Ethical Student Hackers

Password Cracking



The Legal Bit

- The skills taught in these sessions allow identification and exploitation of security vulnerabilities in systems. We strive to give you a place to practice legally, and can point you to other places to practice. These skills should not be used on systems where you do not have explicit permission from the owner of the system. It is VERY easy to end up in breach of relevant laws, and we can accept no responsibility for anything you do with the skills learnt here.

- If we have reason to believe that you are utilising these skills against systems where you are not authorised you will be banned from our events, and if necessary the relevant authorities will be alerted.

- Remember, if you have any doubts as to if something is legal or authorised, just don't do it until you are able to confirm you are allowed to.

- Relevant UK Law: https://www.legislation.gov.uk/ukpga/1990/18/contents



Code of Conduct

- Before proceeding past this point you must read and agree to our Code of Conduct - this is a requirement from the University for us to operate as a society.
- If you have any doubts or need anything clarified, please ask a member of the committee.
- Breaching the Code of Conduct = immediate ejection and further consequences.
- Code of Conduct can be found at https://shefesh.com/conduct



Downloading Hashcat

- You will need to download hashcat and rockyou.txt for this session

  https://hashcat.net/hashcat/

  https://shefesh.com/rockyou



How are passwords stored?

- The password you provide is inputted into a hashing algorithm.
- There are various hashing algorithms, for example MD5.
- When a password is inputted, a hashing algorithm (e.g. MD5) produces a hash.

The input of “frog” (the pre-image) to the MD5 algorithm produces this hash (image):
938c2cc0dcc05f2b68c4287040cfcf71

How are passwords stored?

- Password hashes are stored on a server – usually on a database.
- When the user inputs a password, the same hashing algorithm is used, and the hashes are compared.
- If the hash of the inputted password is the same as the hash stored in the database, the password is correct.



What makes hashing secure?

Mostly, one unique input = one unique fixed-length output
Frog will always hash to 938c2cc0dcc05f2b68c4287040cfcf71 when using the MD5 algorithm.

The pre-image cannot *feasibly* be worked out from the output of the algorithm. This is called the pre-image resistance strength.

This is why forget password cannot show you your password



Potential Issues with Hashing

Collisions:

- where two pre-images (inputs) match to the same image(output)
- MD5 is prone to collisions and is the main reason why it is no longer used.

Brute force attack:

- You could try many common passwords to see if any work. This is a valid attack method due to human error/complacency.

Cracking a Hash

- Brute force – run all possible combinations
- Dictionary attacks – run all common combinations
- Hash Table – key-value pairs for passwords and their associated hash, for a particular hashing algorithm
- Rainbow table – in the end a slower version of a hash table, but significantly less storage space requirements

(want to know more? This is a good starting point: https://security.stackexchange.com/questions/92865/what-is-the-difference-between-a-hash-table-and-a-rainbow-table-and-how-are-the)



Bad Passwords

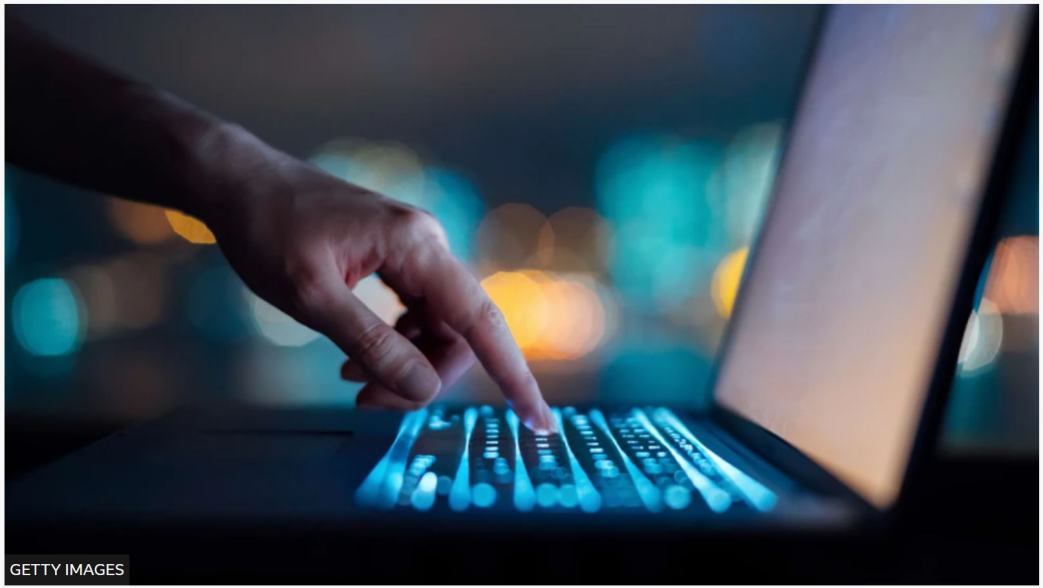
Top 10 most common passwords:

1) 123456
2) admin
3) 12345678
4) 123456789
5) 12345
6) password
7) Aa123456
8) 1234567890
9) Pass@123
10) admin123

https://nordpass.com/most-common-passwords-list/

Bad Passwords

Hackers steal children's data after company used 'Password1' to protect old account



Hackers stole information from Data Cymru's systems in 2024 which exposed the personal information of 3,300 people

Bad Passwords

- Ogi Networks Limited (ONL) and Data Cymru
- A test account “test2” had been left on the system for 12 years by ONL
- Hackers encrypted 182GB of data and stole 42GB of data from Data Cymru
- ONL and Data Cymru found to have failed to follow GDPR by failing to properly protect their systems and regularly check whether security methods were effective.

