Proof of concepts :
Recherche de collision de Hash

Prérequis

name-that-hash
→ pip3 install name-that-hash

Hashcat
→ apt install hashcat

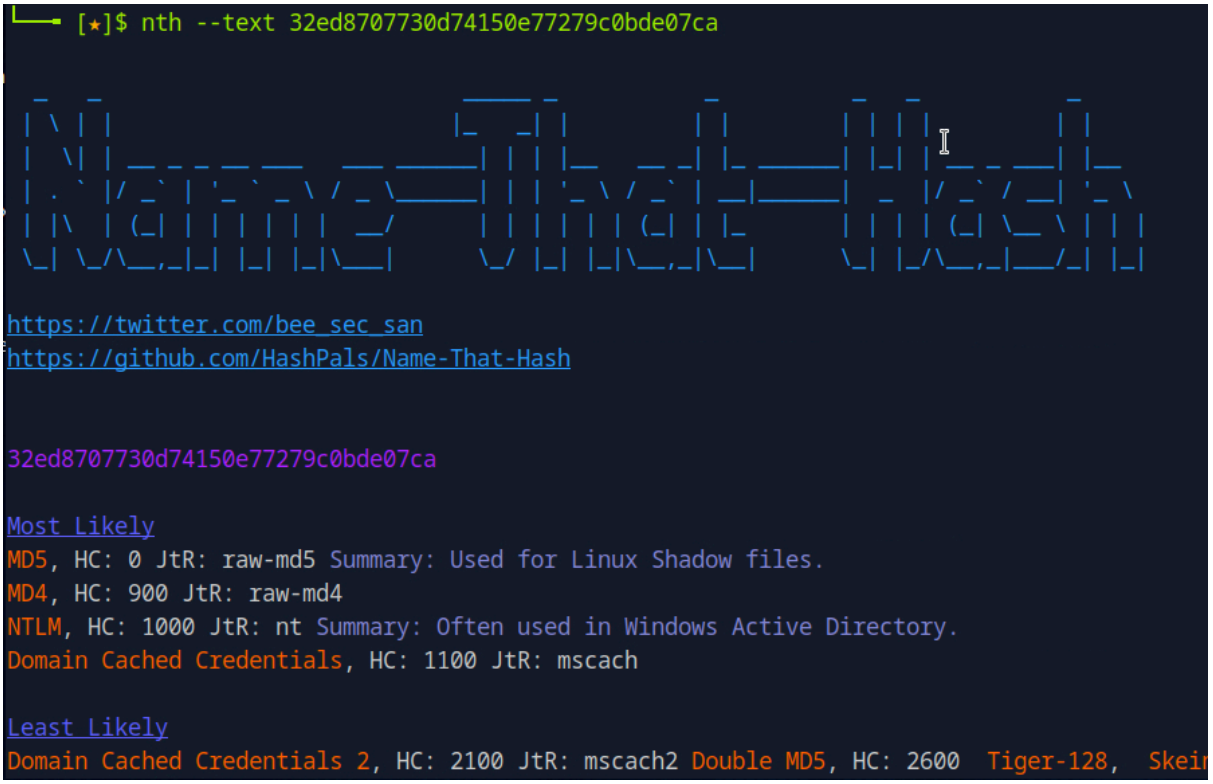
JohnTheRipper
→ apt install john

Mot choisi pour le test :
test
→ md5 : 098f6bcd4621d373cade4e832627b4f6

Procédure

1) Récupération du format

Name-that-hash



Maintenant nous savons que ce hash est surement en MD5.
Name-That-Hash nous donne directement le flag a utiliser (HC: 0), sinon :
hashcat -h

2) Brute Force

2.1) Hashcat

Hashcat fonctionne de la façon suivante :
hashcat -m [ModeID] [HASH] [Wordlist]

- -m : définit le mode que l'on veut utiliser, 0 pour MD5, 1000 pour du NTLM etc…
- HASH : le hash que l'on as en amont. Peut êtres contenue dans un .txt
- Wordlist : liste de mots que l'on veut utiliser pendant le brute force.

```
098f6bcd4621d373cade4e832627b4f6:test

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 098f6bcd4621d373cade4e832627b4f6
Time.Started.....: Wed Feb 25 08:25:21 2026 (0 secs)
Time.Estimated...: Wed Feb 25 08:25:21 2026 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (Documents/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#2.........:  6492.3 kH/s (0.13ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 167936/14344385 (1.17%)
Rejected.........: 0/167936 (0.00%)
Restore.Point....: 165888/14344385 (1.16%)
Restore.Sub.#2...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#2....: tyson4 -> playpen

Started: Wed Feb 25 08:25:20 2026
Stopped: Wed Feb 25 08:25:23 2026
```

On retrouve bien notre “test”, on peut voir que pour un mot aussi court, hashcat a pris 3 secondes.

2.2) John the ripper

John the ripper est plus “intelligent” et peut souvent détecter lui-même le format du hash.
Il fonctionne de la façon suivante :
john –wordlist=[wordlist] [hashfile] (--format=[format])

- wordlist : liste de mots que l’on veut utiliser pendant le brute force.
- hashfile : John as besoin que le hash soit stocker dans un fichier txt.

