Łamanie haseł - Dictionary attack

4.1 *Pod adresem* ***http://127.0.0.1:4001*** *działa prosty serwer HTTP udostępniający dwa endpointy:* ***http://127.0.0.1:4001/hash*** *oraz* ***http://127.0.0.1:4001/submit*** *.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4001:4001 --name ex1 docker.io/mazurkatarzyna/pass-cracking-ex1:latest
podman run -p 4001:4001 --name ex1 docker.io/mazurkatarzyna/pass-cracking-ex1:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4001:4001 --name ex1 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex1:latest
podman run -p 4001:4001 --name ex1 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex1:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4001/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash MD5** do złamania (jako `hash`).

(c) Wykorzystując narzędzia `hashcat` oraz `John the Ripper` i dostępne **słowniki**, spróbuj złamać hash.

(d) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4001/submit` wartość złamanego hasha (jako `word`).

(e) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.2 *Pod adresem* ***http://127.0.0.1:4002*** *działa prosty serwer HTTP udostępniający dwa endpointy:* ***http://127.0.0.1:4002/hash*** *oraz* ***http://127.0.0.1:4002/submit*** *.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4002:4002 --name ex2 docker.io/mazurkatarzyna/pass-cracking-ex2:latest
podman run -p 4002:4002 --name ex2 docker.io/mazurkatarzyna/pass-cracking-ex2:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4002:4002 --name ex2 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex2:latest
podman run -p 4002:4002 --name ex2 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex2:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4002/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-1** do złamania (jako `hash`).

(c) Wykorzystując narzędzia `hashcat` oraz `John the Ripper` i dostępne **słowniki**, spróbuj złamać hash.

(d) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4002/submit` wartość złamanego hasha (jako `word`).

(e) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.3 *Pod adresem* ***http://127.0.0.1:4003*** *działa prosty serwer HTTP udostępniający dwa endpointy:* ***http://127.0.0.1:4003/hash*** *oraz* ***http://127.0.0.1:4003/submit*** *.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4003:4003 --name ex3 docker.io/mazurkatarzyna/pass-cracking-ex3:latest
podman run -p 4003:4003 --name ex3 docker.io/mazurkatarzyna/pass-cracking-ex3:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4003:4003 --name ex3 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex3:latest
podman run -p 4003:4003 --name ex3 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex3:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4003/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-256** do złamania (jako `hash`).

(c) Wykorzystując narzędzia `hashcat` oraz `John the Ripper` i dostępne **słowniki**, spróbuj złamać hash.

(d) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4003/submit` wartość złamanego hasha (jako `word`).

(e) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.4 *Pod adresem* ***http://127.0.0.1:4004*** *działa prosty serwer HTTP udostępniający dwa endpointy:* ***http://127.0.0.1:4004/hash*** *oraz* ***http://127.0.0.1:4004/submit*** *.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4004:4004 --name ex4 docker.io/mazurkatarzyna/pass-cracking-ex4:latest
podman run -p 4004:4004 --name ex4 docker.io/mazurkatarzyna/pass-cracking-ex4:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4004:4004 --name ex4 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex4:latest
podman run -p 4004:4004 --name ex4 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex4:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4004/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-512** do złamania (jako `hash`).

(c) Wykorzystując narzędzia `hashcat` oraz `John the Ripper` i dostępne **słowniki**, spróbuj złamać hash.

(d) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4004/submit` wartość złamanego hasha (jako `word`).

(e) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.5 *Pod adresem* `http://127.0.0.1:4005` *działa prosty serwer HTTP udostępniający dwa endpointy:* `http://127.0.0.1:4005/hash` *oraz* `http://127.0.0.1:4005/submit`.

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4005:4005 --name ex5 docker.io/mazurkatarzyna/pass-cracking-ex5:latest
podman run -p 4005:4005 --name ex5 docker.io/mazurkatarzyna/pass-cracking-ex5:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4005:4005 --name ex5 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex5:latest
podman run -p 4005:4005 --name ex5 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex5:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4005/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash bcrypt** do złamania (jako `hash`).

(c) Wykorzystując narzędzia `hashcat` oraz `John the Ripper` i dostępne **słowniki**, spróbuj złamać hash.

