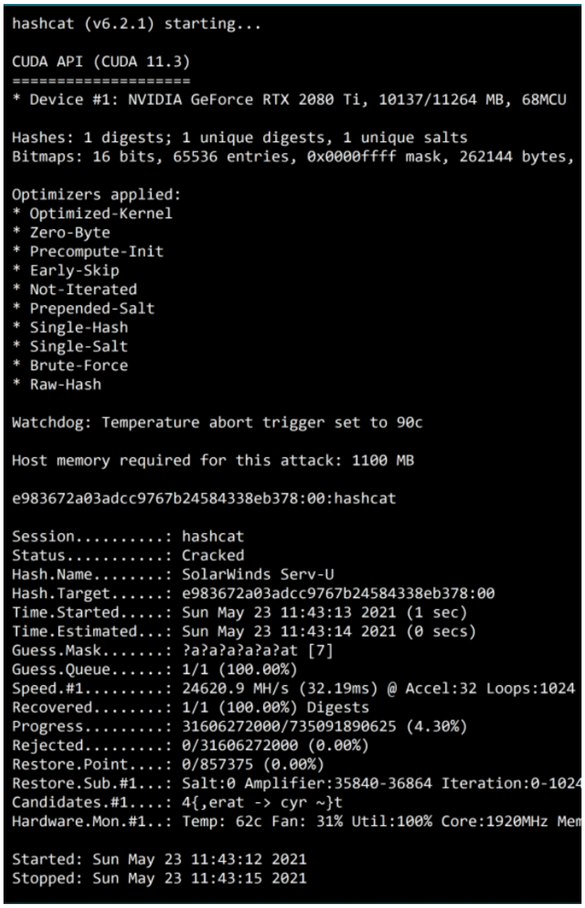
Lab 4

Cracking Passwords



```
hashcat (v6.2.1) starting...

CUDA API (CUDA 11.3)
====================
* Device #1: NVIDIA GeForce RTX 2080 Ti, 10137/11264 MB, 68MCU

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes,

Optimizers applied:
* Optimized-Kernel
* Zero-Byte
* Precompute-Init
* Early-Skip
* Not-Iterated
* Prepended-Salt
* Single-Hash
* Single-Salt
* Brute-Force
* Raw-Hash

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1100 MB

e983672a03adcc9767b24584338eb378:00:hashcat

Session..........: hashcat
Status...........: Cracked
Hash.Name........: SolarWinds Serv-U
Hash.Target......: e983672a03adcc9767b24584338eb378:00
Time.Started.....: Sun May 23 11:43:13 2021 (1 sec)
Time.Estimated...: Sun May 23 11:43:14 2021 (0 secs)
Guess.Mask.......: ?a?a?a?a?a?at [7]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........: 24620.9 MH/s (32.19ms) @ Accel:32 Loops:1024
Recovered........: 1/1 (100.00%) Digests
Progress.........: 31606272000/735091890625 (4.30%)
Rejected.........: 0/31606272000 (0.00%)
Restore.Point....: 0/857375 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:35840-36864 Iteration:0-1024
Candidates.#1....: 4{,erat -> cyr ~}t
Hardware.Mon.#1..: Temp: 62c Fan: 31% Util:100% Core:1920MHz Mem

Started: Sun May 23 11:43:12 2021
Stopped: Sun May 23 11:43:15 2021
```

Estimated time to complete: 20 Min

WHAT YOU WILL LEARN

Learn how passwords and hashing works. Use Hashcat and RockYou.txt to crack an MD5 secured password. Find out how secure your password is with haveibeenpwned and other online tools.

WHY IT'S IMPORTANT

Understand the importance of password security and how hackers are most likely to attack your passwords.

Understand the basics hashing and basic password cracking tools like Hashcat and Rockyou.txt.

