XLS RECOVERY INJECT REPORT

3/11/2023

Bilal Anarwala

Table of Contents

1 EXECUTIVE SUMMARY

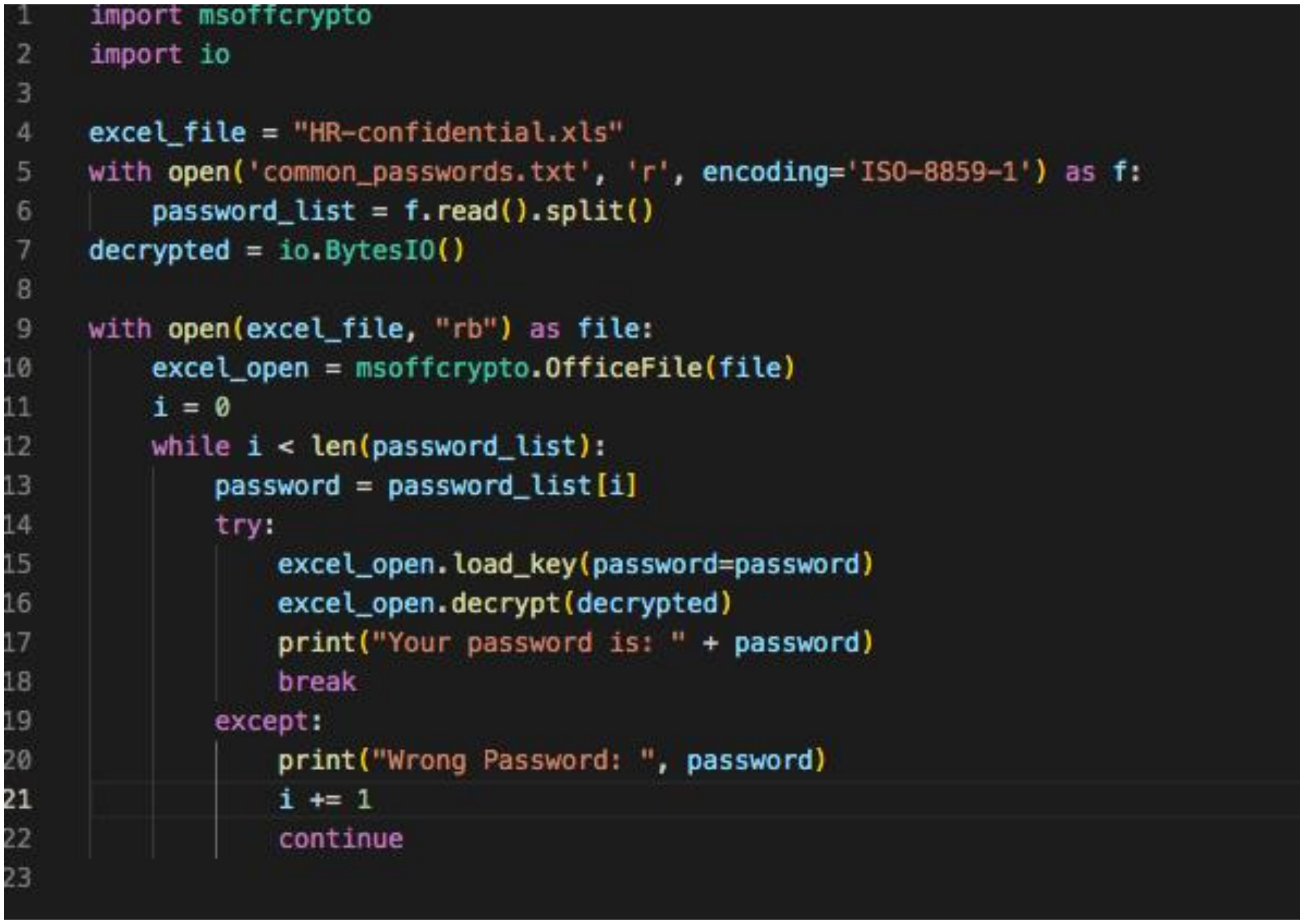
The new HR director, Mr. Frank-Lee InTrouble is unable to access an excel sheet containing codes for payroll raise processing. This is because the previous HR director put a password on it that is unknown to the company. With no way to contact the previous HR director, the information security team went ahead and cracked the password. This was done using a dictionary attack. As a result, the company can move forward with processing pay raises for its consultants in the upcoming fiscal year. This document will walkthrough how the password was cracked as well as provide alternative ways to crack passwords like this in the future.