(d) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4005/submit` wartość złamanego hasha (jako `word`).

(e) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.6 *Pod adresem* `http://127.0.0.1:4006` *działa prosty serwer HTTP udostępniający dwa endpointy:* `http://127.0.0.1:4006/hash` *oraz* `http://127.0.0.1:4006/submit`.

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4006:4006 --name ex6 docker.io/mazurkatarzyna/pass-cracking-ex6:latest
podman run -p 4006:4006 --name ex6 docker.io/mazurkatarzyna/pass-cracking-ex6:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4006:4006 --name ex6 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex6:latest
podman run -p 4006:4006 --name ex6 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex6:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4006/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash scrypt** do złamania (jako `hash`).

(c) Wykorzystując narzędzia `hashcat` oraz `John the Ripper` i dostępne **słowniki**, spróbuj złamać hash.

(d) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4006/submit` wartość złamanego hasha (jako `word`).

(e) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.7 *Pod adresem* ***http://127.0.0.1:4007*** *działa prosty serwer HTTP udostępniający dwa endpointy:* ***http://127.0.0.1:4007/hash*** *oraz* ***http://127.0.0.1:4007/submit*** *.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4007:4007 --name ex7 docker.io/mazurkatarzyna/pass-cracking-ex7:latest
podman run -p 4007:4007 --name ex7 docker.io/mazurkatarzyna/pass-cracking-ex7:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4007:4007 --name ex7 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex7:latest
podman run -p 4007:4007 --name ex7 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex7:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4007/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash MD5** do złamania (jako `hash`).

(c) Wiedząc, że **hasło składa się z 3 znaków, z których każdy jest cyfrą od 0 do 9**, za pomocą narzędzia `crunch` wygeneruj słownik, którego użyjesz do złamania hasha

(d) Do złamania hasha użyj narzędzia `hashcat` lub `John the Ripper`.

(e) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4007/submit` wartość złamanego hasha (jako `word`).

(f) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.8 *Pod adresem* ***http://127.0.0.1:4008*** *działa prosty serwer HTTP udostępniający dwa endpointy:* ***http://127.0.0.1:4008/hash*** *oraz* ***http://127.0.0.1:4008/submit*** *.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4008:4008 --name ex8 docker.io/mazurkatarzyna/pass-cracking-ex8:latest
podman run -p 4008:4008 --name ex8 docker.io/mazurkatarzyna/pass-cracking-ex8:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4008:4008 --name ex8 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex8:latest
podman run -p 4008:4008 --name ex8 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex8:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4008/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash MD5** do złamania (jako `hash`).

(c) Wiedząc, że **hasło składa się z 4 znaków, z których każdy jest wielką literą**, za pomocą narzędzia `crunch` wygeneruj słownik, którego użyjesz do złamania hasha.

(d) Do złamania hasha użyj narzędzia `hashcat` lub `John the Ripper`.

(e) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4008/submit` wartość złamanego hasha (jako `word`).

(f) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.9 *Pod adresem* ***http://127.0.0.1:4009*** *działa prosty serwer HTTP udostępniający dwa endpointy:* ***http://127.0.0.1:4009/hash*** *oraz* ***http://127.0.0.1:4009/submit*** *.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4009:4009 --name ex9 docker.io/mazurkatarzyna/pass-cracking-ex9:latest
podman run -p 4009:4009 --name ex9 docker.io/mazurkatarzyna/pass-cracking-ex9:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4009:4009 --name ex9 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex9:latest
podman run -p 4009:4009 --name ex9 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex9:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4009/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash MD5** do złamania (jako `hash`).

(c) Wiedząc, że **hasło to słowo `pass` z sufiksem składającym się z 4 cyfr od 0 do 9, czyli np. `pass0000`, `pass0001`, ...** , za pomocą narzędzia `crunch` wygeneruj słownik, którego użyjesz do złamania hasha.

(d) Do złamania hasha użyj narzędzia `hashcat` lub `John the Ripper`.

(e) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4009/submit` wartość złamanego hasha (jako `word`).

