Cracking

- [Exploitation/Cracking] Mots de passe
- [Exploitation/Cracking] Hash
- [Cracking] Wordlists
- [Exploitation/Cracking] Rainbow Tables
- [Exploitation/Cracking] Wifi

[Exploitation/Cracking] Mots de passe

Introduction

Cette page est dédiée au crackage de mots de passe avec des outils et des procédures.



Psudohash

Cet outil permet de générer des mutations d'un mot de passe.

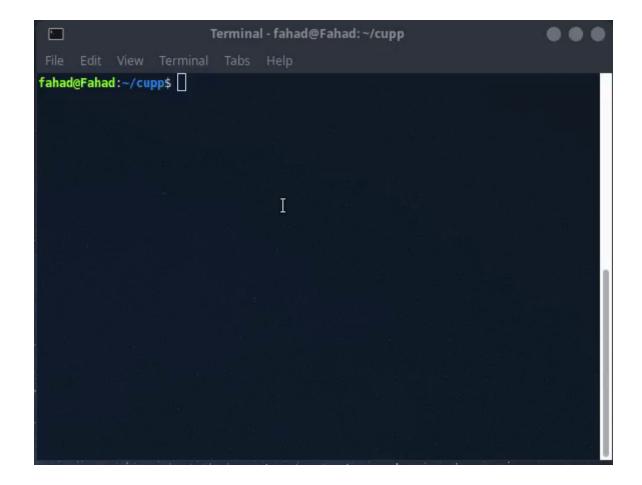
• https://github.com/t3l3machus/psudohash

```
:/opt/psudohash# ./psudohash.py -w microsoft --common-paddings-after -y 2020-2023
                          by t3l3machus
[Info] Calculating output length and size...
[Warning] This operation will produce 49040640 words, 834.6 MB. Are you sure you want to proceed? [y/n]: y
[*] Mutating keyword: microsoft
   - Producing character case-based transformations...
   Mutating word based on commonly used char-to-symbol and char-to-number substitutions...
   Appending year patterns after each word mutation...
   Appending common paddings after each word mutation...
 L Done!
[Info] Completed! List saved in output.txt
          nachus:/opt/psudohash# grep 'm!Cr0sofT123' output.txt
m!Cr0sofT123
m!Cr0sofT1234
m!Cr0sofT12345
m!Cr0sofT123456
m!Cr0sofT123!@#
               :/opt/psudohash# grep 'M1Cr0$of7!' output.txt
M1Cr0$of7!@#
M1Cr0$of7!@!
M1Cr0$of7!@#$%
M1Cr0$of7!
M1Cr0$of7!!
M1Cr0$of7!!!
M1Cr0$of7!@
```

Cupp

Permet de générer des mots de passes pour cibler une personne ou une entité en établissant le profil de ce dernier.

https://github.com/Mebus/cupp



Crunch

Permet de générer des mots de passe selon un patern et des propriétés comme une longueur spécifique ou l'utilisation de caractères spéciaux.

• https://github.com/jim3ma/crunch

Hydra



Certainement le plus grand outil de brute force, il permet d'attaquer divers services et des pages de login web.

HTTP Get

hydra -l \$USER -P <WORDLIST> -f <TARGET> http-get-form
"<LOGIN_URL>:<PARAMETER>=<VALUE>:S=<SUCCESS_CONDITION>" -f

Exemple:

hydra -l admin -P /resources/rockyou.txt -f cozyhosting.htb http-get-form "/login:username=^USER^&password=^PASS^:S=logout.php" -f

HTTP Post

hydra -l \$USER -P <WORDLIST> -f <TARGET> http-post-form <LOGIN_URL>:<PARAMETER>=<VALUE>

Exemple:

hydra -l admin -P /resources/rockyou.txt -f cozyhosting.htb http-post-form

"/login:username=admin&password=^PASS^:Invalid username or password"

SSH

hydra -l <USER> -P <WORDLIST> -s 22 <IP> ssh

Exemple:

hydra -l michael -P /resources/rockyou.txt -s 22 10.10.243.36 ssh

WFuzz



Outil de fuzzing de sous domaines mais ausside brute force de formulaires sur des pages web.