2 THE PROCEDURE

After much research, I figured that the best way to approach this was to use Python given that it has a plethora of modules and API's that could be utilized to break the password. With some research, I found a Python module called "msoffcrypto" which is a tool used specifically for cracking any type of Microsoft Office file that is password protected. With that taken care of, the next thing I did was utilize a well-known file that stores a list of commonly used passwords that people have used. Using this gives us a huge list of passwords that Python can attempt to use on the excel file to crack the password. To process this file, I utilized a Python module known as IO. This module provides an easier and efficient way to create and manipulate files, streams, and other data sources. With these two modules and the list of commonly used passwords, it was time to write the script.

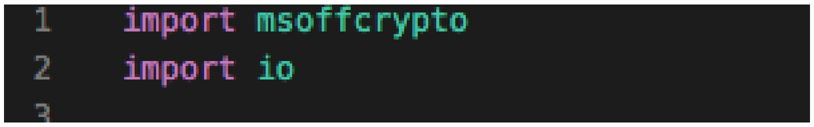
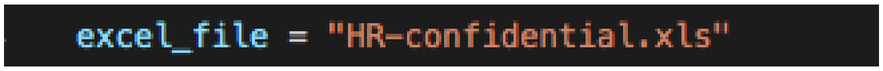
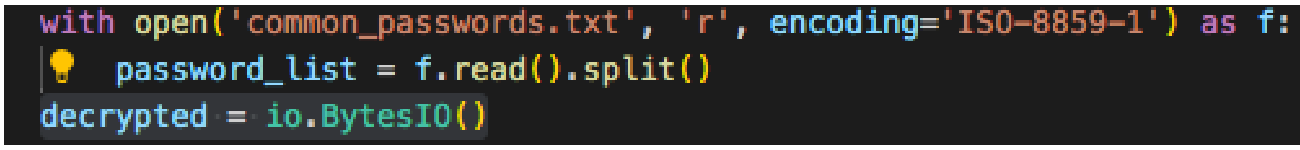
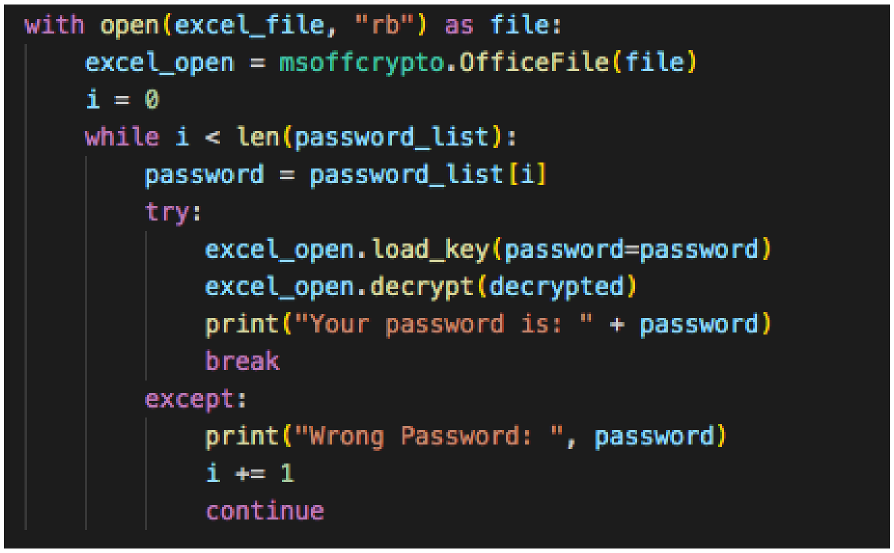
2.1 The Script

Below is the Python script that was written to crack the excel sheet password:



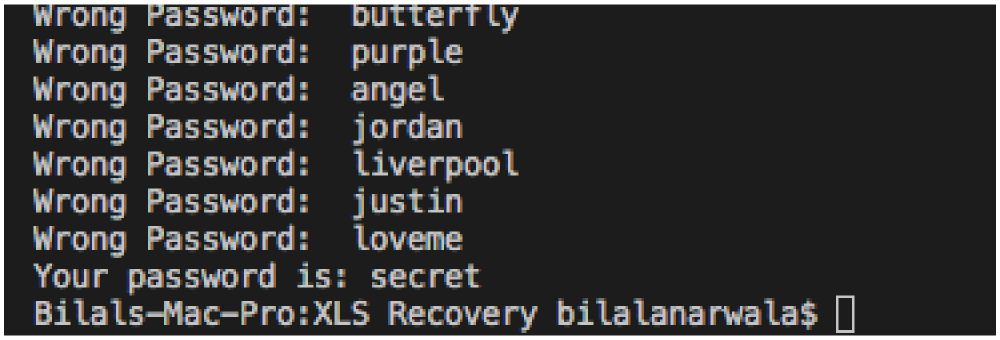
```python
import msoffcrypto
import io

excel_file = "HR-confidential.xls"
with open('common_passwords.txt', 'r', encoding='ISO-8859-1') as f:
    password_list = f.read().split()
decrypted = io.BytesIO()

with open(excel_file, "rb") as file:
    excel_open = msoffcrypto.OfficeFile(file)
    i = 0
    while i < len(password_list):
        password = password_list[i]
        try:
            excel_open.load_key(password=password)
            excel_open.decrypt(decrypted)
            print("Your password is: " + password)
            break
        except:
            print("Wrong Password: ", password)
            i += 1
            continue
```

