Usage of HASHCAT

The project consists of two text(.txt) files

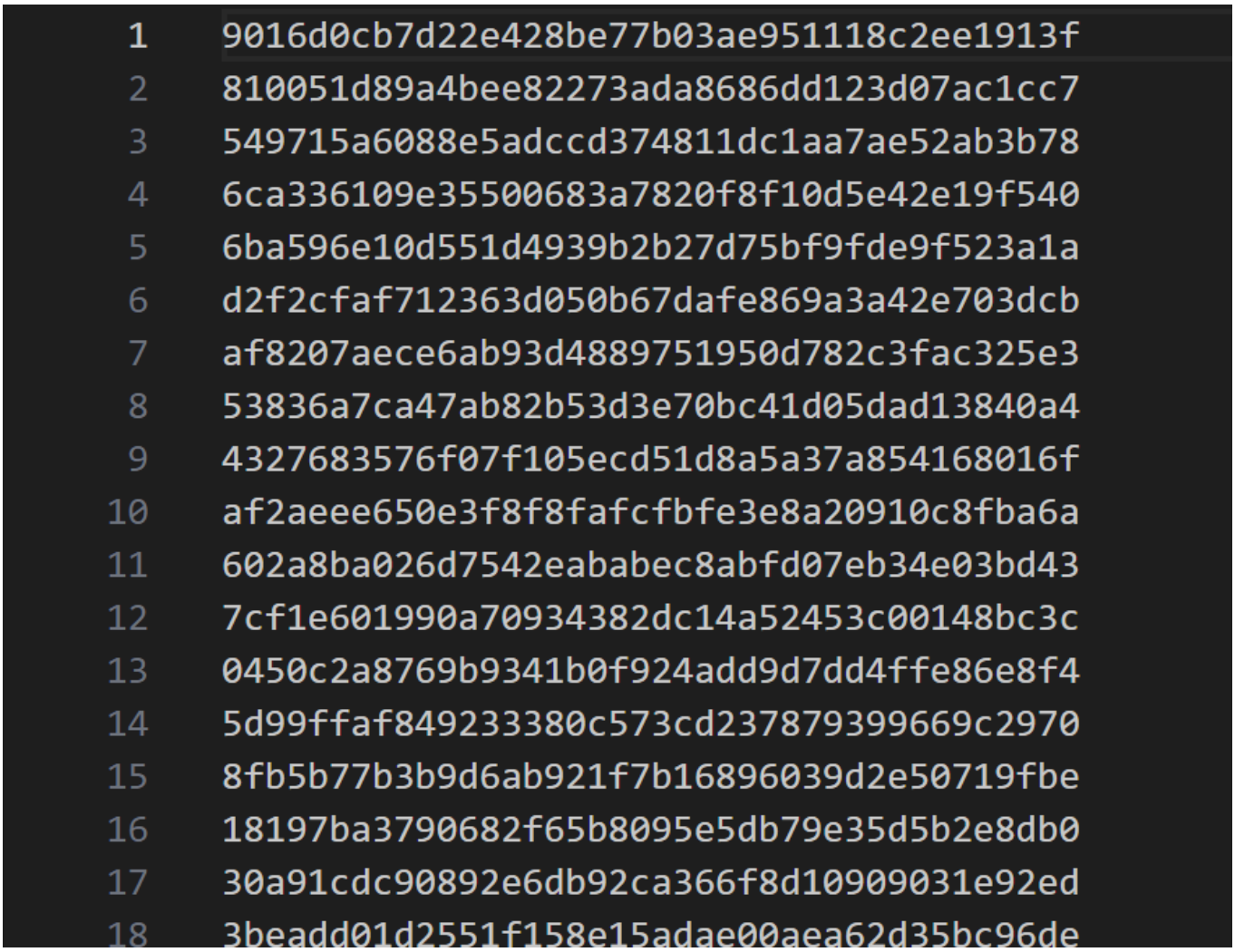
dictionary.txt

This file consists a list of 47603 passwords made of english alphabets

```
'spring
'buster+'
++++
+monday
+rose
+rosemg    "rosemg": Unknown word.
----
-abril    "abril": Unknown word.
-ali
-amcld    "amcld": Unknown word.
-asce    "asce": Unknown word.
-assd    "assd": Unknown word.
-aug
-beloi    "beloi": Unknown word.
-beloit
-boston
-charl    "charl": Unknown word.
-dan
-desig    "desig": Unknown word.
-emh
-ever
-ignet    "ignet": Unknown word.
-jacs    "jacs": Unknown word.
-love
-mar
-march
-nov
-oct
-param
-patriot
-pens
-piv-
-year
=bart
=damian
=dublin
=nov
=oct
```

hashes.txt

For the respective passwords in dictionary.txt, hashes.txt consists of hashed password we got after hashing the passwords using SHA-1 Algorithm