SKILLS GAINED

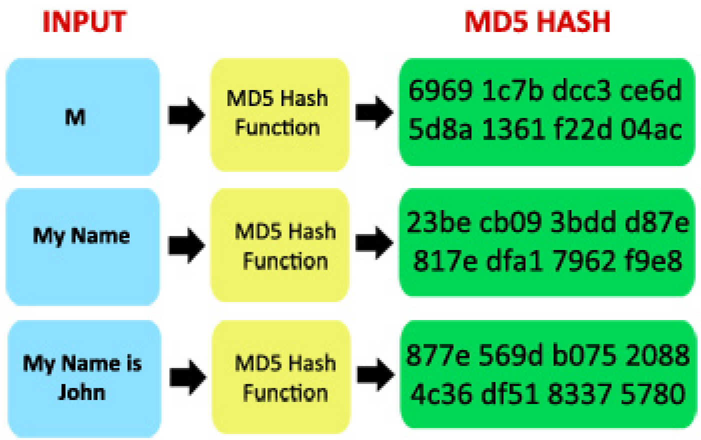
- Password hashing
- How to create a secure password
- Using Hashcat and Rockyou.txt
- Password management

REQUIRED HARDWARE

- Raspberry Pi
- Internet Connection

INTRODUCTION TO HASHING

Hashing is an algorithm performed on data such as a file or message to produce a number called a hash (sometimes called a checksum). The hash is used to verify that data is not modified, tampered with, or corrupted. In other words, you can verify the data has maintained integrity. A key point about a hash is that no matter how many times you execute the hashing algorithm against the data, the hash will always be the same if the data is the same. However, hashing can only be performed one way. This means that we cannot take a created hash and turn it back into a password. Message Digest 5 (MD5) is a common hashing algorithm that produces a 128-bit hash, this is the algorithm that we will be cracking.


INPUT
MD5 HASH
M
MD5 Hash Function
6969 1c7b dcc3 ce6d 5d8a 1361 f22d 04ac
My Name
MD5 Hash Function
23be cb09 3bdd d87e 817e dfa1 7962 f9e8
My Name is John
MD5 Hash Function
877e 569d b075 2088 4c36 df51 8337 5780

ROCKYOU.TXT

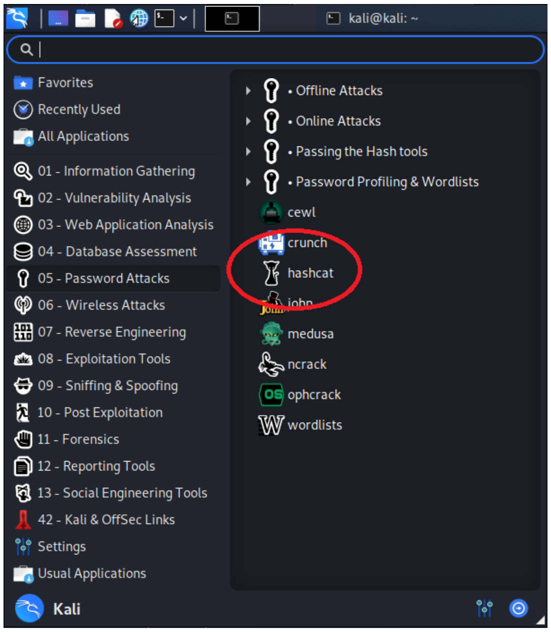
In 2009, an online development company named RockYou was hacked when hackers used a 10 year old SQL vulnerability to access their database. This normally wouldn't be that big of a deal but it was found that they stored all of their usernames and passwords in plaintext. This list of over 32 million user accounts and passwords was released to the public and is now included in every Kali Linux distro, because of the availability, this list is commonly used as a starting point in many password cracking attempts.

CRACKING PASSWORDS WITH HASHCAT

Hashcat is an open source piece of software that allows its users to determine what password is stored in a hash. Even though we cannot reverse the hashing process, we can create a bunch of hashes and compare them to the ones we have. Hashcat does this process automatically and determines if we have one of these common passwords.

