# Ethical Student Hackers

## Advanced 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 **shefesh.com/conduct**

# Acknowledgement

This session / talk is heavily inspired by a talk from Steelcon by Will Hunt - Plundering and Pillaging Password and Passphrase Plains for Profit

# Prerequisites

- Basic Linux command line knowledge
- Hashcat
- Password cracking

Key tidbits:

MD5: useless
bcrypt: strong

# Hashcat Recap

hashcat -m <hash-type> -a <attack-mode> hash.txt wordlist.txt [mask]
hashcat -m 0 -a 0 hashes.txt rockyou.txt

Modes:

0 -> dictionary
3 -> brute force
6 -> wordlist + mask (append) ?d?d?d -> password000
7 -> mask + wordlist (prepend) ?d?d?d -> 000password

?d?l?u?s?a -> digit, lowercase, uppercase, symbol, any

# Rules

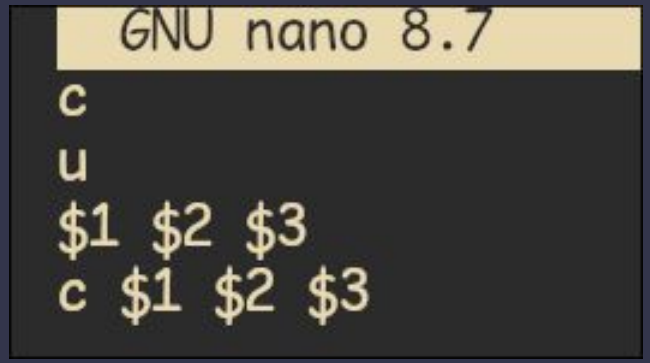
We can modify our wordlist at runtime to make it more powerful

Add to your command “-r ruleList.rules”

https://github.com/wolframalpha/hashcat/blob/master/rules/best64.rule

ONLY WORKS ON MODE 0

# Rules





--stdout flag to output the results of the rules

https://hashcat.net/wiki/doku.php?id=rule_based_attack

# Combinators and Delimiters

CombinatorX, PrinceProcessor

Hashcat -m0 hashes.txt -a1 wordlist.txt wordlist2.txt -w3 -o
Can be more powerful, add delimiters, add rules. Using --stdout

Using combinator tools, you can do combinator attacks far more powerfully than raw hashcat.

https://github.com/Sudo-Aju/CombinatorX

https://github.com/hashcat/princeprocessor

I will leave you to look into these.

# Hash Shucking

Some websites use MD5

What happens when you upgrade?

Update when the user next logs in? (password stored insecurely until then)

Hash the hash: SHA256{MD5{password}}

Can we find the password?

# Hash Shucking

If we have (from a previous breach) a set of (e.g.) MD5 Hashes

And from a new breach, a set of bcrypt{MD5} hashes

We can use use the MD5 hashes as a wordlist

And if any match, we can crack the MD5 to get the password!

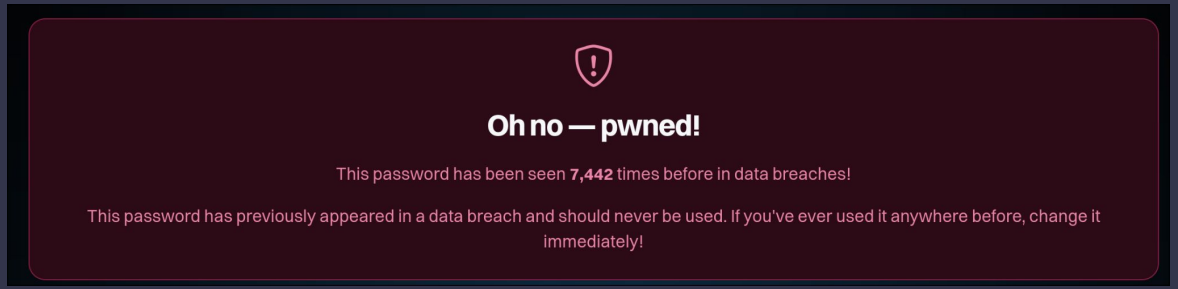
Some people WILLINGLY will SHA256/MD5 their password!!! Idiots.

What are problems with this?