2.2 Explanation of the Script

1. I began by importing the modules at the very top. As mentioned before, they were the "msoffcrypto" and "io" modules.
   a. 
   ```python
   import msoffcrypto
   import io
   ```
2. Next, I created a variable that stored the name of the excel file which was "HR-confidential.xls". This variable can also be the directory location of the file. Since I kept the excel file in the same folder as the
   a. 
   ```python
   excel_file = "HR-confidential.xls"
   ```
3. I then created a variable that opens the file containing the list of common passwords used named "common_passwords.txt" in read mode and dumped its contents into a list named "password_list". This is followed by the creation of a variable called "decrypted" that stores an empty "BytesIO" object.
   a. 
   ```python
   with open('common_passwords.txt', 'r', encoding='ISO-8859-1') as f:
       password_list = f.read().split()
   decrypted = io.BytesIO()
   ```
4. Following that, the Excel file is opened in binary mode and read into the variable "file". The "msoffcrypto.OfficeFile" method is used to open the excel file into a variable called "excel_open". The script then begins a while-loop that will iterate through each password in the "password_list". Within the while-loop, the current password is attempted on the excel file which is opened through the "excel_open " variable using the "load_key" method. If the password is correct, the file is decrypted using the "decrypt" method and the password is printed to the console. If the password is incorrect, the script will print a message indicating that the password was incorrect, and the loop will then continue to the next password in the list until the correct password is found. Finally, Once the correct password is found and the file is decrypted, the loop will terminate, and the script will stop running
   a. 
   ```python
   with open(excel_file, "rb") as file:
       excel_open = msoffcrypto.OfficeFile(file)
       i = 0
       while i < len(password_list):
           password = password_list[i]
           try:
               excel_open.load_key(password=password)
               excel_open.decrypt(decrypted)
               print("Your password is: " + password)
               break
           except:
               print("Wrong Password: ", password)
               i += 1
               continue
   ```

3 THE RESULT

Once the script was written and run, it combed through the list of known passwords trying each one until it found a password that worked. It found that the password to the excel file was “secret”.



```
Wrong Password:  butterfly
Wrong Password:  purple
Wrong Password:  angel
Wrong Password:  jordan
Wrong Password:  liverpool
Wrong Password:  justin
Wrong Password:  loveme
Your password is: secret
Bilals-Mac-Pro:XLS Recovery bilalanarwala$ 
```

4 ALTERNATIVE METHODS/MOVING FORWARD

In addition to this Python script, there are a handful of other tools out there that are solely made to crack passwords. Examples include, John the Ripper, L0phtCrack, Hydra, and Hashcat. When it comes to the company as a whole, they should make sure that they do not limit access to these types of files to just one employee. This can prevent future situations like this where they are left with files, they do not have the password to. Giving more than one authorized employee access to these files can prevent a situation like this from happening again.

REFERENCES

Burns, William J. "Common Password List ." *Kaggle*, 13 Jan. 2019, https://www.kaggle.com/datasets/wjburns/common-password-list-rockyoutxt.

Greistaen. "Encrypted MS Files." *Dataiku Community*, Khoros, 10 Jan. 2023, https://community.dataiku.com/t5/Setup-Configuration/Encrypted-Excel-files/m-p/31741.

Horbach, Ariane. "The Infamous Dictionary Attack in Words and Code." *Medium*, Medium, 19 Nov. 2022, https://medium.com/@arianehorbach/the-basics-of-the-basic-dictionary-attack-2234b72a4fee.

"IO - Core Tools for Working with Streams." *Python Documentation*, Python Software Foundation, https://docs.python.org/3/library/io.html.

Nolze. "Msoffcrypto Module." *PyPI*, https://pypi.org/project/msoffcrypto-tool/.