(f) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.10 *Pod adresem* ***http://127.0.0.1:4010*** *działa prosty serwer HTTP udostępniający dwa endpointy:* ***http://127.0.0.1:4010/hash*** *oraz* ***http://127.0.0.1:4010/submit*** *.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4010:4010 --name ex10 docker.io/mazurkatarzyna/pass-cracking-ex10:latest
podman run -p 4010:4010 --name ex10 docker.io/mazurkatarzyna/pass-cracking-ex10:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4010:4010 --name ex10 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex10:latest
podman run -p 4010:4010 --name ex10 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex10:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4010/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash MD5** do złamania (jako `hash`).

(c) Wiedząc, że **hasło to permutacje słów `admin`, `password` oraz `123`**, za pomocą narzędzia `crunch` wygeneruj słownik, którego użyjesz do złamania hasha.

(d) Do złamania hasha użyj narzędzia `hashcat` lub `John the Ripper`.

(e) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4010/submit` wartość złamanego hasha (jako `word`).

(f) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.11 *Pod adresem* `http://127.0.0.1:4011` *działa prosty serwer HTTP udostępniający dwa endpointy:* `http://127.0.0.1:4011/hash` *oraz* `http://127.0.0.1:4011/submit`.

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4011:4011 --name ex11 docker.io/mazurkatarzyna/pass-cracking-ex11:latest
podman run -p 4011:4011 --name ex11 docker.io/mazurkatarzyna/pass-cracking-ex11:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4011:4011 --name ex11 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex11:latest
podman run -p 4011:4011 --name ex11 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex11:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4011/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash MD5** do złamania (jako `hash`).

(c) Wiedząc, że **hasło składa się z 5 znaków, gdzie:**

- pierwszy znak to wielka litera,
- drugi znak to mała litera,
- trzeci znak to cyfra od 0 do 9,
- czwarty i piąty znak to znaki specjalne (czyli np. `!`, `@`, `#`, `$` itp.),

za pomocą narzędzia `crunch` wygeneruj słownik, którego użyjesz do złamania hasha.

(d) Do złamania hasha użyj narzędzia `hashcat` lub `John the Ripper`.

(e) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4011/submit` wartość złamanego hasha (jako `word`).

(f) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

Łamanie haseł - Brute-force (Mask attack)

4.12 *Pod adresem* `http://127.0.0.1:4012` *działa prosty serwer HTTP udostępniający dwa endpointy:* `http://127.0.0.1:4012/hash` *oraz* `http://127.0.0.1:4012/submit`.

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4012:4012 --name ex12 docker.io/mazurkatarzyna/pass-cracking-ex12:latest
podman run -p 4012:4012 --name ex12 docker.io/mazurkatarzyna/pass-cracking-ex12:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4012:4012 --name ex12 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex12:latest
podman run -p 4012:4012 --name ex12 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex12:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4012/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-1** do złamania (jako `hash`).

(c) Wiedząc, że **hasło składa się z 4 znaków, gdzie każdy z nich jest małą literą od `a` do `z`**, złam hash.

(d) Nie generuj własnych, ani nie używaj gotowych słowników. Wykorzystaj metodę bruteforce, definiując wzorzec hasła używając odpowiedniej maski. (Maski to wzorce definiujące strukturę hasła. Zamiast sprawdzać wszystkie możliwe kombinacje, możesz określić dokładny format hasła.)

(e) Do złamania hasha użyj narzędzia `hashcat` lub `John the Ripper`.

(f) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4012/submit` wartość złamanego hasha (jako `word`).

(g) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.13 *Pod adresem `http://127.0.0.1:4013` działa prosty serwer HTTP udostępniający dwa endpointy: `http://127.0.0.1:4013/hash` oraz `http://127.0.0.1:4013/submit`.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4013:4013 --name ex13 docker.io/mazurkatarzyna/pass-cracking-ex13:latest
podman run -p 4013:4013 --name ex13 docker.io/mazurkatarzyna/pass-cracking-ex13:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4013:4013 --name ex13 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex13:latest
podman run -p 4013:4013 --name ex13 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex13:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4013/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-1** do złamania (jako `hash`).

(c) Wiedząc, że **hasło składa się z 4 znaków, gdzie pierwszy z nich jest wielką literą (od `A` do `Z`), a reszta cyframi od 0 do 9**, złam hash.