# Transliteration

ji32k7au4a83

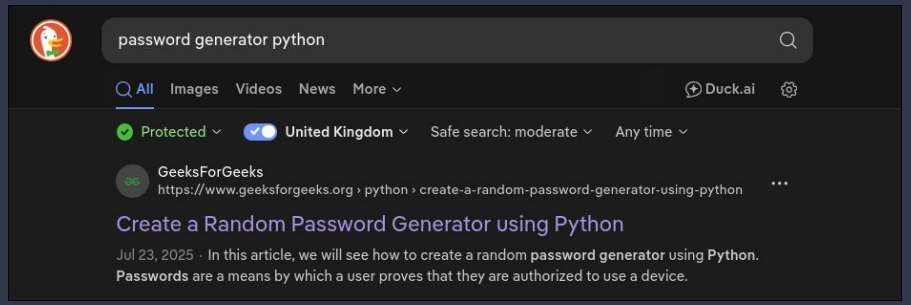
ji3 -> m
2k7 -> y
au4 -> pass
a83 -> word



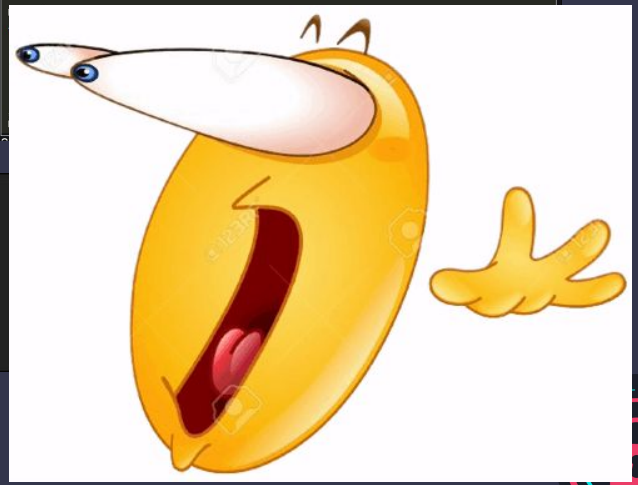

Zhuyin / BoPoMoFo keyboard

# Stupid Passwords

Using python “random”




```
import string
import random

# Getting password length
length = int(input("Enter password length: "))

print('''Choose character set for password from these :
        1. Digits
        2. Letters
        3. Special characters
        4. Exit''')

characterList = ""

# Getting character set for password
while(True):
    choice = int(input("Pick a number "))
    if(choice == 1):

        # Adding letters to possible characters
        characterList += string.ascii_letters
    elif(choice == 2):

        # Adding digits to possible characters
        characterList += string.digits
    elif(choice == 3):

        # Adding special characters to possible
        # characters
        characterList += string.punctuation
    elif(choice == 4):
        break
    else:
        print("Please pick a valid option!")
```



```
random.seed(a=None, version=2)
```

Initialize the random number generator.

If *a* is omitted or `None`, the current system time is used. If randomness sources are provided by the operating system, they are used instead of the system time (see the `os.urandom()` function for details on availability).

# Practical

1) $2a$12$ixp2x4HzHxKFC5zsy.RrYuO3ZXcXy8crCedvxzbzWKp9XviyYffhG
2) 1e69d4314f9b9439cac0acb9ac5179f52a1be059b11de6672e7545f56180d840
3) 5B338CF333708D59751561431BDA046E
4) $2a$12$kMHdRXVtp5HTi7nI9UUCOus/emSlh3LW0GC1ehOQ1aaDhvHd3KQNy
5) 5493a8316373df21e0226c11a26f0c31
6) 81ad37eec22086a3680370e2454b6a9e

https://github.com/HaveIBeenPwned/PwnedPasswordsDownloader

# Feedback

Please leave your feedback :) We want to know what we can do to improve.

Please leave constructive and honest feedback only.

https://forms.gle/VTYd74K5BHqbC7F68

# Inclusions Concerns

If there's anything preventing you from enjoying our sessions, please let our Inclusions Officer know. You can contact them by email or fill in the form below:

jgledhill2@sheffield.ac.uk

https://forms.gle/Qct6Wyfesv8dWmej7

# Upcoming Sessions

What's up next?

www.shefesh.com/sessions

Emails (TBC)

Cryptography

# Any Questions?


www.shefesh.com

Thanks for coming!

# Starting Soon!


www.shefesh.com

Please take a seat