STEP 1: STARTING HASHCAT

Open Hashcat by clicking: Start Button -> “05 - Password Attacks” -> hashcat



STEP 2: CRACKING PASSWORDS

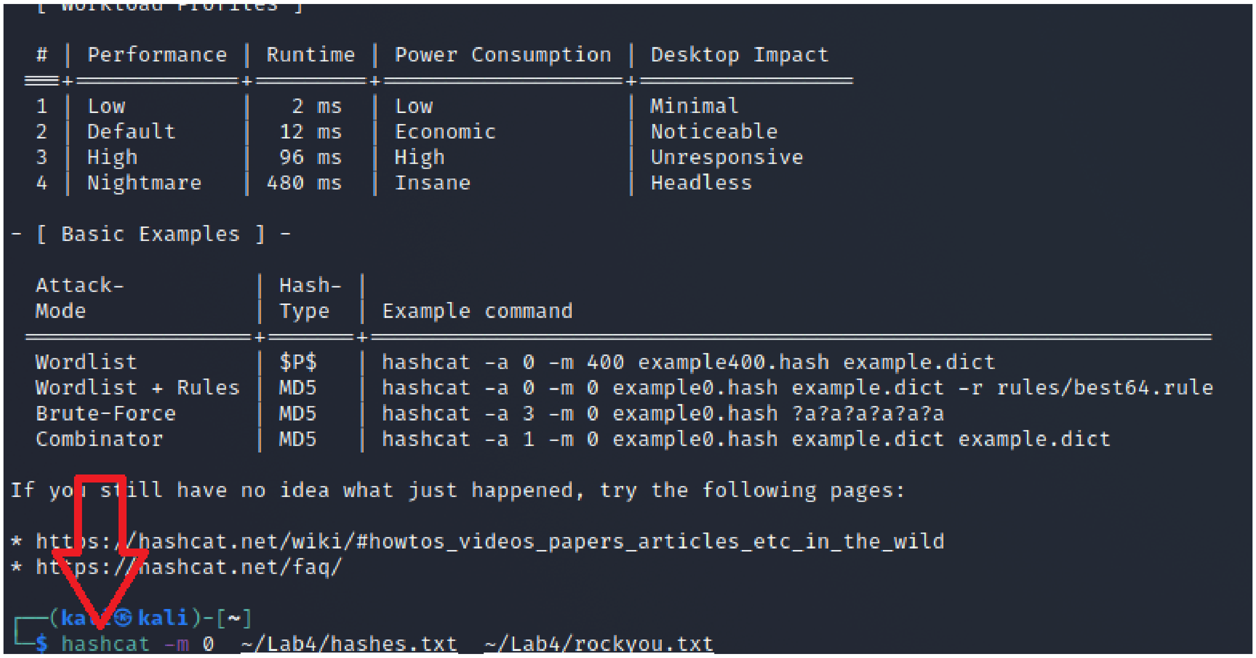
Now that we have hashcat running we can start creating and comparing! To crack our passwords we want to enter the command:

“***hashcat -m 0 ~/Lab4/hashes.txt ~/Lab4/rockyou.txt***”

The meaning of this command is below:

- hashcat - Runs the program
- -m 0 - Specifies mode 0 (MD5)
- ~/Lab4/hashes.txt - Our passwords we will be cracking
- ~/Lab4/rockyou.txt - Our list of previously cracked passwords

