

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/downloads/SESH%20Code%20of%20Conduct.pdf>



How are passwords stored?

- Storing plaintext passwords is very insecure, so almost no one does it
- Most of the time, people hash the passwords
- This is an algorithm that produces a random looking output based on any input
- Similar inputs result in **vastly different outputs**
- The hash is stored on the server
- Passwords are hashed on the client, and then sent to the server to compare
- Plaintext passwords are **never sent over the internet**



Where do we get the hashes?

- Linux `/etc/shadow`
- Mac `/var/db/dslocal/nodes/Default/users/username.plist`
- Windows `%SystemRoot%\system32\config\`

- SSH
- FTP
- Stored in a random file in a website
- SQL injection – get the password field
- Buy a list of them (don't do this its REALLY not clever)



How do we break them?

Brute force:

- Hash a bunch of passwords and see if they match
- Uses a dictionary
- [Rockyou](#) is a famous dictionary
- [Hashcat](#), [john the ripper](#), other online tools
- More time, less memory

Rainbow table:

- Precomputed hashes in a table
- See if a hash matches one in the table
- You already have a password for it with no extra compute
- <https://freerainbowtables.com>
- More memory, less time



Hashcat

Runs through a dictionary, hashes and compares

- hashcat [options]... [hash|hashfile|hccapxfile](#) [dictionary|mask|directory]
- -m [num] → hash type, e.g. [fill in later]
- -o [directory] → output file
- -s [num] → skip to n words from the start of the dictionary
- -D [num] → device ([CPU](#), [GPU](#), [APU](#))

Example usage: hashcat -D 2 -m 0 [Hash/hash.txt](#) Dict/rockyou.txt

<https://hashcat.net/hashcat/>



Rockyou.txt

- A huge dictionary of lots of potential passwords
- We hash each one, checking to see if a password matches
- Roughly 14 million passwords

<https://github.com/zacheller/rockyou/blob/master/rockyou.txt.tar.gz>

Or from the tryhackme page



Salting

- To prevent rainbow table creation, passwords are “salted”
- `hash(password + salt)`
- The salt is passed back to the client and added to the password before the hash
- The salt is stored in plaintext next to the password hash

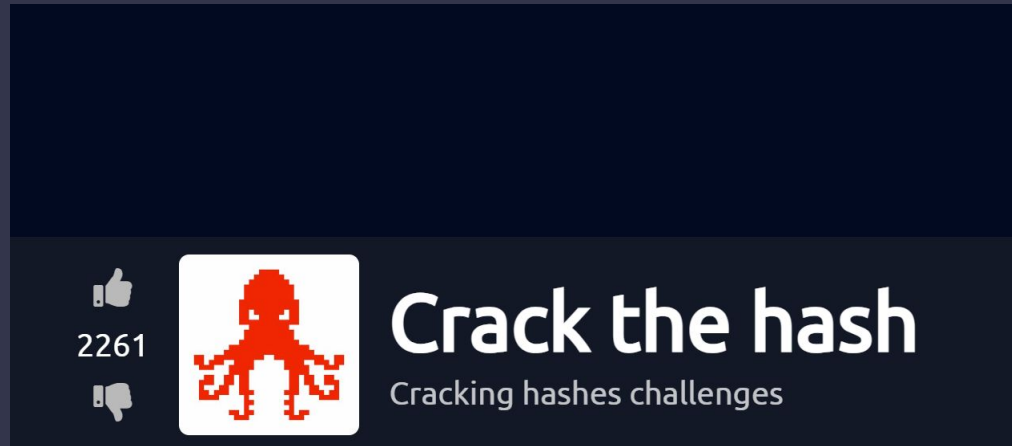
- To get around salting, we brute force
- We add the salt to every item in our dictionary
- Hash all of these new values
- And see if any match

Note: in hashcat, salts are displayed `hash:salt`




Practice

<https://tryhackme.com/room/crackthehash>



A screenshot of a YouTube video player interface. The video title is "Crack the hash" and the description is "Cracking hashes challenges". The video has 2261 likes. The channel name is "Crack the hash" and the channel icon is a red octopus. The video player is currently blank.

2261

 **Crack the hash**
Cracking hashes challenges



Upcoming Sessions

What's up next?

www.shefesh.com/sessions

13th November: ??? - Hicks LT10

20th November: Bounty Hunting - Hicks LT10

Any Questions?



www.shefesh.com
Thanks for coming!