(d) Nie generuj własnych, ani nie używaj gotowych słowników. Wykorzystaj metodę bruteforce, definiując wzorzec hasła używając odpowiedniej maski. (Maski to wzorce definiujące strukturę hasła. Zamiast sprawdzać wszystkie możliwe kombinacje, możesz określić dokładny format hasła.)

(e) Do złamania hasha użyj narzędzia hashcat lub John the Ripper.

(f) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4013/submit` wartość złamanego hasha (jako `word`).

(g) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.14 *Pod adresem `http://127.0.0.1:4014` działa prosty serwer HTTP udostępniający dwa endpointy: `http://127.0.0.1:4014/hash` oraz `http://127.0.0.1:4014/submit`.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4014:4014 --name ex14 docker.io/mazurkatarzyna/pass-cracking-ex14:latest
podman run -p 4014:4014 --name ex14 docker.io/mazurkatarzyna/pass-cracking-ex14:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4014:4014 --name ex14 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex14:latest
podman run -p 4014:4014 --name ex14 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex14:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4014/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-1** do złamania (jako `hash`).

(c) Wiedząc, że **hasło składa się z 3 znaków, gdzie:**

- pierwszy znak hasła to litera ze zbioru `{a, b, c}`,
- drugi znak hasła to cyfra za zbioru `{4, 6, 8}`
- trzeci znak hasła to znak specjalny ze zbioru: `{*, %, :}`

złam hash. (Przykładowe hasła to: `a8*`, `b6%`, `c4:` ....)

(d) Nie generuj własnych, ani nie używaj gotowych słowników. Wykorzystaj metodę bruteforce, definiując wzorzec hasła używając odpowiedniej maski. (Maski to wzorce definiujące strukturę hasła. Zamiast sprawdzać wszystkie możliwe kombinacje, możesz określić dokładny format hasła.)

(e) Do złamania hasha użyj narzędzia hashcat lub John the Ripper.

(f) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4014/submit` wartość złamanego hasha (jako `word`).

(g) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.15 *Pod adresem `http://127.0.0.1:4015` działa prosty serwer HTTP udostępniający dwa endpointy: `http://127.0.0.1:4015/hash` oraz `http://127.0.0.1:4015/submit`.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4015:4015 --name ex15 docker.io/mazurkatarzyna/pass-cracking-ex15:latest
podman run -p 4015:4015 --name ex15 docker.io/mazurkatarzyna/pass-cracking-ex15:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4015:4015 --name ex15 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex15:latest
podman run -p 4015:4015 --name ex15 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex15:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4015/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-1** do złamania (jako `hash`).

(c) Wiedząc, że **hasło jest adresem MAC, zaczynającym się od** `00:14:22` **(OUI Dell), gdzie kolejne dwa bajty również są stałe, i równe** `FF:FF`, złam hash.
(Przykładowe hasła to: `00:14:22:ff:ff:01`, `00:14:22:ff:ff:02` ....)

(d) Nie generuj własnych, ani nie używaj gotowych słowników. Wykorzystaj metodę bruteforce, definiując wzorzec hasła używając odpowiedniej maski. (Maski to wzorce definiujące strukturę hasła. Zamiast sprawdzać wszystkie możliwe kombinacje, możesz określić dokładny format hasła.)

(e) Do złamania hasha użyj narzędzia hashcat lub John the Ripper.

(f) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4015/submit` wartość złamanego hasha (jako `word`).

(g) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

Łamanie haseł - Hybrid attacks (dictionary + mask/rule)

4.16 *Pod adresem* `http://127.0.0.1:4016` *działa prosty serwer HTTP udostępniający dwa endpointy:* `http://127.0.0.1:4016/hash` *oraz* `http://127.0.0.1:4016/submit`.

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4016:4016 --name ex16 docker.io/mazurkatarzyna/pass-cracking-ex16:latest
podman run -p 4016:4016 --name ex16 docker.io/mazurkatarzyna/pass-cracking-ex16:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4016:4016 --name ex16 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex16:latest
podman run -p 4016:4016 --name ex16 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex16:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4016/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-256** do złamania (jako `hash`).