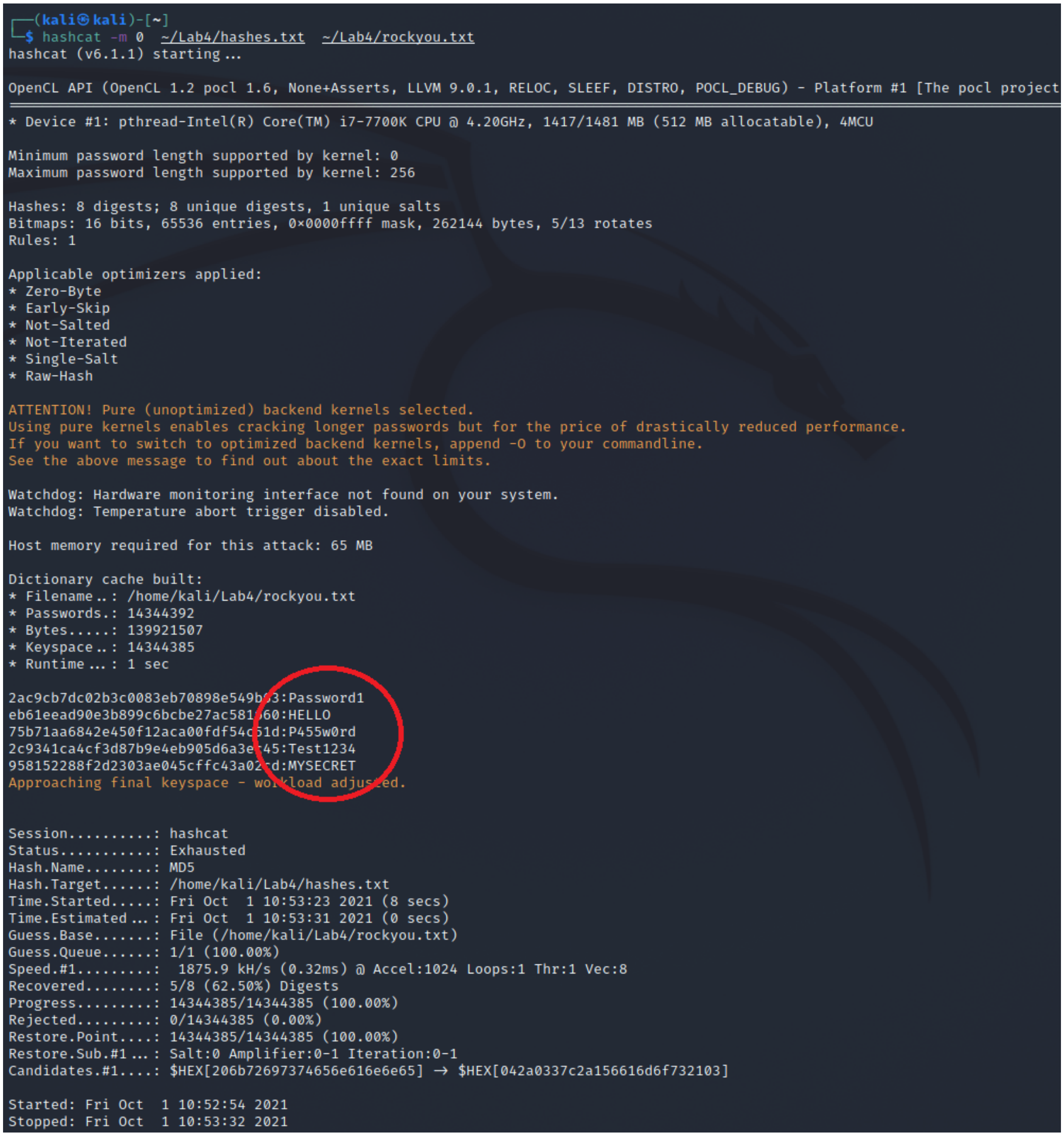
Once you’ve put this command in the command line, hit enter and watch the passwords start flowing in!



```
  [ Workload Profiles ]

  # | Performance | Runtime | Power Consumption | Desktop Impact
 ===+=============+=========+===================+=================
  1 | Low         |   2 ms  | Low               | Minimal
  2 | Default     |  12 ms  | Economic          | Noticeable
  3 | High        |  96 ms  | High              | Unresponsive
  4 | Nightmare   | 480 ms  | Insane            | Headless

- [ Basic Examples ] -

  Attack-          | Hash- |
  Mode             | Type  | Example command
 ==================+=======+==================================================================
  Wordlist         | $P$   | hashcat -a 0 -m 400 example400.hash example.dict
  Wordlist + Rules | MD5   | hashcat -a 0 -m 0 example0.hash example.dict -r rules/best64.rule
  Brute-Force      | MD5   | hashcat -a 3 -m 0 example0.hash ?a?a?a?a?a?a
  Combinator       | MD5   | hashcat -a 1 -m 0 example0.hash example.dict example.dict

If you still have no idea what just happened, try the following pages:

* https://hashcat.net/wiki/#howtos_videos_papers_articles_etc_in_the_wild
* https://hashcat.net/faq/

┌──(kali㉿kali)-[~]
└─$ hashcat -m 0  ~/Lab4/hashes.txt  ~/Lab4/rockyou.txt
```

