

Bruteforce Task

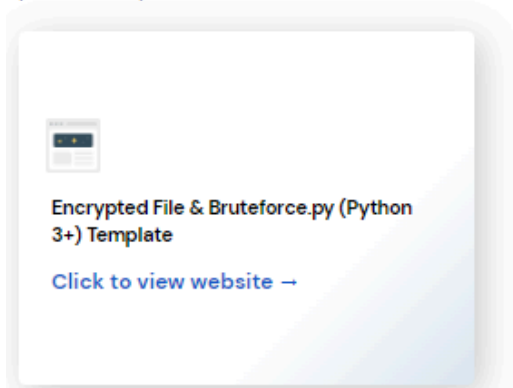
Here is the background information for your task

In this task, you will write a Python script to bruteforce the decryption key of the encrypted file.

Bruteforcing is the act of repeatedly trying different combinations to break the password encryption (based on either randomly generated passwords, or from a list of passwords to try). In the resource below, we've provided a small subset of passwords from Rockyou – a widely know password wordlist that contains thousands of common passwords in one wordlist.

Ransomware will often encrypt all files on a device, and sometimes give the decryption key after the ransom has been paid (but this is not always the case!). In this task, we would like you to break the encryption without paying the ransom.

A foundational Python 3+ template has also been provided for you in the resource below. One potential implementation is described in the code comments.



After, open the decrypted word doc and **paste your Python code in the text field below**. We'll show you an example answer on the next step, but we encourage you to give it a go first!

My Response

```
from zipfile import ZipFile, BadZipFile
import sys

def attempt_extract(zf_handle, password):
    """
    Attempt to extract the ZIP file with a given password.

    :param zf_handle: The ZipFile handle object
    :param password: The password to attempt
    :return: True if extraction was successful, False otherwise
    """
    try:
        # Attempt to extract the file using the provided password
        zf_handle.extractall(pwd=password)
        return True
    except (RuntimeError, BadZipFile, ValueError) as e:
        # If extraction fails, return False
        return False

def main():
    print("[+] Beginning bruteforce ")
    # Open the encrypted ZIP file
    with ZipFile('enc.zip') as zf:
        # Open the Rockyou password list
        with open('rockyou.txt', 'r', encoding='utf-8') as f:
            # Iterate through each password in the list
            for line in f:
                # Strip any whitespace from the password
                password = line.strip().encode('utf-8')

                print(f"Trying password: {password.decode('utf-8')}")

                # Attempt to extract the ZIP file with the current password
                if attempt_extract(zf, password):
                    print(f"[+] Password found: {password.decode('utf-8')}")
                    break
            else:
                # If the loop completes without finding the correct password
                print("[+] Password not found in list")

if __name__ == "__main__":
    main()
```