Si il ne trouve pas le type de hash que vous lui avez donner, ajouter : --format=[format]

- format : type de hash que vous lui donnez.

```
└─╼ [★]$ echo "098f6bcd4621d373cade4e832627b4f6" > hash.txt

└─╼ [★]$ cat hash.txt
098f6bcd4621d373cade4e832627b4f6

└─╼ [★]$ john --format=raw-md5 --wordlist=Documents/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
test             (?)
1g 0:00:00:00 DONE (2026-02-25 08:59) 50.00g/s 8313Kp/s 8313Kc/s 8313KC/s tyson4..tauruz
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Remédiation

1) Utilisez des versions modernes (argon2, bcrypt…)

Ces algorithmes sont plus complexes, nécessitent plus de puissance pour être calculés et de ce fait, sont plus chronophage lors des brutes force.

2) Ajouter “salt” et “pepper”

Ajoutez une valeur aléatoire au hash existant, ainsi, 2 utilisateurs ayant un même mot de passe auront un hash différent.
Salt : valeur ajoutée directement au hash
Pepper : clé secrète stocker ailleurs

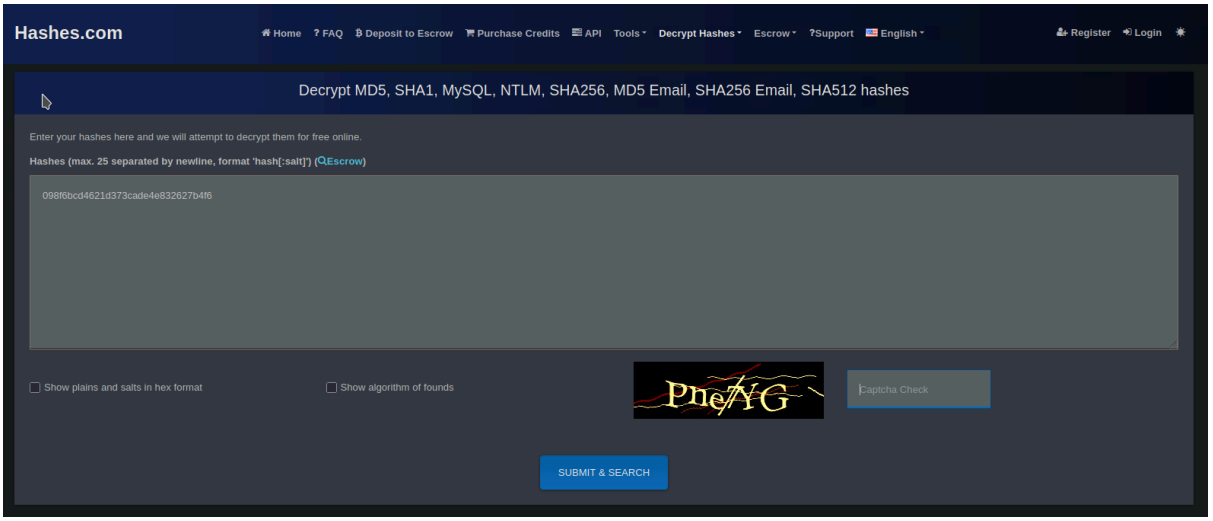
3) MFA

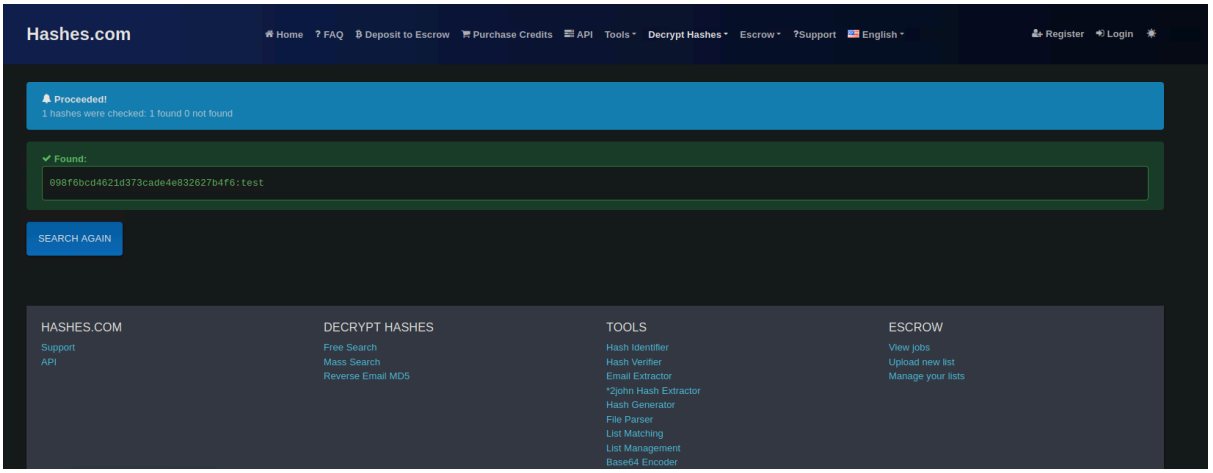
Evidemment. Même si le mot de passe venait à être cassé, l'utilisation de facteurs multiples compliquerait grandement l’authentification.

Autres

Outils en ligne

[Hashes.com](https://hashes.com)





[Crackstation.net](https://crackstation.net)