recovered.txt

After running HASHCAT on hashes.txt and dictionary.txt, here's what we get:

```
636f5dbc72c1a3d15a8005bf85e3c58fc76ca95d:alyssa
aef1a877b7d98f0272ecf9326eacd002f2890331:alyssab    "alyssa
91bb850e8a411b83cd5662e117d457406ecb78c1:alyssad    "alyssa
91059c2e24104c575b8968cca4df5a6ce7eac458:alyssak    "alyssa
e7db7332c269553b6566f3ed74c6d518ee6b9424:alyssam    "alyssa
af0e9fb3353d3aa6d43818d3e5d9d22ee5b154a2:alyssas    "alyssa
203d2071a3026e3281e772709575894b1229d3a9:alysse    "alysse"
e68558bc56c50a15dc92806d8135c0e66e92c901:alyssia    "alyssi
35b5a3d5e653d52115c6dccdf741e8dcf7c5de02:amaa    "amaa": Ur
6aec8e9125279a8b5229ff09a7eccd46443c4d2e:amaama    "amaama"
607503d3b0e5c60e3c7aa92d39fc541fec623059:amab    "amab": Ur
e033aff4209e9fa62fd11042a6be65044dbe658d:amable    "amable"
bba4d2e285812a355f740a4b2dc4deab72a15ef5:amac    "amac": Ur
8ab3f2ccf50720826f294f45bd471c3e61e329a0:amad    "amad": Ur
de3ece35d710816cd0441a75269a4bb13d5895d3:amada    "amada":
cc8caff161a4c910313fb796a47f735f30c15cb4:amadeo    "amadeo"
ebd6ca3abb01f628385fa8ba6227d076719a97be:amader    "amader"
c2139c65e0b627c867ad32317e4e51b5203b60c9:amadeus
```

hash.py

```python
import hashlib

with open("dictionary.txt", "r") as f:
    passwords = f.read().splitlines()

with open("hashes.txt", "w") as f:
    for password in passwords:
        sha1_hash = hashlib.sha1(password.encode()).hexdigest()
        f.write(sha1_hash + "\n")

total_hashes = len(passwords)
```

```
cracked_hashes = set()



with open("recovered.txt", "r") as f:

    for line in f:

        if ":" in line:

            hash_part = line.split(":")[0].strip()

            cracked_hashes.add(hash_part)


success_rate = (len(cracked_hashes) / total_hashes) * 100



print(f"Total hashes: {total_hashes}")

print(f"Unique hashes cracked: {len(cracked_hashes)}")

print(f"Success rate: {success_rate:.2f}%")
```

Command to use HASHCAT

```
C:\hashcat-6.2.6>hashcat.exe -a 0 -m 100 "D:\SNU\6th Sem\FIS\Hashcat\hashes.txt" "D:\SNU\6th Sem\FIS\Hashcat\dictionary.
txt" --outfile="D:\SNU\6th Sem\FIS\Hashcat\recovered.txt" --force
```

```
Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1172 MB

Dictionary cache built:
* Filename..: D:\SNU\6th Sem\FIS\Hashcat\dictionary.txt
* Passwords.: 47603
* Bytes.....: 322069
* Keyspace..: 47603
* Runtime...: 0 secs
```

```
Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 100 (SHA1)
Hash.Target......: D:\SNU\6th Sem\FIS\Hashcat\hashes.txt
Time.Started.....: Sun Mar 30 23:45:27 2025, (10 mins, 5 secs)
Time.Estimated...: Sun Mar 30 23:55:32 2025, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (D:\SNU\6th Sem\FIS\Hashcat\dictionary.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   323.0 kH/s (0.20ms) @ Accel:512 Loops:1 Thr:64 Vec:1
Speed.#2.........:      375 H/s (1.74ms) @ Accel:128 Loops:1 Thr:64 Vec:1
Speed.#*.........:   323.4 kH/s
Recovered........: 47603/47603 (100.00%) Digests (total), 47603/47603 (100.00%) Digests (new)
Remaining........: 0 (0.00%) Digests
Recovered/Time...: CUR:0,N/A,N/A AVG:702.08,N/A,N/A (Min,Hour,Day)
Progress.........: 47603/47603 (100.00%)
Rejected.........: 0/47603 (0.00%)
Restore.Point....: 0/47603 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Restore.Sub.#2...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: "spring -> beltran
Candidates.#2....: beltre -> ~bruins
Hardware.Mon.#1..: Temp: 55c Util:  7% Core: 990MHz Mem:5000MHz Bus:4
Hardware.Mon.#2..: N/A
```

Success rate in recovering hashed password

```
PS D:\SNU\6th Sem\FIS\Hashcat> & C:/Users
6th Sem/FIS/Hashcat/hash.py"
Total hashes: 47603
Unique hashes cracked: 47603
Success rate: 100.00%
PS D:\SNU\6th Sem\FIS\Hashcat>
```