Exemple d'utilisation pour brute force un login sur une page web :

wfuzz -c -z file,<WORDLIST> --hs <MOT_ECHEC_PASS_PAGE> -d "<login>=<LOGIN>&<password>=FUZZ" <URL>

Fusionner des wordlists

Cela peut être utile lorsque vous avez générer des wordlists avec différents outils avec différents critères.

Plusieurs techniques sont possibles mais certaines sont plus optimisées.

Par ailleurs, il faut supprimer les doublons pour encore optimiser et réduire la taille de la wordlist de le temps de cracking.

Cat

L'outil cat permet de fusionner de manière basique deux fichiers textes :

cat file1.txt file2.txt file3.txt > combined_list.txt

On peut ensuite supprimer les doublons avec la commande suivante :

Duplicut

Cet outil permet de fusionner vos wordlist de manière optimisée :

https://github.com/nil0x42/duplicut

Username generator

Parfois, vous n'avez pas en votre possession le nom d'utilisateur de votre cible.

Cependant, avec son nom et son prénom vous pouvez créer une wordlist grâce à Username-Anarchy :

https://github.com/urbanadventurer/username-anarchy

CeWL

Cet outil permet de générer une wordlist à partir d'un site web :

https://github.com/digininja/CeWL

Bash

Avec du scripting bash vous pouvez générer des wordlists.

Voici un exemple de script :

```
#!/bin/bash

for year in {2020..2021};

do

[for char in '!' '@' '#' '$' '%' '^' '&' '*' '(")';

[do

[echo "Fall${year}${char}";

[done;

done > psw.lst
```

Cette technique fonctionne bien lorsque vous avez une idée précise du pattern que doitav le mot de passe cible.	oir/

[Exploitation/Cracking] Hash

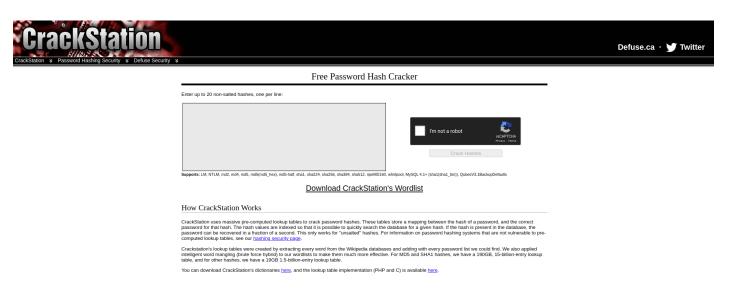
Introduction

Cette page présente plusieurs outils et méthodologies pour casser des hashs.

Crackstation

C'est le site web de référence pour cracker vos hashs en tout genre.

https://crackstation.net/

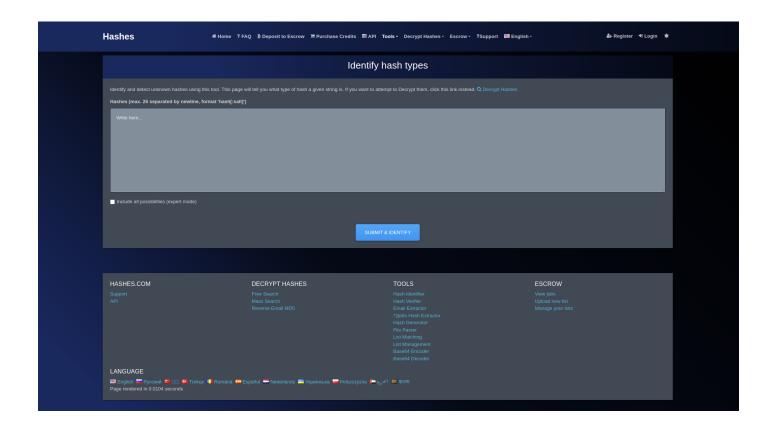


Hashes.com

Cet outil en ligne est un peu similaire à crackstation mais plus performant.