(c) Wiedząc, że **hasło jest słowem ze słownika zawierającego najpopularniejsze hasła, do którego dodano na końcu dwie cyfry (każda od 0 do 9)**, złam hash. (Hasło to jedno z 10 pierwszych haseł **ze słownika zawierającego najpopularniejsze hasła**) z dodanymi 2 cyframi na końcu, czyli np. `shadow39` czy `sunshine00`.)

(d) W celu złamania hasha, wykorzystaj **słownik popularnych haseł**, oraz zastosuj odpowiednią maskę, dodającą do każdego słowa ze słownika 2 cyfry na końcu słowa.

(e) Do złamania hasha użyj narzędzia `hashcat` lub `John the Ripper`.

(f) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4016/submit` wartość złamanego hasha (jako `word`).

(g) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.17 *Pod adresem* `http://127.0.0.1:4017` *działa prosty serwer HTTP udostępniający dwa endpointy:* `http://127.0.0.1:4017/hash` *oraz* `http://127.0.0.1:4017/submit`.

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4017:4017 --name ex17 docker.io/mazurkatarzyna/pass-cracking-ex17:latest
podman run -p 4017:4017 --name ex17 docker.io/mazurkatarzyna/pass-cracking-ex17:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4017:4017 --name ex17 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex17:latest
podman run -p 4017:4017 --name ex17 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex17:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4017/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-256** do złamania (jako `hash`).

(c) Wiedząc, że **hasło jest słowem ze słownika zawierającego najpopularniejsze hasła, do którego dodano na początku małą literę (każda od a do z), cyfrę (od 0 do 9), oraz znak specjalny ze zbioru znaków !, @, #, $, %, &, ***, złam hash. (Hasło to jedno z 10 pierwszych haseł **ze słownika zawierającego najpopularniejsze hasła**) z dodanymi 3 znakami na początku słowa, czyli np. `f4@shadow` czy `k9!sunshine`.)

(d) W celu złamania hasha, wykorzystaj **słownik popularnych haseł**, oraz zastosuj odpowiednią maskę, dodającą do każdego słowa ze słownika 2 cyfry na końcu słowa.

(e) Do złamania hasha użyj narzędzia hashcat lub John the Ripper.

(f) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4017/submit` wartość złamanego hasha (jako `word`).

(g) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.18 *Pod adresem `http://127.0.0.1:4018` działa prosty serwer HTTP udostępniający dwa endpointy: `http://127.0.0.1:4018/hash` oraz `http://127.0.0.1:4018/submit`.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4018:4018 --name ex18 docker.io/mazurkatarzyna/pass-cracking-ex18:latest
podman run -p 4018:4018 --name ex18 docker.io/mazurkatarzyna/pass-cracking-ex18:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4018:4018 --name ex18 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex18:latest
podman run -p 4018:4018 --name ex18 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex18:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4018/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-256** do złamania (jako `hash`).

(c) Wiedząc, że **hasło to jedno z żeńskich imion z języka polskiego, przekształconych w następujący sposób:**

- litera `a` jest zamieniona na `@`,
- litera `e` jest zamieniona na `3`,
- litera `i` jest zamieniona na `1`,

złam hash. Przykładowe hasła to: `n3l@`, `kun3gund@`, ... (Sprawdź też czym jest **Leet speak**.)

(d) W celu złamania hasha, wykorzystaj **słownik polskich imion żeńskich**, oraz stwórz plik z powyższymi regułami.

(e) Do złamania hasha użyj narzędzia hashcat lub John the Ripper.

(f) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4018/submit` wartość złamanego hasha (jako `word`).

(g) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.19 *Pod adresem `http://127.0.0.1:4019` działa prosty serwer HTTP udostępniający dwa endpointy: `http://127.0.0.1:4019/hash` oraz `http://127.0.0.1:4019/submit`.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4019:4019 --name ex19 docker.io/mazurkatarzyna/pass-cracking-ex19:latest
podman run -p 4019:4019 --name ex19 docker.io/mazurkatarzyna/pass-cracking-ex19:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4019:4019 --name ex19 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex19:latest
podman run -p 4019:4019 --name ex19 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex19:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4019/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-256** do złamania (jako `hash`).

