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

- If you don't have a linux distribution, you will need to download hashcat and rockyou.txt for this session

- https://hashcat.net/hashcat/

- https://github.com/topics/rockyou-download

How are passwords stored?

The password you provide is inputted into a hashing algorithm. There are various hashing algorithms, for example MD5.

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

The hash is then stored on a server - usually in a database. When you type your password into a website, it is put through the same hash algorithm, then the hashes are compared.

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)

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

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
use the hashcat wiki to find numbers for each algorithm / attack https://hashcat.net/wiki/doku.php?id=hashcat
<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

Try the first 5 questions!

If you finish! Try 6 and 7 then...
https://tryhackme.com/r/room/crackthehash

Working through No.4

```
awk -b 'length == 4' /usr/share/wordlists/rockyou.txt > new.txt
```

This creates a new file (new.txt) with all the passwords of length equal to 4. This is put wherever your active directory is.

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

Salt and Pepper Examples

Base: frog = 938c2cc0dcc05f2b68c4287040cfcf71

Salted: frog = 585f85938c2cc0dcc05f2b68c4287040cfcf71

Peppered:
frog = 598t5g938c2cc0dcc05f2b68c4287040cfcf71
Salt and Peppered:
robot = 598t5g87b7cb79481f317bde90c116cf36084b47df5df

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 pre-computed 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

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

hashcat -m 0 -a 0 <your hash>:**<your salt>** /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

Try the last 2 questions!

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

Upcoming Sessions

What's up next?

www.shefesh.com/sessions

27th October: Enumeration

Any Questions?

www.shefesh.com

Thanks for coming!