STEP 3: READING THE PASSWORDS

Once hashcat has finished searching for the passwords we can see the passwords in plain english in the output right next to their hashes. The passwords should be Password1, HELLO, P455w0rd, Test1234, MYSECRET. Now you can decrypt found passwords!



```
┌──(kali㉿kali)-[~]
└─$ hashcat -m 0  ~/Lab4/hashes.txt  ~/Lab4/rockyou.txt
hashcat (v6.1.1) starting...

OpenCL API (OpenCL 1.2 pocl 1.6, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project
==========================================================================================================================
* Device #1: pthread-Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz, 1417/1481 MB (512 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 8 digests; 8 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Using pure kernels enables cracking longer passwords but for the price of drastically reduced performance.
If you want to switch to optimized backend kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 65 MB

Dictionary cache built:
* Filename..: /home/kali/Lab4/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

2ac9cb7dc02b3c0083eb70898e549b63:Password1
eb61eead90e3b899c6bcbe27ac581660:HELLO
75b71aa6842e450f12aca00fdf54c51d:P455w0rd
2c9341ca4cf3d87b9e4eb905d6a3ec45:Test1234
958152288f2d2303ae045cffc43a02cd:MYSECRET
Approaching final keyspace - workload adjusted.


Session..........: hashcat
Status...........: Exhausted
Hash.Name........: MD5
Hash.Target......: /home/kali/Lab4/hashes.txt
Time.Started.....: Fri Oct  1 10:53:23 2021 (8 secs)
Time.Estimated...: Fri Oct  1 10:53:31 2021 (0 secs)
Guess.Base.......: File (/home/kali/Lab4/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:  1875.9 kH/s (0.32ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered........: 5/8 (62.50%) Digests
Progress.........: 14344385/14344385 (100.00%)
Rejected.........: 0/14344385 (0.00%)
Restore.Point....: 14344385/14344385 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: $HEX[206b72697374656e616e6e65] -> $HEX[042a0337c2a156616d6f732103]

Started: Fri Oct  1 10:52:54 2021
Stopped: Fri Oct  1 10:53:32 2021
```

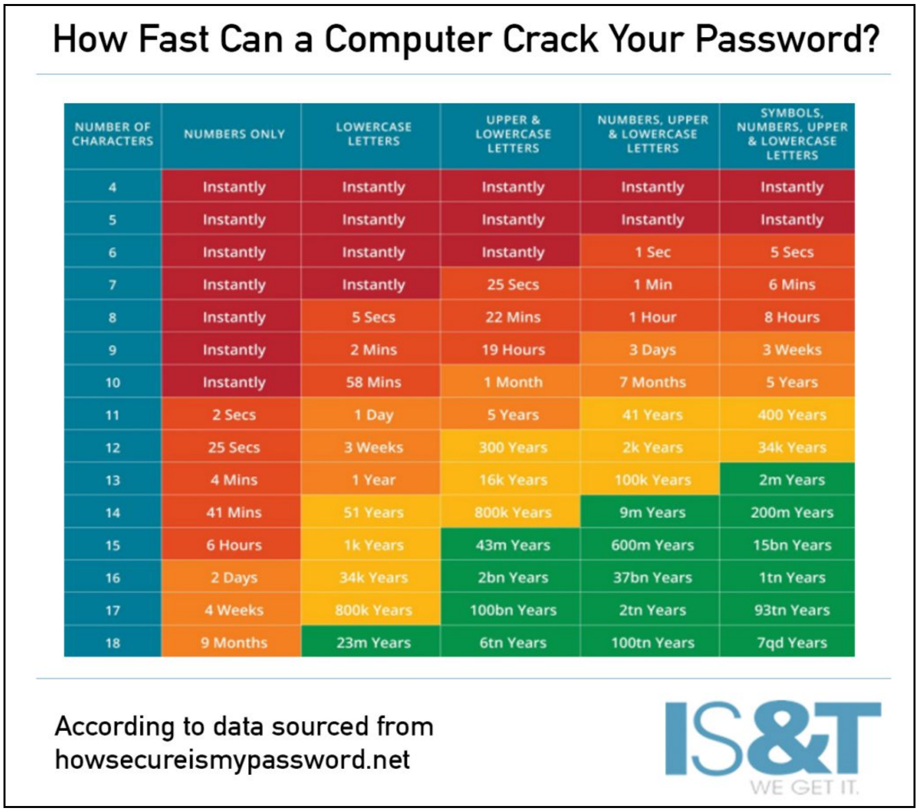
HAVEYOUBEENPWNED?

';--have i been pwned?

Have you ever wondered if your personal data has been compromised? You can find out by visiting haveibeenpwned.com. Have I Been Pwned is a website that allows you to search your own email address to see if any of your data has been leaked as part of a data breach. If any data associated with your email address has been leaked, the website provides detailed information about the breach of information that occurred such as when the breach took place, what types of data were compromised, and the backstory of the breach that took place. This website suggests that you use a password manager to use very strong passwords for different sites (we will be doing this later), and also allows you to get notified if your email becomes involved in any future breaches so that you can take action immediately.

HOW SECURE IS YOUR PASSWORD?

