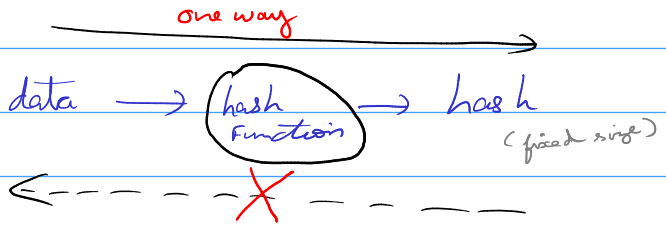
Hashing



this hashes are cracked by the help of rainbow table.
↳ Contains hashes which matches to data before hashed.

Tools:

mostly used is

* hashcat (built for use of GPU)
* John the ripper. (built for use of CPU)

John: Use man John (or) john --help

↳ Automated cracking — john --wordlist=[path to wordlist] [path to file]

↳ Format specific cracking. — john --format=[format] --wordlist=[path to wordlist] [path to file]

Commands:-

* john --list=formats → to show the list of formats supported by John
* john --show → TO show the cracked hash

To find the type of hash use hash-identifier tool

Cracking Windows Passwords:-

note:

Window's passwords are hashed in NTLM → Type of MD4 hash.

These hashes are dumped using tools like mimikatz.

Windows passwords are cracked using John as,

john --format=NT --wordlist=<wordlist> <hash file>

Cracking Unix Passwords :-

Here's a quick table of the most Unix style password prefixes that you'll see.

Prefix	Algorithm
1	md5crypt, used in Cisco stuff and older Linux/Unix systems
2, $2a$, $2b$, $2x$, $2y$	Bcrypt (Popular for web applications)
6	sha512crypt (Default for most Linux/Unix systems)

A great place to find more hash formats and password prefixes is the hashcat example page, available here: https://hashcat.net/wiki/doku.php?id=example_hashes.

Files that related to passwords :-

↳ /etc/passwd → stores the users of Linux System (include system services)

↳ /etc/shadow → stores the password hashes of all users

TO crack Linux password,

i) Unshadowing

↳ combine /etc/passwd with /etc/shadow to tell the john what data we given, for this john has unshadow.

Usage : unshadow [path to passwd] [path to shadow] >output.txt

eg) unshadow /etc/passwd /etc/shadow > unshadowed.txt

ii) Cracking :

Usage :- john --wordlist=/usr/share/wordlists/rockyou.txt --format=sha512crypt unshadowed.txt

depends on systems

Single crack mode :

↳ without using wordlist, John uses word mangling
(makes wordlist based on the given input.)

Usage :

↳ john --single --format=[format] [path to file]

If we take the username: Markus

Some possible passwords could be:

Markus1, Markus2, Markus3 (etc.)
MArkus, MARkus, MARKus (etc.)
Markus!, Markus$, Markus* (etc.)

Note :

If we have a hash abcd..... in hash.txt file, then we have to change the file to mike:abcd..... if mike is username.

Custom rules :

↳ If an organisation has password policy which includes,

* a capital letter
* special characters
* Numbers.

Then it should be exploited by using custom rules.

Creating custom rules :

↳ Custom rules are created in /etc/john/john.conf (or) /opt/john/john.conf based on the installation.

name our rules as [List.Rules.Rule_NAME]

We then use a regex style pattern match to define where in the word will be modified,

Az - Takes the word and appends it with the characters you define

A0 - Takes the word and prepends it with the characters you define

c - Capitalises the character positionally

Lastly, we then need to define what characters should be appended, prepended or otherwise included, we do this by adding character sets in square brackets [ ] in the order they should be used. These directly follow the modifier patterns inside of double quotes " ".

eg.)

[0-9] - Will include numbers 0-9

[0] - Will include only the number 0

[A-z] - Will include both upper and lowercase

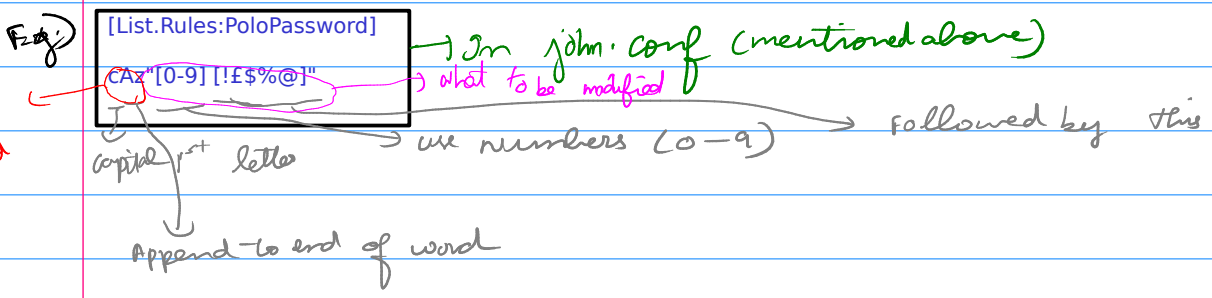
[A-Z] - Will include only uppercase letters

[a-z] - Will include only lowercase letters

[a] - Will include only a

[!£$%@] - Will include the symbols !£$%@

Putting this all together, in order to generate a wordlist from the rules that would match the example password "Polopassword1!" (assuming the word polopassword was in our wordlist) we would create a rule entry that looks like this:



Then, we use this Rule in John as,

Usage: john --wordlist=[path to wordlist] --rule=PoloPassword [path to file]

Cracking zip File Password:

→ Like unshadow, we use a tool from John suite **zip2john**

not necessary

Usage: zip2john [options] [zip file] > [output file]

we make our zip file to hash john understood

eg) zip2john new.zip > hash.txt

→ Then we crack it.

Usage: john --wordlist=/usr/share/wordlists/rockyou.txt zip_hash.txt

Cracking RAR Password :

↳ Here, we use Rar2 john (similar to Zip2john).

Usage : rar2john [rar file] > [output file]

"By, we are making rar file to format which john understands

Usage : john --wordlist=/usr/share/wordlists/rockyou.txt rar_hash.txt

Cracking SSH Password :

↳ In Pentest, when log in to SSH, sometimes key file is enough (i.e) SSH user@12.13.1.1 -i <private key file>

↳ But, sometimes it requires a id-rsa password for login.

SSH2 john :

you can find the Python script of ssh2john.py at /usr/share/john/ssh2john.py

Usage : ssh2john [id_rsa private key file] > [output file]

Eg.) ssh2john id_rsa > id-rsa-hash.txt

Usage : john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa_hash.txt

References :

https://www.openwall.com/john/