(c) Wiedząc, że **hasło to jedno z męskich imion z języka polskiego, przekształconych w następujący sposób:**

- pierwsza litera imienia jest wielką literą,
- litery są zamienione w następujący sposób: `a` na `4`, `e` na `3`, `o` na `0`,
- na końcu słowa jest dodany znak `_` oraz bieżący rok i znak `!`

złam hash. Przykładowe hasła to: `J4c3k_2025!`, `Andrz3j_2025!`, ... (Sprawdź też czym jest **Leet speak**.)

(d) W celu złamania hasha, wykorzystaj **słownik polskich imion męskich**, oraz stwórz plik z powyższymi regułami.

(e) Do złamania hasha użyj narzędzia `hashcat` lub `John the Ripper`.

(f) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4019/submit` wartość złamanego hasha (jako `word`).

(g) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

4.20 *Pod adresem `http://127.0.0.1:4020` działa prosty serwer HTTP udostępniający dwa endpointy: `http://127.0.0.1:4020/hash` oraz `http://127.0.0.1:4020/submit`.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4020:4020 --name ex20 docker.io/mazurkatarzyna/pass-cracking-ex20:latest
podman run -p 4020:4020 --name ex20 docker.io/mazurkatarzyna/pass-cracking-ex20:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4020:4020 --name ex20 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex20:latest
podman run -p 4020:4020 --name ex20 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex20:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4020/hash` używając metody HTTP `GET`. Otrzymasz w odpowiedzi **hash SHA-256** do złamania (jako `hash`).

(c) Wiedząc, że **hasło to słowo ze słownika CERT Polska, z dodanym znakiem `!` na początku i znakiem `#` na końcu** słowa, złam hash.

(d) W celu złamania hasha, wykorzystaj słownik popularnych polskich haseł, zebrany podczas **wycieków danych**, oraz stwórz plik z powyższymi regułami.

(e) Do złamania hasha użyj narzędzia `hashcat` lub `John the Ripper`.

(f) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4020/submit` wartość złamanego hasha (jako `word`).

(g) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie - zweryfikuje poprawność złamanego hasha.

Łamanie haseł plików `*.zip`, `*.rar`, ...

4.21 *Pod adresem* `http://127.0.0.1:4021` *działa prosty serwer HTTP udostępniający dwa endpointy:* `http://127.0.0.1:4021/zip` *oraz* `http://127.0.0.1:4021/submit`.

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4021:4021 --name ex21 docker.io/mazurkatarzyna/pass-cracking-ex21:latest
podman run -p 4021:4021 --name ex21 docker.io/mazurkatarzyna/pass-cracking-ex21:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4021:4021 --name ex21 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex21:latest
podman run -p 4021:4021 --name ex21 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex21:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4021/zip` używając metody HTTP `GET`. Otrzymasz w odpowiedzi plik `*.zip` zaszyfrowany hasłem.

(c) Wiedząc, że hasło jest popularnym hasłem znajdującym się w **słownikach popularnych haseł**, używając:

- **fcrackzip**
- **hashcat** wraz z **zip2john**
- **John the Ripper** wraz z **zip2john**

złam hasło i rozpakuj plik `*.zip`.

(d) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4021/submit` wartość złamanego hasła (jako `password`).

(e) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie.

4.22 *Pod adresem* `http://127.0.0.1:4022` *działa prosty serwer HTTP udostępniający dwa endpointy:* `http://127.0.0.1:4022/zip` *oraz* `http://127.0.0.1:4022/submit`.

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4022:4022 --name ex22 docker.io/mazurkatarzyna/pass-cracking-ex22:latest
podman run -p 4022:4022 --name ex22 docker.io/mazurkatarzyna/pass-cracking-ex22:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4022:4022 --name ex22 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex22:latest
podman run -p 4022:4022 --name ex22 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex22:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4022/zip` używając metody HTTP `GET`. Otrzymasz w odpowiedzi plik `*.zip` zaszyfrowany hasłem.

(c) Wiedząc, że **hasło jest hasłem o długości pomiędzy 5-6 znaków, i zawiera jedynie cyfry od 0 do 9**, używając:

- **fcrackzip**
- **hashcat** wraz z **zip2john**
- **John the Ripper** wraz z **zip2john**

złam hasło i rozpakuj plik `*.zip`. Możesz wygenerować swój własny słownik, używając narzędzia `crunch`.

(d) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4022/submit` wartość złamanego hasła (jako `password`).

(e) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie.