S'il n'arrive pas à cracker le hash, il vous donnera au moins le type de hash (ce qui est pratique pour pouvoir lancer une attaque avec hashcat ou john par la suite).

https://hashes.com/en/tools/hash identifier



Name That Hash

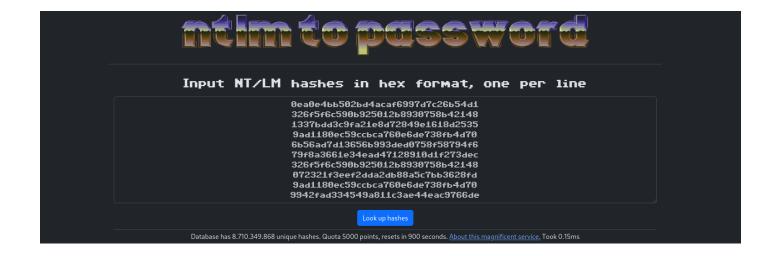
Ce site permet d'identifier le type de hash :

https://nth.skerritt.blog/

NTLM.pw

Ce site permet de casser vos hashs NTLM presque instantannément. Il est composé d'une base de donnée de 8,7 milliards de hashs donc vous devriez réussir à casser vos hashs tranquillement.

https://ntlm.pw/



JohnTheRipper

C'est un incontournable pour les attaques brute force à partir de wordlist.

Il supporte plusieurs formats comme des archives zip ou rar et même des mots de passes



Password Unix

Saisir la ligne de l'utilisateur concerné du fichier /etc/passwd dans le fichier passwd.txt :

root:x:0:0:root:/root:/bin/bash

Saisir la ligne de l'utilisateur concerné du fichier /etc/shadow dans le fichier shadow.txt :

root:\$6\$riekpK4m\$uBdaAyK0j9WfMzvcSKYVfyEHGtBfnfpiVbYbzbVmfbneEbo0wSijW1GQussvJSk8X1M56kzgGj8f7DFN1h4dy1:18226:0:99999:7:::

Obtenir le fichier unshadowed.txt qui sera approprié pour casser le hash avec john :

unshadow passwd.txt shadow.txt > unshadowed.txt

Puis lancer l'attaque avec john :

john --wordlist=/usr/share/wordlists/rockyou.txt unshadowed.txt

Mot de passe archive ZIP

Obtenir le hash du mot de passe de l'archive grâce à zip2john :

zip2john archive.zip > hash.txt

Lancer l'attaque:

john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

Mot de passe archive RAR

Obtenir le hash du mot de passe de l'archive grâce à zip2john :

zip2john archive.zip > hash.txt

Lancer l'attaque :

john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

Phrase de passe SSH

Obtenir le hash du mot de passe de l'archive grâce à zip2john :

ssh2john archive.zip > hash.txt

Lancer l'attaque:

john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

Mot de passe mySQL

Si vous parvenez à extraire des mots de passe d'utilisateurs dans une table de base de donnée mySQL, vous pouvez l'attaquer avec john :

john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt

Génération de wordlist

John permet la génération d'une wordlist mutée (des mutations vont être générées selon des règles) :

```
john --wordlist=<LIST> --rules=<RULE> --stdout > <OUTPUT>
```

Par exemple:

```
john --wordlist=clinic.lst --rules=Single-Extra --stdout > custom-02.lst
```

Vous pouvez lister les règles disponibles avec la commande suivante :

```
cat /etc/john/john.conf|grep "List.Rules:" | cut -d"." -f3 | cut -d":" -f2 | cut -d"]" -f1 | awk NF
```

L'emplacement du fichier de configuration de john peut varier selon votre système.

Sur Exegol, il est situé dans /opt/tools/john/run/

Vous pouvez créer vos propres règles en ajoutant ce type de configuration dans le fichier :

[List.Rules:Example-01]

Az"[0-9]" ^[!@#\$]

- Az représente un mot de la wordlist originale.
- "[0-9]" représente un chiffre.
- ^[!@#\$] représente l'utilisation de caractères spéciaux à la fin de la chaîne (ici les caractères ! @ # et \$). Utiliser \$ à la place de ^ mettra le caractère spécial au début de la chaîne.

HashID

Cet outil en ligne de commande permet d'identifier le type de hash auquel vous faites face.

Pour l'installer sur les distributions basées sur Debian :

sudo apt install -y hashid

Et voici la syntaxe :

hashid -m <HASH>

Hashcat

Il s'agit d'un outil similaire à John mais ayant une option permettant d'utiliser la puissance GPU pour décupler les performances et essayer un nombre bien plus grand de combinaisons par seconde.

Il est vivement conseillé de l'utiliser sur Windows avec les cartes graphiques Nvidia puisque les pilotes ne sont pas pleinement supportés sous Linux et réduisent donc la puissance de calcul utilisable.