A good way to test the strength of your password is to check it in https://howsecureismypassword.net/. Here you can safely enter in a password and get an estimate on how long it would take a computer to crack that password based on metrics such as password length, types of characters used, and uniqueness of the password.



How Fast Can a Computer Crack Your Password?

NUMBER OF CHARACTERS	NUMBERS ONLY	LOWERCASE LETTERS	UPPER & LOWERCASE LETTERS	NUMBERS, UPPER & LOWERCASE LETTERS	SYMBOLS, NUMBERS, UPPER & LOWERCASE LETTERS
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	1 Sec	5 Secs
7	Instantly	Instantly	25 Secs	1 Min	6 Mins
8	Instantly	5 Secs	22 Mins	1 Hour	8 Hours
9	Instantly	2 Mins	19 Hours	3 Days	3 Weeks
10	Instantly	58 Mins	1 Month	7 Months	5 Years
11	2 Secs	1 Day	5 Years	41 Years	400 Years
12	25 Secs	3 Weeks	300 Years	2k Years	34k Years
13	4 Mins	1 Year	16k Years	100k Years	2m Years
14	41 Mins	51 Years	800k Years	9m Years	200m Years
15	6 Hours	1k Years	43m Years	600m Years	15bn Years
16	2 Days	34k Years	2bn Years	37bn Years	1tn Years
17	4 Weeks	800k Years	100bn Years	2tn Years	93tn Years
18	9 Months	23m Years	6tn Years	100tn Years	7qd Years

According to data sourced from
howsecureismypassword.net

PASSWORD SECURITY TIPS

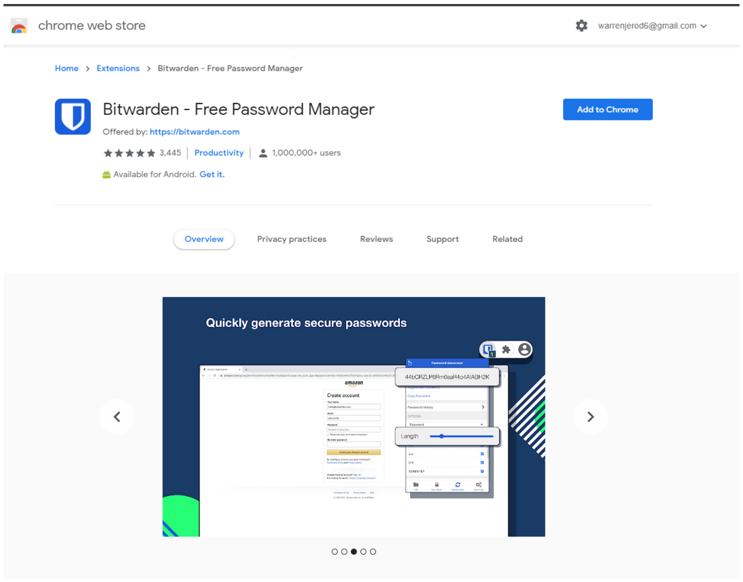
Avoid common mistakes: Never put personal information in your passwords, this includes birthdates, names, addresses, or phone numbers. Make sure your password isn't just one word that can be found in a dictionary, as many attacks will use a dictionary as their pool of guesses. **NEVER** use the same password in more than one place, if this password is cracked, all other accounts with the same password will be compromised as well.

Length and complexity: It is recommended that you use the longest password permissible, the longer and more complex your password is, the harder it will be to crack. Make sure to use a combination of uppercase, lowercase, number and symbols.

Use a password manager: The most optimal way to keep your passwords secure is to use a password manager such as LastPass and Bitwarden. These managers come as a plugin for your browser and allow you to generate super secure passwords and keep them organized by site, account, or application. All of your passwords are secured with end-to-end encryption and are easily accessible in a vault behind a single master password and multifactor authentication. This means you only have to remember a single password, while keeping all your other passwords completely random and impossibly complex.

HOW TO USE A PASSWORD MANAGER

Head on over to the Chrome or Firefox extension store located at chrome.google.com/webstore or addons.mozilla.org and search for BitWarden.