4.23 *Pod adresem* `http://127.0.0.1:4023` *działa prosty serwer HTTP udostępniający dwa endpointy:* `http://127.0.0.1:4023/pdf` *oraz* `http://127.0.0.1:4023/submit`. *Na endpoincie* `http://127.0.0.1:4023/pdf` *serwer zwraca plik* `*.pdf` *zaszyfrowany numerem PESEL.*

(a) Uruchom serwer za pomocą poniższego polecenia:

```
docker run -p 4023:4023 --name ex23 docker.io/mazurkatarzyna/pass-cracking-ex23:latest
podman run -p 4023:4023 --name ex23 docker.io/mazurkatarzyna/pass-cracking-ex23:latest
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run -p 4023:4023 --name ex23 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex23:latest
podman run -p 4023:4023 --name ex23 ghcr.io/mazurkatarzynaumcs/pass-cracking-ex23:latest
```

(b) Wyślij request do endpointa `http://127.0.0.1:4023/pdf` używając metody HTTP `GET`. Otrzymasz w odpowiedzi plik `*.pdf` zaszyfrowany hasłem. Hasło jest numerem pesel.

(c) Wiedząc, że **hasło jest numerem pesel osoby, która urodziła się pomiędzy 01.07.1992 a 31.12.1992 i jest kobietą,** oraz używając:

- **pdfcrack**
- **hashcat** wraz z **pdf2john**
- **John the Ripper** wraz z **pdf2john**

złam hasło i odczytaj zawartość pliku `*.pdf`.

(d) Za pomocą metody HTTP `POST`, wyślij pod endpoint `http://127.0.0.1:4023/submit` wartość złamanego hasła (jako `password`).

(e) W odpowiedzi serwer zwróci odpowiednią informację o sukcesie lub błędzie.

(f) Jeśli nie posiadasz zainstalowanego `pdf2john` w swoim systemie, możesz skorzystać z jego Dockerowej wersji, którą uruchomisz w następujący sposób:

```
docker run --rm -v $(pwd):/data docker.io/mazurkatarzyna/pdf2john:latest challenge.pdf > hash.txt
podman run --rm -v $(pwd):/data docker.io/mazurkatarzyna/pdf2john:latest challenge.pdf > hash.txt
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run --rm -v $(pwd):/data ghcr.io/mazurkatarzynaumcs/pdf2john:latest challenge.pdf > hash.txt
podman run --rm -v $(pwd):/data ghcr.io/mazurkatarzynaumcs/pdf2john:latest challenge.pdf > hash.txt
```

gdzie `challenge.pdf` to plik `*.pdf` pobrany z serwera, a `hash.txt` to hash wyodrębniony z pliku `*.pdf`.

(g) Jeśli nie posiadasz zainstalowanego `pdfcrack` w swoim systemie, możesz skorzystać z jego Dockerowej wersji, którą uruchomisz w następujący sposób:

```
docker run --rm -v $(pwd):/data docker.io/mazurkatarzyna/pdfcrack:latest -f challenge.pdf -w wordlist.txt
podman run --rm -v $(pwd):/data docker.io/mazurkatarzyna/pdfcrack:latest -f challenge.pdf -w wordlist.txt
```

Jeśli Docker Hub nie odpowiada, użyj obrazu zapasowego:

```
docker run --rm -v $(pwd):/data ghcr.io/mazurkatarzynaumcs/pdfcrack:latest -f challenge.pdf -w wordlist.txt
podman run --rm -v $(pwd):/data ghcr.io/mazurkatarzynaumcs/pdfcrack:latest -f challenge.pdf -w wordlist.txt
```

gdzie `challenge.pdf` to plik `*.pdf` pobrany z serwera, `wordlist.txt` to słownik, a `hash.txt` to hash wyodrębniony z pliku `*.pdf`.

(h) Kilka ciekawych artykułów dotyczących plików zabezpieczonych peselem jako hasłem:

- https://informatykzakladowy.pl/lamiemy-haslo
- https://informatykzakladowy.pl/szyfrowanie-dokumentow-numerem-pesel
- https://sekurak.pl/dostalem-pit-a-w-pdfie
- https://niebezpiecznik.pl/post/jak-zlamac-haslo-do-zipa-pesel/