Bad Passwords

- "Password1 is not an appropriate password, it is a wide-open door to nefarious actors.” - David Doodson, head of investigations at the ICO.

- 59th most common password worldwide, 6th most common in the UK

https://www.bbc.co.uk/news/articles/c6r7y03e9m3ro

Cracking in Practice

Hashcat is a password cracker. The command below will crack a hash:

hashcat -m 0 -a 0 <your hash> /usr/share/wordlists/rockyou.txt

Let's break it down...



hashcat

hashcat -m 0 -a 0 <your hash> /usr/share/wordlists/rockyou.txt

- “hashcat” is the main command for running hashcat



-m (mode)

hashcat -m 0 -a 0 <your hash> /usr/share/wordlists/rockyou.txt

- “-m” (mode) is used to denote the hashing algorithm
- In this example, 0 is evaluated to be MD5
- Other supported algorithms include SHA2-256 (1400) and BitLocker (22100)
- For a full list, see the hashcat wiki: https://hashcat.net/wiki/doku.php?id=hashcat



-a (attack)

hashcat -m 0 -a 0 <your hash> /usr/share/wordlists/rockyou.txt

- “-a” (attack) is used to denote the attack method
- In this example, 0 is evaluated to be a dictionary attack
- Other supported modes include combination (1) and brute force (2)
- For a full list, see the hashcat wiki: https://hashcat.net/wiki/doku.php?id=hashcat



hash

hashcat -m 0 -a 0 <your hash> /usr/share/wordlists/rockyou.txt

- The hash you want to crack



filepath

hashcat -m 0 -a 0 <your hash> /usr/share/wordlists/rockyou.txt

- The absolute path for the rockyou dictionary
- Pre-installed on many linux distros

- Use the file path for rockyou on your own device



Cracking in Practice

hashcat -m 0 -a 0 <your hash> /usr/share/wordlists/rockyou.txt

hashcat – the main command for running hashcat
-m 0 – the '-m' denotes the hash algorithm, 0 is evaluated to be MD5
-a 0 – the '-a' denotes the type of attack, 0 is evaluated to be a dictionary attack
<your hash> -- the hash that you want to crack
/usr/share/wordlists/rockyou.txt – the absolute path for the dictionary that is pre-installed on many linux distros.



Practice Cracking

https://pwck.shefesh.com/



Protecting against attacks - salt

A salt is a random string of characters. This is added to each password before it is hashed.

This protects against brute force attacks as each password now has a random element to it, thus hash and rainbow tables cannot be generated to match them.

You can prepend or append a salt to a password



Protecting against attacks - pepper

A ‘pepper’ is the same as a salt, but not unique for each password.

A website would use the same random string for each password.

A pepper is not stored with the passwords, so is kept secret so if a password is salted and peppered, it cannot be cracked without knowing the pepper.



Disadvantage of Peppering

As a pepper is not unique, it is essential to keep it secret.

If the pepper is discovered, the passwords for ALL accounts using the same pepper must be reset.



Salt and Pepper Examples

Base: frog = 938c2cc0dcc05f2b68c4287040cfcf71

frogsalt = cdb7d225befdabd7ae4aa59588959e64
pepperfrog = bf9784db6a9e2d80cd212faf9b396e29

pepperfrogsalt = 0907d5631a94734ff5d41f776cbd1851



Why Salt AND Pepper?

The salt is usually stored in the database, and is directly associated with a user. If the database is exposed, so are all the salts, effectively rendering them useless.

The pepper is stored separately, and would not be exposed if an outside actor gains access to the database.

salt + pepper = TWO layers of protection



Protecting against attacks - speed

Most algorithms are designed to be fast. Usually as fast as possible.

Hashing is different, as brute force attacks are the only feasible way of cracking a password.

So hashing algorithms are designed to be slow.



Why being slow matters

Salting completely cancels out hash and rainbow tables, along with any other precomputed tables of hashes.

This means the only attack method is computing a hash ‘at runtime’.

Therefore, by having a slow algorithm it takes a very long time for a human/robot to try hundreds or thousands of passwords



Cracking in practice

The command below will crack a **salted** hash:

hashcat -m 10 -a 0 <your hash>:**<your salt>** /usr/share/wordlists/rockyou.txt

hashcat – the main command for running hashcat
-m 10 – the '-m' denotes the hash algorithm, 10 is evaluated to be MD5 + salt
-a 0 – the '-a' denotes the type of attack, 0 is evaluated to be a dictionary attack
<your hash> -- the hash that you want to crack
<your salt> –
/usr/share/wordlists/rockyou.txt – the absolute path for the dictionary that is pre-installed on many linux distros.



Practice Cracking

https://pwck.shefesh.com/ - more challenges available!

If you finish!
https://tryhackme.com/r/room/crackthehash



Upcoming Sessions

What's up next?

www.shefesh.com/sessions

2nd November: South Yorkshire Regional Crime Unit !!! (Membership Required)

Any Questions?



www.shefesh.com

Thanks for coming!