Syntaxe globale

hashcat --hash-type <NUMBER> --attack-mode 0 <HASH_FILE> <PASSWORD_LIST>

Le type de hash varie selon le type de mot de passe que vous essayez de casser.

Vous pouvez retrouver la liste ici :

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

OPHcrack



Cet outil disponible sur Windows et Linux permet de lancer des attaques **Rainbow tables** sur les hashs.

Ce type d'attaque consiste à prendre des listes de mots de passe avec leur hash associé (appelées rainbow tables), et de comparer ce hash avec celui à cracker. Si le hash correspond, cela veut dire que l'on a trouvé le mot de passe.

Cependant, ce type d'attaque ne marche que sur certains algorithmes de hashages faibles tel que **LM**, **NTML** ou **MD5** car ils n'utilisent pas de salage.

Voici le site officiel du projet où vous pourrez télécharger le logiciel ainsi que des rainbow tables :

https://ophcrack.sourceforge.io/

Si vous souhaitez télécharger d'autres rainbow tables, vous pouvez vous rendre sur le site suivant :

http://project-rainbowcrack.com/

[Cracking] Wordlists

Introduction

Quelques wordlists très pratiques pour faire de l'énumération de site web ou du cracking de mots de passe.



Wordlists

RockYou

• https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt

Stun

• https://korben.info/une-liste-de-15-milliards-de-mots-de-passe.html

Fuzzing Dirbuster

• <u>Small</u>: <u>https://github.com/daviddias/node-dirbuster/blob/master/lists/directory-list-2.3-small.txt</u>

- <u>Medium</u>: <u>https://github.com/daviddias/node-dirbuster/blob/master/lists/directory-list-2.3-medium.txt</u>
- Big : https://github.com/daviddias/node-dirbuster/blob/master/lists/directory-list-2.3-big.txt

SecLists

Plusieurs listes intéressantes d'utilisateurs, mots de passe, sous-répertoire web etc :

• https://github.com/danielmiessler/SecLists

Richelieu (french wordlist)

• https://github.com/tarraschk/richelieu

Kaonashi

• https://github.com/kaonashi-passwords/Kaonashi

PacketStormSecurity (wordlist par pays)

• https://packetstormsecurity.com/Crackers/wordlists/page4/

Awesome wordlists

https://github.com/gmelodie/awesome-wordlists

[Exploitation/Cracking] Rainbow Tables

Introduction

Les **rainbow tables** sont des bases de données pré-calculés de hash qui permettent de gagner un temps colossal lors des attaques par force brute.

Il s'agit en outre d'une table associant un mot de passe avec son hash ce qui évite de devoir recalculé le hash lors de l'attaque brute force.

Cependant, ce type d'attaque ne fonctionne plus lorsqu'un algorithme utilisant le salage est utilisé car pour un même mot de passe, plusieurs hashs seront valides (en fonction du salage bien sûr).



Algorithme vulnérables

Voici la liste des algorithmes de hashage vulnérables :

- NTLM
- SHA-1
- MD5
- LM
- Half-I M

Télécharger des rainbow tables

• Free Rainbow Tables

[Exploitation/Cracking] Wifi

Introduction

Cette page décrit des méthodologies et des outils pour compromettre des réseaux utilisant le protocole Wifi.

Aircrack-ng

Certainement l'outil le plus célèbre pour pirater des réseaux wifi, il est aussi celui demandant le plus de ressources.

Il requiert une carte wifi supportant l'injection de paquet et le mode moniteur et une puissance de calcul suffisante pour casser le mot de passe par la suite.

En effet, l'objectif va être d'envoyer un paquet de *désauthentification* pour récupérer un **handshake** contenant le hash du mot de passe du wifi qu'on appelle **psk**.

Une fois le hash capturé, il ne reste plus qu'à lancer une attaque hors-ligne sur celui-ci pour obtenir le mot de passe en clair.

Le temps de cracking peut être plus ou moins long selon la puissance de calcul disponible et la complexité du mot de passe.



Fern



WiFiPumpkin3

Anciennement nommé EvilGinx, cet outil est en mesure de créer des faux points d'accès wifi permettant de piéger des utilisateurs du réseau wifi cible en espérant récupérer le mot de passe du réseau wifi par des attaques phishing exécutées en local.