Home > Extensions > Bitwarden - Free Password Manager

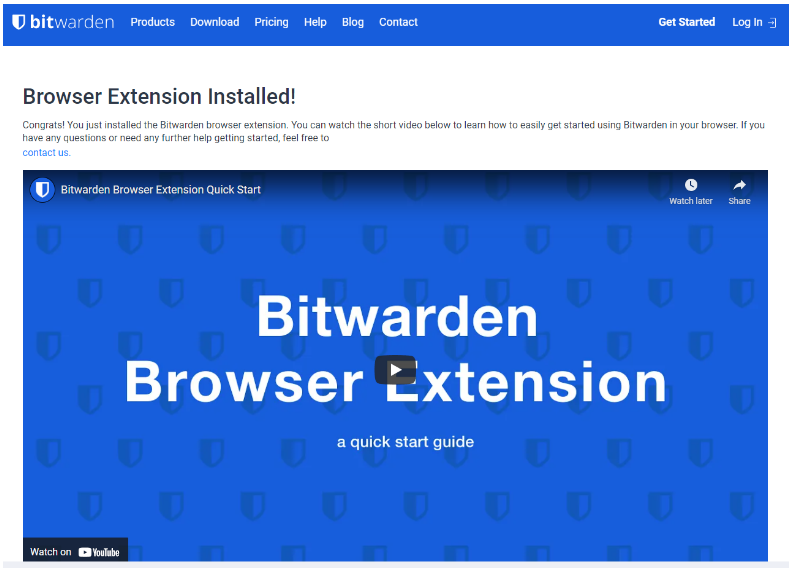
Bitwarden - Free Password Manager

Offered by: https://bitwarden.com

★★★★★ 3,445 | Productivity | 1,000,000+ users

Available for Android. Get it.

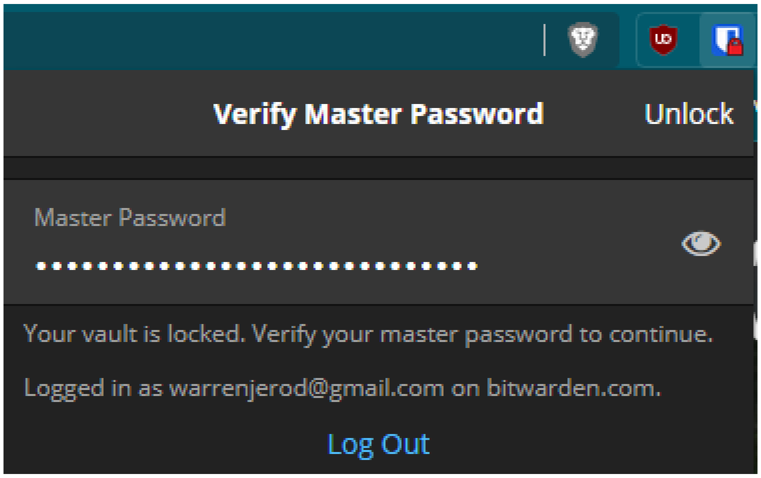
Once you find it just click “Add to Chrome/Firefox”, then click “Add extension” on the pop up. The Bitwarden shield icon will appear in the top right of your browser. If it does not, click the puzzle piece instead, Bitwarden will be in the drop down menu, you can pin it if you like.



Browser Extension Installed!

Congrats! You just installed the Bitwarden browser extension. You can watch the short video below to learn how to easily get started using Bitwarden in your browser. If you have any questions or need any further help getting started, feel free to contact us.

You will also be directed to the Bitwarden home page, where you can watch the quickstart guide. When finished, click “Get Started” in the top right to make an account. Here you will be asked to make a *master password*, this is the password that will allow you to access your vault, make sure it is very secure and something you will always remember.

