

SRP Vježbe2

Cilj vježbe je riješiti crypto izazov. Svaki student dobiva vlastiti ciphertext koji je potrebno dešifrirati uz činjenicu da student nema pristup enkripcijskom ključu. Koristimo brute force metodu.

Za pripremu *crypto* izazova, odnosno enkripciju korištena su Python biblioteke cryptography i Fernet. Fernet koristi sljedeće *low-level* kriptografske mehanizme:

- AES šifru sa 128 bitnim ključem u CBC enkripcijskom načinu rada
- HMAC (*hash MAC*) sa 256 bitnim ključem za zaštitu integriteta poruka
- Timestamp za osiguravanje svježine (*freshness*) poruka

Koraci

Otvaramo srp docker kontenjer

Otkrivamo ime personaliziranog izazova za svakog studenta izvršavanjem koda upisujući svoje ime i prezime u obliku "prezime_ime"

```
from cryptography.hazmat.primitives import hashes
def hash(input):
    if not isinstance(input, bytes):
        input = input.encode()
    digest = hashes.Hash(hashes.SHA256())
    digest.update(input)
    hash = digest.finalize()
    return hash.hex()
filename = hash('prezime_ime') + ".encrypted"
```

Profesor nam daje hint, tj. znamo da je datoteka koju dešifriramo PNG file, stoga definiramo funkciju koja provjerava PNG format

```
def test_png(header):
    if header.startswith("\211PNG\r\n\032\n"):
        return True
    return False
```


Kod za brute force funkciju. Ulazi se u petlju i Fernet rješenju za enkripciju šalje ključeve jeadn po jedan, tražeći vrijednost koja ima PNG format.

Za ovaj izazov je korištena 22 bitna enkripcija i samim time broj mogućih kombinacija je bio 2^{22} , tj oko 4 milijuna kombinacija. U prosjeku, brute force će trebati obaviti barem pola mogućih kombinacija da dođe do rješenja

```
import base64
from cryptography.hazmat.primitives import hashes
from cryptography.fernet import Fernet

def brute_force():
    filename = "ime_moje_datoteke.encrypted"
    with open(filename, "rb") as file:
        ciphertext = file.read()
    ctr = 0
    while True:
        key_bytes = ctr.to_bytes(32, "big")
        key = base64.urlsafe_b64encode(key_bytes)
        if not (ctr + 1) % 1000:
            print(f"[*] Keys tested: {ctr + 1:,}", end="\r")
        try:
            plaintext = Fernet(key).decrypt(ciphertext)
            header = plaintext[:32]
            if test_png(header):
                print(f"[+] KEY FOUND: {key}")
                with open("BINGO.png", "wb") as file:
                    file.write(plaintext)
                break
        except Exception:
            pass
        ctr += 1
    if __name__ == "__main__":
        brute_force()
```

Rješenje izazova tj. dobiveni png:



Congratulations Maretic Josip
You made it!