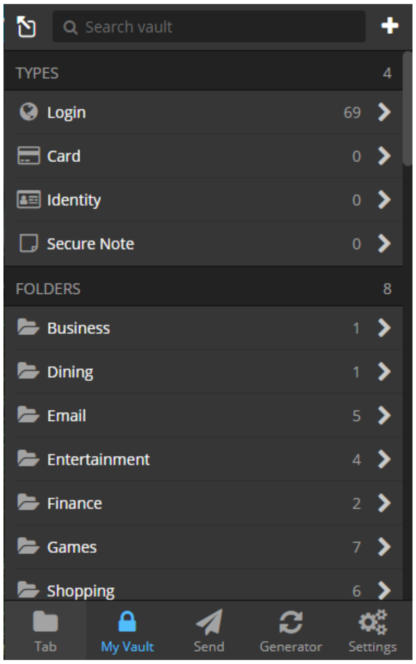


Master Password: ••••••••••••••••••••••••••••••

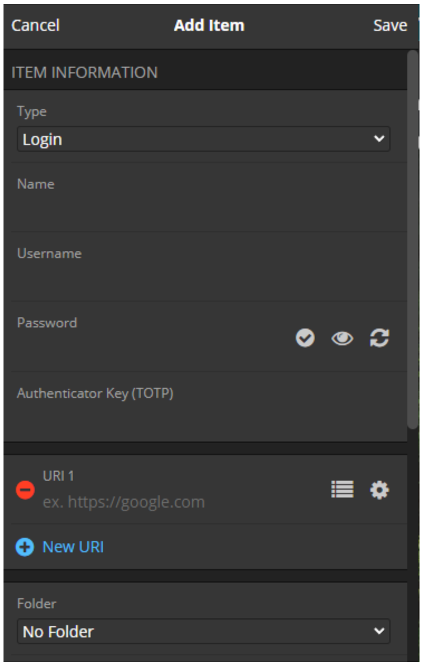
Your vault is locked. Verify your master password to continue.

Logged in as warrenjerod@gmail.com on bitwarden.com.

Click the Bitwarden icon on your browser and enter your Master Password to enter your vault.



TYPES	4
Login	69
Card	0
Identity	0
Secure Note	0
FOLDERS	8
Business	1
Dining	1
Email	5
Entertainment	4
Finance	2
Games	7
Shopping	6

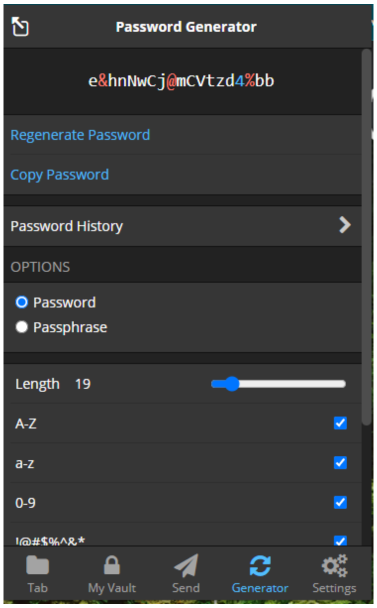


Type: Login

Folder: No Folder

You are now inside your vault, you can click the plus on the top right to enter a new password. Here you can enter the username, password and the website or application for the account.

When logging into an account, you can search for the site and copy the password to your clipboard with one click, so you don't have to worry about remembering and typing out your long and secure passwords ever again.



e&hnNwCj@mCVtzd4%bb

● Password
○ Passphrase

Length 19

A-Z ☑

a-z ☑

0-9 ☑

!@#$%^&* ☑

You can also use the generator at the bottom to make new secure passwords for every site. Here you can choose the length and complexity of your passwords. Make sure you regenerate a new password for every new account.

EXERCISES:

1: Crack a Password:

Earlier in the lab we walked you through cracking the passwords found in ~/Lab4/hashes.txt . Now try cracking the password in ~/Lab4/extrahash.txt . We heard the user that created this password lives in a really big castle!

2: Determine Password Security:

With rockyou containing over 32 million passwords, you have plenty of examples of what not to make your password. For this exercise, we want to see how bad it would be if you were to use one of these passwords found in rockyou. Go to https://www.kaggle.com/wjburns/common-password-list-rockyoutxt and put some of the passwords into https://howsecureismypassword.net/ in order to determine how fast some of these passwords could be cracked. If your password is in this list, CHANGE IT

REVIEW

1: MD5 Stands for

A) Media-Digital algorithm 5

B) Message-Digest algorithm 5

C) Much-Doge algorithm 5

D)My-Data algorithm 5

2: The hashing process can be reversed (True/False)

3: Passwords found in rockyou came from?

A) People giving them away

B) Monkeys on typewriters

C) Bored AI robots

D) Data-breaches

4: What is *not* a good method for creating a strong password?

A) Being over 16 characters

B) Including special characters

C) Reusing a password from another site

D) Using upper and lowercase letters

5: Good password managers can:

A) Create strong passwords

B) Notify of duplicate passwords

C) Securely send passwords

D) All of the above

6: Hashcat can create and compare hashes to determine what password a hash represents (True/False)

Answers: 1:B 2:F 3:D 4:C 5:D 6:T